

**Serie:
Docencia Universitaria - EEES**

**INVESTIGACIÓN EN DISEÑO DOCENTE
DE LOS ESTUDIOS DE SEGUNDO CURSO
DE INFORMÁTICA**

Migallón Gomis, Violeta; Saiz Noeda, Maximiliano (Eds.)

Escuela Politécnica Superior

Vicerrectorado de Calidad y Armonización Europea
Instituto de Ciencias de la Educación
Universidad de Alicante



Esta publicación no puede ser reproducida, ni totalmente ni parcialmente, ni registrada, ni transmitida por un sistema de recuperación de información, ya sea fotomecánico, electrónico, por fotocopia o cualquier otro medio, sin el permiso previo de los propietarios de copyright.

SERIE: DOCENCIA UNIVERSITARIA - EEES
Dirección de la Serie: M.^a Ángeles Martínez Ruiz
Coordinador de la Obra: Francisco Ramón Pastor Verdú

ICE/VICERRECTORADO DE CALIDAD Y ARMONIZACIÓN EUROPEA
UNIVERSIDAD DE ALICANTE

© del texto: Los autores

© de esta edición: Editorial Marfil, S.A.
C/ San Eloy, 17 • 03804 Alcoy
Tel.: 96 552 33 11 • Fax: 96 552 34 96
e-mail: editorialmarfil@editorialmarfil.com

Universidad de Alicante
Campus de Sant Vicent del Raspeig
03080 Alicante

Foto portada: N. Sauleda
Portada: Francisco Ramón Pastor Verdú

Maquetación: Sergio Gracia Martínez
Maximiliano Saiz Noeda

I.S.B.N.: 978-84-268-1146-2

Depósito legal: A-315-2007

Fotomecánica, fotocomposición e impresión:
Gráficas Alcoy • San Eloy, 17 • 03804 ALCOY



Prólogo

Durante el curso académico 2004-2005, en la Escuela Politécnica Superior de la Universidad de Alicante, se realizó un trabajo sobre la adecuación del primer curso de los estudios de Informática al Espacio Europeo de Educación Superior. Para mí fue un honor, como subdirector de Informática en dicho momento, coordinar el grupo de trabajo. Este grupo estuvo fundamentalmente formado por los profesores de primer curso de las titulaciones de Informática. Los resultados se vieron plasmados en un libro donde se contenía nuestra visión de lo que debe ser un primer curso de Informática.

No obstante, además del interés del libro *publicado*, pude extraer muchísimas conclusiones de las numerosas e interminables reuniones que llevamos a cabo. La primera fue comprobar la implicación que la mayoría de los profesores tuvieron en este trabajo. En unos momentos donde se critica al profesor universitario por su falta de implicación en la docencia, pude comprobar que estas críticas carecen totalmente de fundamento, no ya sólo por el número de horas trabajado por cada profesor, bien de forma individual o bien en grupo, sino por la ilusión con la que se nombraban los conceptos de docencia, innovación, calidad, nuevas metodologías, Europa y muchos otros. Elementos que se asociaban, indefectiblemente, al Espacio Europeo de Educación Superior (EEES).

El inicio del concepto de EEES, que procede de la famosa Declaración de Bolonia realizada en 1999, está cada vez más lejos en el tiempo. Por otro lado, el objetivo fijado por el mismo, la armonización de las enseñanzas y titulaciones universitarias impartidas en Europa, se halla, irremisiblemente, cada vez más cerca, ya que el año 2010 era el año en el que se pretendía cubrir dicho objetivo.

Este cercano objetivo ha hecho que, desde la Escuela Politécnica Superior (EPS), hayamos decidido continuar con este proceso de trabajo en el EEES. Durante este curso académico (2005-2006), mientras los profesores de primer curso de Informática ponían en funcionamiento algunos de los acuerdos tomados el curso anterior, los profesores de segundo curso han asumido el relevo de sus compañeros de primero y han trabajado de la misma forma en hacer la propuesta de lo que debería ser, a nuestro entender, el segundo curso de Informática.

Además hay un aspecto que realza el trabajo realizado este año. En el momento de escribir estas líneas, flota en el aire una sensación de oscuros presagios, que confío que, en el momento de ser leídas por usted, no sea más que algo olvidado. Estoy hablando de la indecisión que a veces se vislumbra por parte de las autoridades académicas a nivel nacional, de poner en marcha el modelo EEES. El modelo EEES habla de legibilidad de los títulos, que se iba a conseguir con la redefinición del catálogo de titulaciones disponibles, que parece que, finalmente, no se va a poner en marcha. La estructura de grado y pos-

tgrado que se iba a seguir, ha sido puesta en tela de juicio, con lo cual parece que se transmite la idea de cambiar para que nada cambie.

En la EPS somos conscientes de la importancia del proceso de mejora docente y proponemos nuevos modelos para que las cosas sí que cambien, y cambien a mejor. Aquí tienes nuestra propuesta de lo que debe ser un segundo curso de Ingeniería Informática. Esta propuesta ha sido confeccionada en base a un minucioso estudio de los problemas actuales y constituye un conjunto de iniciativas que estamos seguros mejorarán la formación de nuestros estudiantes.

Quiero agradecer sinceramente a todos los profesores que han trabajado en este libro, su implicación y compromiso con la mejora docente, en especial al profesor Maximiliano Saiz Noeda, coordinador de un grupo muy numeroso de profesores y asignaturas que ha encabezado este gran reto. Sé que en la Escuela Politécnica Superior de la Universidad de Alicante contamos con un profesorado de calidad muy implicado en la docencia. Esta implicación se comprueba día a día en trabajos como éste, que ahora tiene en sus manos, y que a mí me permite asegurar que siempre estaremos en la vanguardia del proceso educativo.

Fernando Llopis Pascual
Director EPS-UA

ÍNDICE

<i>El segundo curso de Informática en el Espacio Europeo de Educación Superior</i> Maximiliano Saiz Noeda	9
<i>Guía Docente de Ampliación de Estadística</i> Violeta Migallón Gomis; José Penadés Martínez; Rafael Álvarez Sánchez	59
<i>Guía Docente de Arquitecturas de Computadores</i> José García Rodríguez; Antonio Soriano Payá; Andrés Fuster Guillo; José Antonio Serra Pérez	89
<i>Guía Docente de Bases de Datos I</i> Armando Suárez Cueto; Patricio Martínez Barco; Andrés Montoyo Guijarro; Cristina Cachero Castro; Eva Gómez Ballester; Paloma Moreda Pozo; Rafael Muñoz Guillena; Manuel Palomar Sanz; Juan Carlos Trujillo Mondéjar; Sonia Vázquez Pérez	113
<i>Guía Docente de Computabilidad</i> Pilar Arques Corrales; Rafael Molina Carmona; Ignacio Viché Clavel	141
<i>Guía Docente de Estructuras de Computadores</i> Ángel Grediaga Olivo; Antonio Soriano Payá; Lluïsa Rico Soliveres	163
<i>Guía Docente de Herramientas de Programación</i> Antonio-M. Corbí Bellot; Alicia Garrido Alenda; Ramón Rubio Serna; José Oncina Carratalá	187
<i>Guía Docente de Lenguajes, Gramáticas y Autómatas</i> María Luisa Micó Andrés; Jorge Calera Rubio; Mikel L. Forcada Zubizarreta; José Luis Verdú Más	217
<i>Guía Docente de Lenguajes y Paradigmas de Programación</i> Domingo Gallardo López; Cristina Pomares Puig	241
<i>Guía Docente de Programación Orientada a Objetos</i> Cristina Cachero Castro; Estela Saquete Boró	293
<i>Guía Docente de Programación y Estructuras de Datos</i> Jesús Peral Cortés; Sergio Luján Mora; Antonio Ferrández Rodríguez	327
<i>Guía Docente de Sistemas Operativos</i> David Gil Méndez; Pedro Fuentes Gómez	383
<i>Guía Docente de Sistemas Operativos I</i> Higinio Mora Mora; Francisco José Mora Gimeno	409
<i>Guía Docente de Relaciones Jurídicas Básicas</i> M ^a Teresa Cantó López	439

EL SEGUNDO CURSO DE INFORMÁTICA EN EL ESPACIO EUROPEO DE EDUCACIÓN SUPERIOR

Maximiliano Saiz Noeda

*Departamento de lenguajes y Sistemas Informáticos
Universidad de Alicante
max@dlsi.ua.es*

1. CONTEXTO Y MOTIVACIÓN

1.1 EL ESPACIO EUROPEO DE EDUCACIÓN SUPERIOR

En mayo de 1998, los ministros encargados de la educación superior de Alemania, Francia, Italia y el Reino Unido suscribieron en París la Declaración de la Sorbona, instando al desarrollo de un Espacio Europeo de la Educación Superior (EEES). Un año más tarde, los ministros encargados de la educación superior de 29 países europeos celebraron una conferencia en Bolonia que sentó las bases para conseguir el EEES en 2010. La Declaración de Bolonia marca los objetivos de adoptar un sistema fácilmente legible y comparable de titulaciones basado en dos ciclos principales, establecer un sistema internacional de créditos, promover la movilidad de estudiantes, profesores e investigadores, promover la cooperación europea para garantizar la calidad de la educación superior y, en definitiva, promover una dimensión europea de la educación superior. Los ministros, esta vez 32, se volvieron a reunir en Praga en 2001, en Berlín en 2003 y en Bergen (Noruega) los días 19 y 20 de mayo de 2005. La próxima conferencia tendrá lugar en Londres en el 2007.

Todos los países firmantes de la Declaración de Bolonia han emprendido las reformas legislativas pertinentes para adaptarse al Espacio Europeo de Educación Superior. Hasta el momento en España se han aprobado cuatro reales decretos. Paralelamente a las reuniones de los ministros, la Comisión Europea se implicó en el proceso de convergencia de la educación superior europea. Así, la Comisión ha publicado varios documentos apoyando la iniciativa, entre los que

destaca la Comunicación de mayo de 2003: El papel de las universidades en la Europa del conocimiento.

Las organizaciones de universidades europeas han acogido muy positivamente la iniciativa ministerial. Así, la Asociación de la Universidad Europea (EUA) se implicó en el proceso y ha desarrollado varios estudios (Trends in Learning Structures) sobre la Educación Superior Europea. Además en las conferencias de Salamanca (2001), Graz (2003) y Glasgow (2005) hizo explícito su apoyo y sugerencias a la iniciativa del EEES. La Conferencia de Rectores de Universidades Españolas (CRUE) también ha generado varios documentos de análisis y apoyo al EEES.

1.2 EL CRÉDITO EUROPEO

El crédito europeo de transferencia y acumulación (el llamado crédito ECTS) aparece, por tanto, como un punto de referencia para lograr la transparencia y calidad en la formación que se exigen en la actualidad. La adopción del sistema de créditos ECTS implica una reorganización conceptual de los sistemas educativos para adaptarse a los nuevos modelos de formación centrados en el trabajo del estudiante.

El sistema ECTS constituye un código práctico que permite garantizar la transparencia y facilitar el reconocimiento académico y la organización de programas razonables en cuanto al volumen de trabajo a lo largo del período de estudios. Los elementos básicos en los que se fundamenta el sistema ECTS son:

- La utilización de créditos ECTS como valores que representan el volumen de trabajo efectivo del estudiante y el rendimiento obtenido mediante calificaciones comparables (grados ECTS).
- La información sobre los programas de estudios y los resultados de los estudiantes con documentos con un formato normalizado (Guía docente y certificados académicos).

Los créditos ECTS representan el volumen de trabajo del estudiante de manera relativa, no absoluta. Indican el volumen de trabajo requerido para superar cada módulo o asignatura, teniendo en cuenta el tiempo necesario para atender lecciones magistrales, realizar trabajos prácticos, asistir a seminarios, realizar períodos de prácticas, trabajo de campo, trabajo personal, etc., así como el tiempo necesario para preparar los exámenes u otros posibles métodos de evaluación. Así pues, el sistema ECTS se basa en el volumen total de trabajo del estudiante y no se limita exclusivamente a las horas de asistencia en clases presenciales.

En el marco del ECTS, 60 créditos representan el volumen de trabajo de un año académico. Su equivalencia en horas de trabajo para dicho estudiante medio es de aproximadamente 1600 horas:

$$8 \text{ horas diarias} \times 5 \text{ días a la semana} \times 40 \text{ semanas al año} = 1600 \text{ horas.}$$

Un crédito europeo representa entre 25 y 30 horas de trabajo del estudiante. El método recomendado para la asignación de créditos ECTS es seguir un procedimiento descendente: primero se determina el volumen de trabajo de un curso académico completo y se define con 60 créditos. A cada asignatura de ese curso se le asignará un número de créditos, según la proporción de trabajo que requiera en relación con el total.

Para determinar el volumen de trabajo necesario para una asignatura es preciso tener al menos una estimación general de las horas de trabajo personal dedicadas por los estudiantes a dicha asignatura. Para ello se pueden realizar encuestas, tanto a estudiantes como a profesores, con el fin de obtener datos que orienten y faciliten la asignación de créditos ECTS a las distintas materias. Esto permite contrastar el trabajo realizado por los estudiantes con el exigido por los profesores.

Se han detectado grandes diferencias entre las cargas de trabajo de diferentes asignaturas, así como en las estimaciones realizadas por profesores y alumnos en encuestas de este tipo, por lo que no se recomienda utilizar dichas encuestas como base exclusiva para el cálculo del volumen de trabajo, tal y como se menciona en el *Estudio sobre la asignación de créditos europeos a las distintas materias del currículum de matemáticas*, incluido en el *Documento de trabajo sobre la integración de los estudios españoles de matemáticas en el espacio europeo de enseñanza Superior*:

“La asignación de créditos basada en el volumen total de trabajo del estudiante reflejado en las encuestas no es aconsejable, ya que no hay razones académicas que justifiquen estas diferencias tan grandes en el volumen de trabajo de las asignaturas. La comparación de las encuestas realizadas en distintas universidades no confirma el hecho de que la dificultad esté en la materia en sí misma, si no en su plasmación concreta en una universidad. Aplicar el resultado de las encuestas directamente no haría más que consolidar las situaciones presuntamente patológicas reflejadas en ella. Las estimaciones de los estudiantes deben servir para detectar (y corregir) anomalías en asignaturas concretas, pero no para asignar créditos”.

Tanto en el documento anterior como en el Informe técnico sobre el crédito europeo y el sistema educativo español, se recomienda la utilización de la guía docente para la asignación de créditos ECTS, ya que ésta asigna a cada materia, factores que relacionan el número de horas presenciales y el número de horas de trabajo personal del estudiante. Este factor, para asignaturas con un número de horas de prácticas similar al número de horas de teoría, como la que nos ocupa, es aproximadamente 1,5 horas de estudio por hora presencial (en teoría y práctica).

1.3 EL EEES EN LA UNIVERSIDAD ESPAÑOLA

La adaptación al Espacio Europeo de Educación Superior va a significar un profundo cambio en la educación universitaria, tanto en las titulaciones como en los contenidos o la metodología docente. Las universidades tienen un importante papel que jugar en todo este proceso, y por eso muchas de ellas han creado unidades administrativas encargadas de gestionar la convergencia europea o están llevando a cabo experiencias piloto de adaptación al EEES.

Partiendo de las recomendaciones de las declaraciones de Sorbona (1998), Bologna (1999), la reunión de Salamanca (2000) y la cumbre de Barcelona (2000), así como las recomendaciones de diferentes estamentos estatales y universitarios, se admite que la mejor forma de organizar los estudios de informática en la universidad española es a través de un primer ciclo de cuatro años de duración que tenga un carácter generalista y que capacite para la obtención del título de Ingeniero en Informática con competencias y atribuciones profesionales. Este primer ciclo, además de capacitar para el ejercicio profesional, debe poder completarse con segundos ciclos o másters, que establezcan perfiles profesionales mucho más definidos y específicos que recojan las demandas del mercado laboral del momento, las tendencias de las nuevas tecnologías y las características propias del entorno socio-profesional y universitario donde se imparten.

Por su parte, la LOU [Ley Orgánica 6/2001, de 21 de diciembre, de Universidades] establece en su artículo 88, punto 3, que “el Gobierno, previo informe del Consejo de Coordinación Universitaria, establecerá las normas necesarias para que la unidad de medida del haber académico correspondiente a la superación de cada una de las materias que integran los planes de estudio de las diversas enseñanzas conducentes a la obtención de títulos de carácter oficial y validez en todo el territorio nacional, sea el crédito europeo o cualquier otra unidad que se adopte en el espacio europeo de enseñanza superior”. El Real Decreto 1125/2003, de 5 de septiembre [BOE núm. 224 de jueves 18 de septiembre de 2003, páginas 34355 y 34356], establece el sistema europeo de créditos (créditos ECTS) en las titulaciones universitarias de carácter oficial y validez en todo el territorio nacional. En el mismo se define el crédito europeo como “la cantidad de trabajo del estudiante para cumplir los objetivos del programa de estudios” y en el cual “se integran las enseñanzas teóricas y prácticas, así como otras actividades académicas dirigidas, con inclusión de las horas de estudio y de trabajo que el estudiante debe realizar”. En la asignación de créditos a cada una de las materias se tendrán en cuenta “las horas correspondientes a las clases lectivas, teóricas o prácticas, las horas de estudio, las dedicadas a la realización de seminarios, trabajos, prácticas o proyectos, y las exigidas para la preparación y realización de los exámenes y pruebas de evaluación”.

Las universidades europeas se encuentran inmersas en procesos de convergencia hacia un Espacio Europeo de Educación Superior (EEES) que ha de permitir un

reconocimiento más fácil de las titulaciones, asegurará una formación óptima de los estudiantes y su integración en un mercado laboral unificado y sin fronteras.

Entre las finalidades últimas de los procesos se deben considerar:

- El establecimiento de una Europa del Conocimiento para favorecer el crecimiento social y la formación de los ciudadanos
- El incremento en la competitividad a escala internacional
- El establecimiento de un sistema educativo de calidad

1.4 EL LIBRO BLANCO DEL TÍTULO DE GRADO EN INGENIERÍA INFORMÁTICA

La Conferencia de Decanos y Directores de Informática de España (CODDI) está constituida por los responsables de la totalidad de las escuelas y facultades que imparten estudios conducentes a las titulaciones de Ingeniería en Informática, Ingeniería Técnica en Informática de Gestión o Ingeniería Técnica en Informática de Sistemas, tanto públicas como privadas. Se creó en 1997 como foro de colaboración, debate e intercambio entre las distintas escuelas y facultades, alcanzando una máxima presencia al incorporar a la práctica totalidad de los actores implicados en la educación universitaria de informática. Desde su constitución, la CODDI se reúne, como mínimo, anualmente y con mayor frecuencia cuando los temas lo han requerido.

El interés por el tema del nuevo Espacio Europeo de Educación Superior se suscitó en febrero de 2001, durante la conferencia realizada en Las Palmas de Gran Canaria. Se acordó constituir una comisión de la CODDI para estudiar el impacto de la Declaración de Bolonia en los estudios de Informática en España. Esta comisión elaboró un documento de trabajo que fue presentado y debatido en el pleno de la Conferencia desarrollada en Barcelona en Mayo de 2002. Como consecuencia de esta discusión, la CODDI se manifestó públicamente a favor de una estructura adaptada al modelo de Bolonia y se comprometió a seguir trabajando en la configuración de estos futuros estudios.

En junio de 2003, en la Conferencia celebrada en Valencia, el tema principal fue “Los estudios de informática y la convergencia europea”. En este sentido se consideró la posibilidad de participar en la convocatoria del Programa de Convergencia Europea de la ANECA “Ayudas para el Diseño de Planes de Estudio y Títulos de Grado”, para desarrollar el Libro Blanco de las titulaciones de informática que permita la adaptación de las carreras de nuestro ámbito al nuevo marco del Espacio Europeo de Educación Superior. Así surge el proyecto EICE orientado a la creación del Libro Blanco del Título de Grado en Ingeniería Informática. En este proyecto se han implicado un total de 56 universidades nacionales, algo que pone de manifiesto el interés y el grado de compromiso del colectivo en el futuro de la titulación en el marco europeo.

Los objetivos de la Primera Convocatoria de Ayudas para el Diseño de Planes de Estudio y Títulos de Grado dentro del Programa de Convergencia Euro-

pea de la Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA), y en el cual se enmarca el proyecto EICE (Estudios de Informática y Convergencia Europea) son dos:

- Impulsar en las universidades españolas la realización de estudios y supuestos prácticos para el diseño de planes de estudio y de títulos oficiales de grado adaptados al Espacio Europeo de Educación Superior.
- Elaborar un Libro Blanco del título que recoja el resultado del estudio o supuesto práctico. Dicho estudio, editado por la Agencia Nacional de Evaluación de la Calidad y Acreditación, será remitido a la Dirección General de Universidades (MECD) y al Consejo de Coordinación Universitaria para su consideración.

Todo este proceso de adaptación al Espacio Europeo de Educación Superior ha comenzado su desarrollo sobre unas premisas establecidas por el Ministerio de Educación, Ciencia y Deporte en el Documento Marco publicado en febrero de 2003. En lo referente a los títulos de grado se parte, entre otros, de los siguientes prerequisites (extraído del apartado 5.2.1 del referido Documento Marco):

- Los objetivos formativos de las enseñanzas oficiales de nivel de grado tendrán, con carácter general, una orientación profesional, es decir, deberán proporcionar una formación universitaria en la que se integren armónicamente las competencias genéricas básicas, las competencias transversales relacionadas con la formación integral de las personas y las competencias más específicas que posibiliten una orientación profesional que permita a los titulados una integración en el mercado de trabajo.
- Asimismo, deberá ser posible una cierta flexibilidad que permita a las universidades diversificar su oferta, intensificando o personalizando alguna de las competencias específicas relacionadas con la orientación profesional.
- Asociado con los perfiles profesionales, deberá definirse un catálogo de títulos de primer nivel, tomando como punto de partida el actual, pero propiciando una disminución mediante las fusiones o agrupaciones necesarias para racionalizar el conjunto tanto desde el punto de vista nacional como europeo.

Esta última restricción condiciona el desarrollo y conclusiones de este proyecto, ya que las disciplinas que se agrupan bajo los estudios universitarios de Informática están en continua evolución y tienden a expandirse divergiendo, lo que parecería sugerir una ampliación del espectro de titulaciones universitarias de Informática.

Así, en las recomendaciones de ACM [14], se proponen cuatro titulaciones en el campo de la Informática (Computer Science, Computer Engineering, Software Engineering e Information Systems) y, en el Career Space [13], se encuentran más de una docena de perfiles profesionales asociados a nuestros títulos.

Existen, hoy en día, en España, tres titulaciones universitarias de Informática: una de dos ciclos, Ingeniería en Informática, y dos de un único ciclo, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de

Sistemas. El Libro Blanco establece un cierto equilibrio entre la oferta de estos estudios y su demanda en el contexto español. Son titulaciones ampliamente demandadas, con un cierto grado de consolidación social, aunque la percepción fina de lo que es la Ingeniería en Informática y lo que proporciona cada título dista mucho de estar al nivel que debiera.

El número total de titulados oficiales, más de 65000, constituye otro referente que nos lleva a reflexionar sobre el papel y la influencia de este colectivo en el tejido socioeconómico del país. Una vez más, esta presencia masiva no se corresponde con el peso real del mismo en cuanto al poder de decisión se refiere, incluso si lo restringimos al propio ámbito específico de la profesión. Es este un gran problema de nuestra profesión al que, sin más dilaciones, debemos poner remedio. Ello en gran parte puede venir respaldado por una adecuada definición de las capacidades profesionales de nuestros titulados, de sus competencias y, en definitiva, de sus posibilidades de desarrollo personal y aportaciones en las organizaciones. Debemos aportar personas capaces de adaptarse rápidamente al cambio, a trabajar en equipo, organizar su trabajo y el de sus colaboradores y muchas otras capacidades que se nos exigen.

En el contexto europeo actual, son muchos los países que ofertan distintas titulaciones de grado relacionadas con Informática (Alemania, Reino Unido, Suecia...) y un amplio espectro de titulaciones de máster relacionadas con las especializaciones y tendencias tecnológicas actuales.

Dado que uno de los objetivos planteados consistía en reducir el catálogo de titulaciones, el proyecto EICE se ha diseñado para definir una única titulación de grado: Ingeniería en Informática.

En esas condiciones, los miembros del proyecto han realizado el esfuerzo de converger a una única titulación. La enorme extensión de las diferentes disciplinas que componen lo que denominamos Informática, y su papel central en todo lo relacionado con las Tecnologías de la Información y la Comunicación (TIC) y con la Sociedad de la Información y del Conocimiento, podrían llevar a la tentación de crear múltiples titulaciones informáticas. Sin embargo, los artífices del libro blanco consideran que este tipo de planteamientos no benefician en nada la claridad de visualización de la profesión en nuestra Sociedad y no permiten adaptarse a los cambios en el sector, y a la velocidad a la que debe realizarse esta adaptación, con la flexibilidad y capacidad de respuesta que se nos puede demandar en el futuro.

La esencia fundamental del trabajo desarrollado en lo referente a la estructura que cubra la formación del ámbito de la Ingeniería en Informática se formula en los siguientes puntos:

- Se parte de la premisa que el mercado realizará un fuerte tirón de demanda en un futuro muy próximo. La presencia estratégica de la informática y su capilaridad hacen pensar en un conjunto de soluciones integradas que

- contemplan, a todos los niveles formativos, (educación primaria, secundaria y módulos formativos de grado medio y superior).
- En este sentido, deberá hacerse una reflexión en profundidad sobre la Programación Universitaria, su distribución geográfica y la dotación de los recursos adecuados para alcanzar los objetivos previstos.
 - Estructura organizada en dos ciclos: Grado y Máster.
 - Una única titulación de Grado denominada Ingeniería en Informática.
 - El título de Ingeniero en Informática comportará competencias profesionales plenas para el ejercicio de la profesión.
 - La formación que proporcionará el Grado será de carácter generalista.
 - Los estudios de Grado constarán de 240 créditos ECTS y estarán organizados en 4 años.
 - Entre los contenidos formativos fundamentales del Grado, se considera que debe integrarse en los estudios la realización de un Proyecto Fin de Carrera, que integre los conocimientos adquiridos durante los estudios y aproxime al estudiante a casos reales de la profesión, así como contenidos transversales que potencien habilidades propias del ejercicio de la profesión de ingeniero.
 - Se considera que los Contenidos Formativos Comunes de la titulación deben representar un 60% de la carga de los estudios, incluyendo la carga asignada al Proyecto Fin de Carrera, dejando el 40% restante para materias que serán determinadas discrecionalmente por cada Universidad.
 - Entre las materias a determinar por las universidades, se recomienda tener una oferta suficientemente numerosa de materias que procuren una formación amplia al estudiante en Tecnologías Informáticas actuales así como conocimientos de dominios concretos de aplicación de la informática.
 - El Máster estará destinado a la especialización profesional de los Ingenieros en Informática, o bien a su preparación para la investigación.
 - Se propone que el número de titulaciones de Máster sea el suficiente para cubrir la demanda de formaciones especializadas en cada momento.
 - Los estudios de Máster constarán de entre 60 y 120 créditos ECTS, y podrán incluir la cantidad asignada a la Tesis de Máster.
 - El Máster deberá permitir el acceso a la realización de la tesis doctoral con el objeto de obtener el grado de Doctor.
 - Consideramos básica la participación activa de las estructuras y personas que gestionarán el Grado en los planteamientos básicos, definición, articulación y gestión de los programas de Máster, especialmente en aquellos aspectos estrictamente asociados con la continuidad del Grado y de su conexión con el entorno socio-económico.
 - La gestión de los recursos dedicados al conjunto Grado-Máster será determinante para un desarrollo fructífero y armónico de la Ingeniería en Infor-

mática. El problema es de alta complejidad y la propia dinámica acelerada del sector obliga a planteamientos muy coordinados a fin de evitar desorientaciones, fracasos y malbaratamiento de los recursos disponibles.

- Poseemos una estructura universitaria de buen nivel y con una alta potencialidad de desarrollo. Nuestros titulados son reconocidos como competentes y trabajadores, con un gran nivel conceptual y profesional, aunque con algunas carencias en el ámbito de las competencias transversales que deben mejorarse con urgencia.
- Nuestros programas están sobrecargados y precisan de una reflexión muy profunda que permita mejorar las formas de trabajo, tanto del estudiante como del profesor, y aprovechar convenientemente los recursos globales de forma sensata.
- Las nuevas formas de aprendizaje exigen un esfuerzo extraordinario en todas las dimensiones. Adaptación de estudiantes y profesores, de materiales docentes, de forma de trabajo, de formas de evaluación, de formas de relación, de espacios docentes, de laboratorios dedicados, de prácticas en empresa bien articuladas y controladas y de un largo etcétera sobre el que planea la sombra de una enorme colisión si no se trabaja de forma decidida sobre los aspectos docentes de la actividad del profesorado.
- La dedicación a la docencia debe ser reconocida de forma clara y ha de ser válida en los procesos de promoción del profesorado, en paralelo a los méritos provenientes de otras muchas fuentes. Cerrar los ojos a la realidad de unos mecanismos de promoción excesivamente basados en lo particular y efímero y, en algunos casos, en lo subjetivo o en lo que se tiene más próximo conduce al fracaso del sistema. La rigidez que sigue a todo el proceso agrava, aún más, el problema subyacente y está generando una Universidad con una cortedad de miras y una desconexión con la dinámica y los problemas de la Sociedad que pueden traer consecuencias irreparables y de largo plazo.
- Estamos a tiempo de reaccionar, pero algo nos da a entender que deberán afrontarse los problemas reales de la Universidad Española si queremos implantar, a tiempo, los principios de una educación superior bien integrada en el EEES que promueva una Universidad competitiva, implicada en la innovación y en la mejora de la calidad de vida de los humanos, basada en los principios de respeto y justicia de las personas y en los aspectos relacionados con la sostenibilidad y el respeto por la naturaleza.

1.5 OBJETIVOS DEL TÍTULO

Las personas que han obtenido el título de Ingeniería en Informática son profesionales con una formación amplia y sólida que les prepara para dirigir y realizar las tareas de todas las fases del ciclo de vida de sistemas, aplicaciones y

productos que resuelvan problemas de cualquier ámbito de las Tecnologías de la Información y las Comunicaciones, poniendo en práctica su conocimiento científico y los métodos y técnicas propios de la ingeniería.

Con carácter general, el Ingeniero en Informática está capacitado para aprender a conocer, hacer y convivir, en su ámbito personal, profesional y social, de acuerdo con lo recogido en el informe de la UNESCO sobre las perspectivas de la educación en el siglo XXI.

Por su formación, tanto en su base científica como tecnológica, los titulados en Ingeniería en Informática se caracterizan por:

Estar preparados para ejercer la profesión, teniendo una conciencia clara de su dimensión humana, económica, social, legal y ética.

Estar preparados para, a lo largo de su carrera profesional, asumir tareas de responsabilidad en las organizaciones, tanto de contenido técnico como directivo, y de contribuir en la gestión de la información y en la gestión del conocimiento.

Tener las capacidades requeridas en la práctica profesional de la ingeniería: ser capaces de dirigir proyectos, de comunicarse de forma clara y efectiva, de trabajar y conducir equipos multidisciplinares, de adaptarse a los cambios y de aprender autónomamente a lo largo de la vida.

Estar preparados para aprender y utilizar de forma efectiva técnicas y herramientas que surjan en el futuro. Esta versatilidad les hace especialmente valiosos en organizaciones en las que sea necesaria una innovación permanente.

Ser capaces de especificar, diseñar, construir, implantar, verificar, auditar, evaluar y mantener sistemas informáticos que respondan a las necesidades de sus usuarios. Tener la formación de base suficiente para poder continuar estudios, nacionales o internacionales, de Máster y Doctorado.

1.6 LOS PERFILES PROFESIONALES Y SUS COMPETENCIAS

A partir de las diferentes propuestas de currículos internacionales como el de ACM e IEEE, y teniendo en cuenta el carácter permanentemente cambiante intrínseco a los estudios de informática, se han definido tres grandes perfiles que responden a las tendencias profesionales: Desarrollo Software, Sistemas y Gestión y Explotación de Tecnologías de la Información. A continuación se especifican las competencias asociadas a cada uno de estos perfiles:

Competencias del perfil profesional de Desarrollo Software:

- Dirigir y coordinar el proyecto de desarrollo y mantenimiento de aplicaciones, supervisando las funciones y recursos de análisis funcional, orgánico y programación, asegurando la adecuada explotación de las aplicaciones.
- Dominar todas las etapas de la vida de un proyecto (análisis de concepción, análisis técnico, programación, pruebas, documentación y formación de usuarios).

- Dirigir el equipo de trabajo compuesto por Analistas Funcionales, Analistas de aplicaciones y Programadores.
- Controlar y hacer seguimiento de plazos, indicadores económicos y de calidad.
- Supervisar y coordinar el desarrollo completo de aplicaciones y administrar la introducción de los sistemas de gestión.
- Controlar las aplicaciones en explotación, minimizando las consecuencias negativas sobre las operaciones en producción y desarrollo de aplicaciones.
- Analizar y recoger nuevas técnicas y herramientas del mercado, estudiando su viabilidad y necesidad. Posibilidad de contratar recursos externos.
- Controlar y gestionar el desarrollo del Proyecto Informático.
- Redactar, para la Dirección de Informática y para la Dirección del Proyecto, los informes que se precisan para el seguimiento del proyecto.
- Interpretar las especificaciones funcionales encaminadas al desarrollo de las aplicaciones informáticas.
- Realizar el análisis y el diseño detallado de las aplicaciones informáticas.
- Definir la estructura modular y de datos para llevar a cabo las aplicaciones informáticas que cumplan con las especificaciones funcionales y restricciones del lenguaje de programación.
- Definir y describir procedimientos e interfaces de usuario.
- Realizar pruebas que verifiquen la validez funcional, la integridad de los datos y el rendimiento de las aplicaciones informáticas.
- Elaborar y mantener documentación descriptiva de la génesis, producción y operatividad de las aplicaciones informáticas.
- Diseñar servicios de presentación que faciliten la explotación de las aplicaciones.
- Estudiar el sistema actual existente y analizar e idear mejores medios para llevar a cabo los mismos objetivos u otros adicionales.
- Participar en el diseño de nuevos sistemas informáticos como consecuencia de la informatización de áreas de la empresa, que utilizan para el desarrollo de sus tareas métodos y procesos manuales.
- Integrar sistemas informáticos existentes susceptibles de inter-relacionarse.
- Escuchar y asesorar a los Usuarios en la resolución de los problemas que se les plantean con el uso de los sistemas informáticos.
- Asesorar a los programadores en los problemas que se les plantean con la programación de los sistemas.
- Colaborar con los responsables de Estudios y Explotación en la resolución de los fallos que se originen en los Sistemas en Producción.
- Mantenerse al día en Técnicas, Métodos y Herramientas de Análisis y Diseño.

Competencias del perfil profesional de Sistemas:

- Administrar un sistema de bases de datos, interpretando su diseño y estruc-

tura, y realizando la adaptación del modelo a los requerimientos del sistema gestor de bases de datos (SGBD), así como la configuración y administración del mismo a nivel físico y lógico, a fin de asegurar la integridad, disponibilidad y confidencialidad de la información almacenada.

- Desarrollar y construir las bases de datos. Asegurar la coherencia y la adaptación a las necesidades de la empresa.
- Gestionar las autorizaciones de acceso para los usuarios.
- Asegurar el buen funcionamiento de la base y hacer un seguimiento de la utilización de los usuarios a través de las tareas de mirroring, tuning y desdoblamiento.
- Participar en la instalación de las herramientas de Datawarehouse y herramientas de SIAD.
- Responsabilizarse de la integridad de los datos y de la existencia de copias de seguridad.
- Estimar volúmenes de las estructuras de datos, definiendo mecanismos de migración y carga inicial de datos.
- Ocuparse en la producción de la gestión y operativa asociada a las bases de datos y al software en el que están implementadas.
- Diseñar las soluciones informáticas relacionadas con los cambios en los sistemas existentes o con los Nuevos Sistemas.
- Dirigir y asesorar a los Programadores en la realización de los programas.
- Crear los tests de pruebas para verificar que los Sistemas Informáticos cumplen los requisitos y especificaciones de Análisis y Diseño.
- Asesorar a usuarios, programadores y jefe de estudios en la redacción de la documentación de usuario, instalación y explotación.
- Dirigir el arranque o “lanzamiento” de un nuevo sistema.
- Asesorar al Responsable de Estudios en la elaboración de los criterios que permiten la mejor explotación de los nuevos sistemas.
- Ayudar al Área de Estudios en la resolución de los fallos que se producen en los Sistemas en Producción.
- Evaluar nuevos productos informáticos que pueden aportar mejoras tanto en los sistemas existentes, como para el desarrollo de nuevos sistemas.
- Asesorar a los usuarios para utilizar mejor los sistemas existentes.
- Dirigir y coordinar el desarrollo de reuniones relacionadas con temas que afectan a los Sistemas Informáticos.
- Estudiar métodos, técnicas y herramientas de análisis y diseño.
- Estudiar la evolución de las nuevas tecnologías, sobre todo de aquellas que pueden aportar mejoras importantes en los sistemas utilizados en la empresa.
- Planificar, supervisar y coordinar el desarrollo, implantación y mantenimiento de los sistemas operativos, software de mercado y propio, básico o de soporte.

- Definir y actualizar el software básico.
- Analizar y decidir la alternativa óptima de software de mercado a adquirir.
- Diseñar la política de hardware, respecto a adquisiciones, sustituciones...
- Resolver y coordinar las incidencias de los sistemas.
- Dirigir las actividades y recursos técnicos, materiales y los equipos de soporte en materia de sistemas operativos, bases de datos y comunicaciones.
- Establecer políticas de seguridad, técnicas criptográficas, cortafuegos, componentes, configuraciones, productos. Instalar y configurar. Definir reglas de filtrado, conexiones y servicios.
- Dirigir, planificar y coordinar la gestión de la infraestructura de redes y comunicaciones.
- Asegurar la fiabilidad, la coherencia y la evolución de la arquitectura de la red y de las telecomunicaciones utilizadas por los sistemas informáticos de la empresa.
- Gestionar grandes redes corporativas y/o operadores de telecomunicaciones, redes de acceso, redes de transmisión de voz, datos, imágenes, conmutación, gestión de tráfico, así como de todos los aspectos de las redes WAN y las estrategias ligadas a Internet.
- Poner en marcha las redes tanto a nivel material como logístico. Desarrollar y mantener dichas redes. Elegir los elementos hardware y software para la optimización de los servicios de redes de comunicaciones.
- Gestionar las relaciones con los proveedores y negociar los contratos.
- Hacer seguimiento de los presupuestos, los costes y las inversiones.
- Mantener y seguir la evolución de los sistemas de gestión de las telecomunicaciones.
- Enmarcar los participantes internos y externos en los proyectos de telecomunicaciones.
- Escoger y gestionar los contratos con los operadores.
- Encargarse de la dirección técnica y planificación de proyectos de implantación de soluciones y servicios asociados a las redes de comunicaciones.
- Gestionar el conocimiento en inteligencia de negocio en grandes sistemas de redes de comunicaciones en datos y voz (fija y móvil) y sus servicios de valor añadido.
- Gestionar grandes proyectos de cableado de redes, y las infraestructuras parejas, suelos y techos técnicos, electricidad, etc.

Competencias del perfil profesional de Gestión y Explotación de Tecnologías de la Información:

- Poner en marcha la estrategia de la empresa a nivel informático.
- Garantizar las relaciones entre los departamentos de la empresa. Primordial para una buena acogida de las evoluciones del sistema de información.
- Cuidar la coherencia del sistema de información con respecto a la organi-

zación de la empresa y a su evolución. En el marco de la implantación de sistemas integrados (ERP, CRM), garantiza la puesta en marcha de los cambios de procesos decididos por la Dirección General.

- Definir el presupuesto y gestionar los medios materiales y humanos.
- Definir los planes de formación, reciclaje profesional.
- Definir la política informática de la empresa a medio y largo plazo.
- Establecer el alineamiento de los objetivos informáticos con los objetivos de la empresa y velar por su cumplimiento.
- Evaluar los riesgos empresariales asociados a los sistemas informáticos y establecer las orientaciones y directrices para mitigarlos.
- Establecer las directrices sobre las métricas e indicadores que serán utilizados para permitir a la dirección de la empresa la evaluación y el seguimiento de los sistemas informáticos.
- Realizar estudios funcionales y proyectos específicos.
- Concebir las aplicaciones, pilotar la introducción y los parámetros de los sistemas integrados (ERP, CRM).
- Organizar y distribuir el trabajo de los equipos de análisis y de desarrollo (jefes de proyectos, responsables de aplicación).
- Participar en la elaboración de los esquemas directivos y vigilar la coherencia del sistema de información.
- Tomar a su cargo las relaciones con los prestadores del servicio y ciertos proveedores externos.
- Gestionar la conexión entre los departamentos usuarios.
- Vigilar la tecnología y definir las orientaciones técnicas (metodología, calidad, herramientas de desarrollo...).
- Concretar los objetivos de cualquier sistema informático.
- Planificar el desarrollo de un proyecto informático.
- Estudiar la rentabilidad de los sistemas informáticos.
- Estudiar los riesgos de los sistemas informáticos.
- Redactar, para la dirección de la empresa y la dirección de informática, los informes que se precisan para el seguimiento del proyecto.
- Gestionar los clientes o el área geográfica asignada, según la organización de la empresa.
- Analizar los proyectos y las necesidades y proponer soluciones en el plano técnico, humano y financiero.
- Redactar las propuestas comerciales que pueden implicar soluciones estándar o a medida.
- Negociar los contratos.
- Desarrollar el volumen de negocios y asegurar la gestión administrativa.
- Poner en marcha la estrategia comercial elaborada con la dirección.
- Asegurar el seguimiento de los proyectos y su realización.

- Apoyar a los comerciales en las entrevistas con los clientes. Ayuda a definir la necesidad, presentar la solución o el producto en un plano técnico.
- Definir con mayor precisión la necesidad técnica del cliente.
- Elaborar la parte técnica de la propuesta.
- Gestionar la implantación de la solución asumiendo la gestión del proyecto en su integridad, o asegurar una transferencia de competencia hacia los equipos de implantación.
- Asegurar la comunicación entre los usuarios y el departamento de I+D para adaptar o evaluar el producto.
- Definir comercialmente nuevos productos y servicios.
- Coordinar y participar en el proceso de marketing para el desarrollo de productos y servicios.
- Análizar modelos de negocio asociados a la definición de nuevos productos y servicios.
- Colaborar en los estudios de investigación de mercado.
- Colaborar en la definición de la estrategia evolutiva del producto.
- Hacer seguimiento de los procesos y resultados comerciales.
- Definir las normas de desarrollo en colaboración con la Dirección de Informática.
- Motivar y coordinar los equipos de desarrollo en el marco de aplicación de las normas y métodos en vigor.
- Hacer labores de intermediario y consejero de cara a los desarrollos que se realicen.
- Asegurar la definición de las directrices de calidad, su aplicación así como la estandarización. Responsabilizarse de la adecuación entre los desarrollos realizados y las directrices establecidas.
- Poner en marcha los procedimientos de prueba y de control de calidad.
- Asegurar la coherencia y la coordinación de su trayectoria con la política global de la empresa.
- Tomar a su cargo la campaña de las pruebas de cara al conjunto de los usuarios finales.
- Participar en la distribución de las ediciones originales de las aplicaciones y de los documentos a las entidades de producción garantizando un alto nivel de calidad.
- Garantizar una calidad permanente a través de los procedimientos y de las herramientas.
- Apoyar las demandas cotidianas de los usuarios.
- Asegurar el buen funcionamiento físico de los sistemas informáticos (automatización de copias de seguridad y la seguridad de datos).
- Administrar las incidencias y asegurar las soluciones.
- Organizar y supervisar el trabajo de su equipo, de los técnicos de mantenimiento y los ingenieros de sistemas y redes.

- Administrar tanto los abastecimientos como las relaciones con los proveedores y los constructores.
- Responsabilizarse del buen funcionamiento del sistema informático y sus resultados. Colaborar con el responsable de desarrollo para que el sistema de arquitectura pueda responder a las exigencias de las aplicaciones desarrolladas.
- Definir los procesos, los documentos y ejecutar su control.

1.7 LAS TITULACIONES DE INFORMÁTICA EN LA EPS

En la Escuela Politécnica Superior de la Universidad de Alicante los estudios de Informática se estructuran, por una parte, de una manera terminal con las dos Ingenierías Técnicas (Gestión y Sistemas) y, por otra, cíclicamente con la Ingeniería Informática.

Al primer ciclo se puede acceder desde COU, FP II (ramas: Administrativa, Informática, Electricidad o Electrónica), otras titulaciones superiores o de grado medio y por examen de acceso a la Universidad para mayores de 25 años.

Los objetivos generales del primer ciclo son dos: la formación de profesionales preparados para ejercer su actividad en áreas relacionadas con Informática y, por otro lado, la formación de estudiantes capacitados para continuar sus estudios en segundo ciclo, y probablemente en tercer ciclo. Esta doble finalidad supone encontrar una solución de compromiso entre la formación teórica y práctica que requiere un Ingeniero Técnico en Informática y la sólida base que necesitan aquéllos que continúan sus estudios en los siguientes ciclos.

En cualquier caso, se debe tener siempre como prioridad el proporcionar a los estudiantes los conocimientos, habilidades, experiencia y actitudes necesarias para poder proyectar los estudios realizados al ejercicio de una profesión que deberá adaptarse a las distintas evoluciones que constantemente se producen en el campo de la Informática.

La superación del primer ciclo da derecho a la obtención del título de Ingeniero Técnico en Informática en una de las dos especialidades que se imparten (Sistemas Físicos y Gestión).

La superación del segundo ciclo da derecho a la obtención del título de Ingeniero en Informática. El objetivo general del segundo ciclo es completar y ampliar los conocimientos adquiridos en el primero, profundizando con una orientación más teórica en algunos conceptos ya estudiados anteriormente e introduciendo algunos temas nuevos más específicos.

En la actualidad existe una coexistencia de los planes de estudio de 1993 y los nuevos planes de estudios aprobados en Junta de Centro de la Escuela Politécnica Superior el día 19 de febrero de 2001 y aprobados en la Junta de Gobierno celebrada el 8 de marzo de 2001 (B.O.E. de 25 de septiembre de 2001). La implantación de los planes de estudio del 2001 se ha realizado de forma gradual.

1.7.1 Plan de Estudios del Título de Ingeniero en Informática (II)

El Plan de Estudios de Ingeniero en Informática II tiene una duración de cinco años estructurados en dos ciclos, un primer ciclo de tres años (no terminal) y otro segundo de dos, con una carga lectiva total de 364,5 créditos, cuya distribución se muestra en el siguiente cuadro:

INGENIERÍA EN INFORMÁTICA			
CARACTERÍSTICAS	Primer ciclo	Segundo ciclo	TOTAL
Años	3	2	5
Créditos de materias troncales	109,5	786,5	186
Créditos de materias obligatorias	55,5	13,5	69
Créditos optativos	24	48	72
Créditos de libre elección	19,5	18	37,5
Total	208,5	156	364,5

Cuadro 1. Distribución de créditos del Título de Ingeniero en Informática

Los siguientes cuadros presentan las asignaturas troncales y obligatorias correspondientes al Título de Ingeniero en Informática

INGENIERÍA INFORMÁTICA				
Troncalidad. Primer Ciclo				
MATERIA TRONCAL	Créd.	Curso	ASIGNATURA	Créd.
Estadística	6	1	Estadística	6
Estructura de Datos y de la Información	18	2	Programación y Estructuras de Datos	9
		2	Bases de datos I	9
Estructura y Tecnología de Computadores	18	1	Informática Básica	12
		2	Estructuras de computadores	6
Fundamentos Físicos de la Informática	10,5	1	Fundamentos Físicos de la Informática	10,5
Fundamentos Matemáticos de la Informática	21	1	Álgebra	6
		1	Cálculo Infinitesimal	9
		1	Matemática Discreta	6
Metodología y Tecnología de la Programación	18	1	Fundamentos de la Programación I	6
		2	Fundamentos de la Programación II	6
		3	Diseño y Análisis de Algoritmos	6
Sistemas Operativos	9	2	Sistemas Operativos I	4,5
		3	Sistemas Operativos II	4,5
Teoría de Autómatas y Lenguajes Formales	9	2	Lenguajes, gramáticas y Autómatas	4,5
		2	Computabilidad	4,5

Cuadro 2. Asignaturas troncales del primer ciclo del Título de Ingeniero en Informática

INGENIERÍA INFORMÁTICA Troncalidad. Segundo Ciclo				
MATERIA TRONCAL	Créd.	Curso	ASIGNATURA	Créd.
Arquitectura e Ingeniería de Computadores	12	4	Arquitectura e Ingeniería de computadores	9
Ingeniería del Software	18	4	Análisis y Especificación de S.I.	6
		4	Ingeniería del software I	6
		5	Ingeniería del software II	6
Inteligencia Artificial e Ingeniería del Conocimiento	9	4	Fundamentos de la inteligencia artificial	4,5
		4	Técnicas de la inteligencia artificial	4,5
Processadores de Lenguaje	9	4	Procesadores del Lenguaje	9
Redes	13,5	4	Redes	7,5
		4	Sistemas de Transporte de Datos	6
Sistemas Informáticos	15	5	Sistemas Informáticos	15

Cuadro 3. *Asignaturas troncales del segundo ciclo del Título de Ingeniero en Informática*

Las asignaturas obligatorias para cubrir los 55,5 créditos del primer ciclo y los 13,5 créditos del segundo ciclo de la titulación de Ingeniería en Informática se muestran en los siguientes cuadros.

INGENIERÍA INFORMÁTICA Obligatoriedad propia de la UA		
Curso	ASIGNATURA	Créd.
1	Lógica Computacional	6
1	Lenguajes y Paradigmas de la Programación	6
2	Programación Orientada a Objetos	4,5
2	Herramientas de Programación	6
3	Gráficos por Computador	4,5
3	Bases de Datos II	6
3	Diseño y Programación Avanzada de Aplicaciones	4,5
3	Fundamentos de Arquitectura de Computadores	6
3	Sistemas de Información en la Empresa I	6
3	Sistemas de Información en la Empresa II	6
4	Algoritmia Avanzada	4,5
5	Sistemas Operativos en Red	9

Cuadro 4. *Asignaturas obligatorias del Título de Ingeniero en Informática*

INGENIERÍA EN INFORMÁTICA					
Ordenación de asignaturas troncales y obligatorias. Primer y segundo ciclo					
Curso	Carác.	ASIGNATURA	Créditos		
			Total	Teoría	Prác.
1º	C	Álgebra	6	3	3
1º	A	Cálculo Infinitesimal	9	4,5	4,5
1º	C	Estadística I	6	3	3
1º	A	Fundamentos Físicos de la Informática	10,5	6	4,5
1º	C	Fundamentos de Programación I	6	3	3
1º	C	Fundamentos de Programación II	6	3	3
1º	A	Informática Básica	12	6	6
1º	C	Lógica Computacional	6	3	3
1º	C	Matemática Discreta	6	3	3
Resumen curso 1º			67,5	34,5	33
2º	A	Bases de Datos I	9	6	3
2º	C	Computabilidad	4,5	2,25	2,25
2º	C	Estructura de Computadores	6	3	3
2º	C	Herramientas de Programación	6	3	3
2º	C	Lenguajes, Gramáticas y Autómatas	4,5	3	1,5
2º	C	Lenguajes y Paradigmas de Programación	6	3	3
2º	C	Programación Orientada a Objetos	4,5	2,25	2,25
2º	A	Programación y Estructuras de Datos	9	4,5	4,5
2º	C	Sistemas Operativos I	4,5	2,25	2,25
2º	-	<i>Asignaturas optativas</i>	6	-	-
Resumen curso 2º			60	-	-
3º	C	Bases de Datos II	6	3	3
3º	C	Diseño y Análisis de Algoritmos	6	3	3
3º	C	Diseño y Programación Avanzada de Aplicaciones	4,5	2,25	2,25
3º	C	Fundamentos de Arquitectura de Computadores	6	3	3
3º	C	Gráficos por Computador	4,5	2,25	2,25
3º	C	Sistemas de Información de la Empresa I	6	3	3
3º	C	Sistemas de Información de la Empresa II	6	3	3
3º	C	Sistemas Operativos II	4,5	2,25	2,25
3º	-	<i>Asignaturas optativas</i>	18	-	-
2º y 3º	-	<i>Asignaturas de libre configuración</i>	19,5	-	-
Resumen curso 3º			81	-	-
4º	C	Algoritmia Avanzada	4,5	2,25	2,25
4º	C	Análisis y Especificación de Sistemas de Información	6	3	3
4º	A	Arquitecturas e Ingeniería de Computadores	12	6	6
4º	C	Fundamentos de Inteligencia Artificial	4,5	2,25	2,25
4º	C	Ingeniería del Software I	6	3	3
4º	C	Técnicas de Inteligencia Artificial	4,5	2,25	2,25
4º	A	Procesadores de Lenguajes	9	6	3
4º	C	Redes	7,5	3	4,5
4º	C	Sistemas de Transporte de Datos	6	3	3
4º	-	<i>Asignaturas optativas</i>	6	-	-
Resumen curso 4º			66	-	-

Curso	Carác.	ASIGNATURA	Créditos		
			Total	Teoría	Prác.
5º	C	Ingeniería del Software II	6	3	3
5º	A	Sistemas Informáticos	15	0	15
5º	A	Sistemas Operativos en Red	9	4,5	4,5
5º	-	<i>Asignaturas optativas</i>	42	-	-
4º y 5º	-	<i>Asignaturas de libre configuración</i>	18	-	-
<i>Resumen curso 5º</i>			90	-	-

Cuadro 5. Ordenación de asignaturas troncales y obligatorias del Título de Ingeniero en Informática por cursos y créditos

Cuadro 5 muestra la ordenación por cursos de las asignaturas de carácter troncal y obligatorio de la Ingeniería Informática correspondientes al primer y segundo ciclo. Además se muestran los créditos que deben cubrirse con materias optativas y de libre configuración por cada uno de los cursos hasta el máximo establecido.

1.7.2 Plan de Estudios del Título de Ingeniero Técnico en Informática de Gestión (ITIG)

El Plan de Estudios de ITIG tiene una duración de 3 años con una carga lectiva total de doscientos veinticinco (225) créditos, cuya distribución se muestra en el siguiente cuadro:

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN	
CARACTERÍSTICAS	
Años	3
Créditos de materias troncales	126
Créditos de materias obligatorias	37,5
Créditos optativos	26
Créditos de libre elección	25,5
Total	225

Cuadro 6. Distribución de créditos del Título de Ingeniero Técnico en Informática de Gestión

En el siguiente cuadro se presentan las asignaturas troncales definidas por la Universidad de Alicante para cubrir las materias troncales de las directrices propias de la titulación. En ella se muestran los créditos totales de la materia troncal según las directrices, y además los créditos de cada una de las asignaturas troncales asignados por la Universidad para cubrir esa materia. Entre paréntesis se indican los créditos adicionales que se han aplicado por el plan de estudios.

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN				
Troncalidad				
MATERIA TRONCAL	Créd.	Curso	ASIGNATURA	Créd.
Estadística	10,5	1	Estadística	6
		2	Ampliación de Estadística	4,5
Estructura de Datos y de la Información	18	2	Programación y Estructuras de Datos	9
		2	Bases de datos I	9
Estructura y Tecnología de Computadores	19,5	1	Informática Básica	12
		2	Arquitectura de computadores	7,5
Fundamentos Matemáticos de la Informática	21	1	Álgebra	6
		1	Cálculo Infinitesimal	9
		1	Matemática Discreta	6
Metodología y Tecnología de la Programación	18	1	Fundamentos de la Programación I	6
		1	Fundamentos de la Programación II	6
		3	Diseño y Análisis de Algoritmos	6
Sistemas Operativos	9	3	Sistemas Operativos	9
Ingeniería del Software	12	3	Metodología de Análisis de SI	6
		3	Diseño de Sistemas Software	6
Técnicas de Organización y Gestión Empresarial	18	2	Sistema Económico y Empresa	9
		2	Técnicas de Administración y Contables	9

Cuadro 7. *Asignaturas troncales del Título de Ingeniero Técnico en Informática de Gestión*

Además, el conjunto de asignaturas obligatorias, que junto con las troncales constituyen la obligatoriedad de la titulación, se muestran en el siguiente cuadro:

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN		
Obligatoriedad propia de la UA		
Curso	ASIGNATURA	Créd.
1	Lógica Computacional	6
1	Relaciones Jurídicas Básicas	4,5
2	Programación Orientada a Objetos	4,5
3	Gestión de Proyectos Software	4,5
3	Bases de Datos II	6
3	Diseño y Programación Avanzada de Aplicaciones	4,5
3	Redes	7,5

Cuadro 8. *Asignaturas obligatorias del Título de Ingeniero Técnico en Informática de Gestión*

Por otro lado, el siguiente cuadro presenta la ordenación por cursos de las asignaturas de carácter troncal y obligatorio de la titulación, más los créditos que se deben cubrir de materias optativas y de libre configuración por cada curso.

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN					
Ordenación de asignaturas troncales y obligatorias					
Curso	Carác.	ASIGNATURA	Créditos		
			Total	Teoría	Prác.
1º	C	Álgebra	6	3	3
1º	A	Cálculo Infinitesimal	9	4,5	4,5
1º	C	Estadística	6	3	3
1º	C	Relaciones Jurídicas Básicas	4,5	4,5	0
1º	C	Fundamentos de Programación I	6	3	3
1º	C	Fundamentos de Programación II	6	3	3
1º	A	Informática Básica	12	6	6
1º	C	Lógica Computacional	6	3	3
1º	C	Matemática Discreta	6	3	3
<i>Resumen curso 1º</i>			61,5	33	28,5
2º	C	Ampliación de Estadística	4,5	1,5	3
2º	C	Arquitectura de Computadores	7,5	4,5	3
2º	A	Bases de Datos I	9	6	3
2º	C	Programación Orientada a Objetos	4,5	2,25	2,25
2º	A	Programación y Estructura de Datos	9	4,5	4,5
2º	C	Sistema Económico y Empresa	9	6	3
2º	C	Técnicas de Administración y Contables	9	6	3
2º	-	<i>Asignaturas optativas</i>	18	-	-
<i>Resumen curso 2º</i>			70,5	-	-
3º	C	Bases de Datos II	6	3	3
3º	C	Diseño de Sistemas Software	6	3	3
3º	C	Diseño y Análisis de Algoritmos	6	3	3
3º	C	Diseño y Programación Avanzada de Aplicaciones	4,5	2,25	2,25
3º	C	Gestión de Proyectos Software	4,5	2,25	2,25
3º	C	Metodología de Análisis de Sistemas de Información	6	3	3
3º	C	Redes	7,5	3	4,5
3º	A	Sistemas Operativos	9	4,5	4,5
3º	-	<i>Asignaturas optativas</i>	18	-	-
2º y 3º	-	<i>Asignaturas de libre configuración</i>	25,5	-	-
<i>Resumen curso 3º</i>			93	-	-

Cuadro 9. Ordenación de asignaturas troncales y obligatorias del
Título de Ingeniero Técnico en Informática de Gestión por cursos y créditos

1.7.3 Plan de Estudios del Título de Ingeniero Técnico de Informática de Sistemas (ITIS)

El Plan de Estudios de ITIS tiene una duración de 3 años, con una carga lectiva total de doscientos veinticinco (225) créditos, cuya distribución se muestra en el siguiente cuadro:

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS	
CARACTERÍSTICAS	
Años	3
Créditos de materias troncales	114
Créditos de materias obligatorias	46,5
Créditos optativos	42
Créditos de libre elección	25,5
Total	225

Cuadro 10. *Distribución de créditos del Título de Ingeniero Técnico en Informática de Sistemas*

En el siguiente cuadro se presentan las asignaturas troncales definidas por la Universidad de Alicante para cubrir las materias troncales de las directrices propias de la titulación. En ella se muestran los créditos totales de la materia troncal según las directrices, y además los créditos de cada una de las asignaturas troncales asignados por la Universidad para cubrir esa materia. Entre paréntesis se indican los créditos adicionales que se han aplicado por el plan de estudios.

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS				
Troncalidad				
MATERIA TRONCAL	Créd.	Curso	ASIGNATURA	Créd.
Estadística	10,5	1	Estadística	6
Estructura de Datos y de la Información	18	2	Programación y Estructuras de Datos	9
		2	Bases de datos I	9
Estructura y Tecnología de Computadores	19,5	1	Informática Básica	12
		2	Estructura de computadores	6
Fundamentos Físicos de la Informática	21	1	Fundamentos Físicos de la Informática	6
Fundamentos Matemáticos de la Informática	18	1	Álgebra	6
		1	Cálculo Infinitesimal	9
		1	Matemática Discreta	6
Metodología y Tecnología de la Programación	9	1	Fundamentos de la Programación I	6
		1	Fundamentos de la Programación II	6
		1	Diseño y Análisis de Algoritmos	6

MATERIA TRONCAL	Créd.	Curso	ASIGNATURA	Créd.
Redes	7,5	3	Redes	7,5
Sistemas Operativos	12	2	Sistemas Operativos	6
Teoría de Autómatas y Lenguajes Formales	18	2 2	Lenguajes, Gramáticas y Autómatas Computabilidad	4,5 4,5

Cuadro 11. *Asignaturas troncales del Título de Ingeniero Técnico en Informática de Gestión*

Además, el conjunto de asignaturas obligatorias, que junto con las troncales constituyen la obligatoriedad de la titulación, se muestran en el siguiente cuadro:

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS Obligatoriedad propia de la UA		
Curso	ASIGNATURA	Créd.
1	Lógica Computacional	6
2	Programación Orientada a Objetos	4,5
3	Programación de Sistemas de Tiempo Real	6
3	Arquitectura de Computadores	6
3	Administración de Sistemas Operativos en Red	4,5
3	Periféricos	6
3	Sistemas de Transporte de Datos	6
3	Señales y Sistemas	7,5

Cuadro 12. *Asignaturas obligatorias del Título de Ingeniero Técnico en Informática de Sistemas*

Por otro lado, el siguiente cuadro presenta la ordenación por cursos de las asignaturas de carácter troncal y obligatorio de la titulación, más los créditos que se deben cubrir de materias optativas y de libre configuración por cada curso.

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS Ordenación de asignaturas troncales y obligatorias					
Curso	Carác.	ASIGNATURA	Créditos		
			Total	Teoría	Prác.
1º	C	Álgebra	6	3	3
1º	A	Cálculo Infinitesimal	9	4,5	4,5
1º	C	Estadística	6	3	3
1º	A	Fundamentos Físicos de la Informática	10,5	6	4,5
1º	C	Fundamentos de Programación I	6	3	3
1º	C	Fundamentos de Programación II	6	3	3
1º	A	Informática Básica	12	6	6
1º	C	Lógica Computacional	6	3	3
1º	C	Matemática Discreta	6	3	3
Resumen curso 1º			67,5	34,5	33

Curso	Carác.	ASIGNATURA	Créditos		
			Total	Teoría	Prác.
2º	C	Estructuras de Computadores	6	3	3
2º	A	Bases de Datos I	9	6	3
2º	C	Lenguajes, Gramáticas y Autómatas	4,5	3	1,5
2º	C	Computabilidad	4,5	2,25	2,25
2º	C	Sistemas Operativos	6	3	3
2º	C	Programación Orientada a Objetos	4,5	2,25	2,25
2º	A	Programación y Estructura de Datos	9	4,5	4,5
2º	-	<i>Asignaturas optativas</i>	18	-	-
Resumen curso 2º			61,5	-	-
3º	C	Arquitectura de Computadores	6	3	3
3º	C	Diseño y Análisis de Algoritmos	6	3	3
3º	C	Redes	7,5	3	4,5
3º	C	Programación de Sistemas de Tiempo Real	6	3	3
3º	C	Señales y sistemas	7,5	6	1,5
3º	C	Periféricos	6	3	3
3º	C	Administración de Sistemas Operativos en Red	4,5	2,25	2,25
3º	C	Sistemas de Transporte de Datos	6	3	3
3º	-	<i>Asignaturas optativas</i>	24	-	-
2º y 3º	-	<i>Asignaturas de libre configuración</i>	22,5	-	-
Resumen curso 3º			96	-	-

Cuadro 13. Ordenación de asignaturas troncales y obligatorias del
Título de Ingeniero Técnico en Informática de Sistemas por cursos y créditos

1.8 PROGRAMA DE REDES DE INVESTIGACIÓN EN DOCENCIA

La definición del EEES implica una profunda reestructuración de la docencia universitaria en lo concerniente al diseño curricular, a las estrategias de aprendizaje y a los modelos de evaluación. Este reto únicamente podrá ser asumido con el desarrollo de las adecuadas disposiciones y capacidades profesionales en la comunidad universitaria.

Con la finalidad de apoyar la nueva oportunidad de renovación docente, el Vicerrectorado de Calidad y Armonización Europea y el Instituto de Ciencias de la Educación (ICE) de la Universidad de Alicante convocan la edición 2005-2006 del Programa de Redes de Investigación en Docencia Universitaria, con un alto énfasis en la constitución de equipos que impliquen al profesorado de cursos completos.

Esta edición es la quinta desde que, durante el curso 2001-2002, se definió como objetivo fundamental apoyar la investigación docente. Se han constituido redes docentes, formadas por grupos de profesores implicados en el desarrollo profesional de su docencia a través de proyectos de investigación curricular. Desde esa primera convocatoria, la producción de los docentes e investigadores implicados en este proceso ha sido más que prolífica, dando como resultado una

gran cantidad de publicaciones materializadas en los diferentes volúmenes editados por el ICE y extendidos a las jornadas que, anualmente, se celebran en el marco de estas actividades.

Algunas de estas publicaciones, de las que es posible encontrar información detallada y actualizada en <http://www.ua.es/ice/redes/ediciones.html> son:

- Bernabeu Pastor, G.; Sauleda Parés, N. (Eds.) (2004) *Espacios de Investigación en la Profesionalización Docente Universitaria II*. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Bernabeu Pastor, G.; Sauleda Parés, N. (Eds.) (2004) *Investigar el Espacio Europeo de Educación Superior*. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Frau, M. J. y Sauleda, N. (Eds.) (2006) *El modelo docente en la Universidad: Investigaciones Colegiadas* Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Frau, M. J. y Sauleda, N. (Eds.) (2005) *Investigar en diseño curricular. Redes de docencia en el Espacio Europeo de Educación Superior*. Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Llopis Pascual, F. y Llorens Largo, F. (Eds.) (2005) *Adecuación del primer curso de los estudios de informática al Espacio Europeo de Educación Superior*. Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Serie Docencia Universitaria – EEES. Universidad de Alicante.
- Martínez Ruiz, M. A. (2002) *Investigar en Docencia Universitaria*. Redes de Colaboración para el Aprendizaje. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante: Marfil.
- Martínez Ruiz, M.A.; Carrasco Embuena, V. (Coords.) (2003) *Investigar Colaborativamente en Docencia Universitaria* Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Martínez Ruiz, M.A.; Carrasco Embuena, V. (Eds.) (2004) *Espacios de Participación en la Investigación del Aprendizaje Universitario I*. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Martínez Ruiz, M.A.; Carrasco Embuena, V. (Eds.) (2004) *Hacia la Europa del Conocimiento. II Jornadas de Redes de Investigación en Docencia Universitaria*. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.

- Martínez, M. A. y Carrasco, V. (Eds.) (2006) *La construcción colegiada del modelo docente universitario del siglo XXI. Redes de Investigación docente en el Espacio Europeo de Educación Superior* Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Martínez, M. A.; Carrasco, V. (Eds.) (2005) *III Jornadas de Redes de Investigación en Docencia Universitaria. La configuración del Espacio Europeo de Educación Superior*. Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Martínez, M.A; Castejón, J.L.; Roig, R.I. (Coords) (2003) *Investigación en Docencia Universitaria. Redes de Colaboración para el Análisis de la Práctica Docente* Vicerrectorado de Estudios e Innovación Educativa. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Martínez, M.A; Castejón, J.L; Roig, R.I. (Coords) (2003) *Proyecto de Investigación Docente*. I Programa de Redes de Investigación en Docencia Universitaria. Vicerrectorado de Estudio e Innovación Educativa. Instituto de Ciencias de la Educación. Universidad de Alicante.
- Rico Vercher, M.; Rico Pérez, C. (2004) *El Portfolio Discente*. Ed. Marfil/Serie Docencia Universitaria- EEES. Vicerrectorado de Convergencia Europea y Calidad. Instituto de Ciencias de la Educación. Universidad de Alicante.

El Programa de Redes ha enmarcado sus acciones formativas e investigadoras en el programa de indicadores de calidad universitaria, especialmente, en el referido al índice de participación del profesorado universitario en acciones de formación docente.

Colaborar en la consecución de este indicador mediante el ofrecimiento de oportunidades de participación del profesorado en la investigación de su propia práctica docente y tutorial, indudablemente se refleja en la implicación del profesorado sobre el desarrollo del aprendizaje del alumno. De modo que asimismo podemos contribuir con este Programa al desarrollo de los indicadores siguientes:

- El índice de satisfacción de los alumnos con la docencia
- La tasa de rendimiento discente

Todo esto encuentra su sentido en el diseño de acciones de investigación en el desarrollo de la docencia y tutoría universitaria, no cómo un trabajo individual, sino como una tarea de equipos colaborativos de trabajo, reflexión y debate.

El resultado de la convocatoria de este programa ha supuesto la conformación de un espacio de investigación de la docencia, a partir de una estructura y con una dinámica articulada en redes colaborativas. Las redes han desarrollado procesos de investigación en diseño curricular, implementación de metodologías docentes, estrategias de aprendizaje y modelos de evaluación en concordancia con los criterios que marca el proceso de convergencia hacia el EEES, permi-

tiendo iniciar experiencias investigadoras facilitadoras del proceso de implantación de los créditos ECTS.

Este trabajo se enmarca en la tercera modalidad de la convocatoria del Programa de Redes de Investigación en Docencia universitaria, curso 2005-2006: Redes singulares de investigación en docencia universitaria centradas en 2do. y/o 3er. curso de una titulación, que preferiblemente hayan experimentado en primer curso.

2. PROYECTO DE INVESTIGACIÓN DOCENTE

La Escuela Politécnica Superior de la Universidad de Alicante, en su afán de adaptación a los nuevos parámetros del Espacio Europeo de Educación Superior, ha realizado un gran esfuerzo en la elaboración de materiales y objetivos claramente ajustados a estas nuevas exigencias.

Durante los cursos académicos 2003-2004 y 2004-2005 elaboró un estudio sobre esta adaptación en las asignaturas de primer curso de la Ingeniería Informática, estudio que culminó con la publicación de la guía docente general y específica: “Adecuación del primer curso de los estudios de informática al Espacio Europeo de Educación Superior”.

Además de este esfuerzo conjunto, multitud de asignaturas de ambos ciclos formativos en las titulaciones de informática de la Universidad de Alicante, y bajo el amparo de las redes de investigación propuestas cada año por el Instituto de Ciencias de la Educación de la Universidad de Alicante, han realizado también esfuerzos individuales o colectivos para acelerar este proceso de adaptación.

Con el fin de dar continuidad a este trabajo, este proyecto pretende realizar, a lo largo de los cursos 2005-2006 y 2006-2007, un estudio práctico y un análisis de los resultados para estimar el valor de los créditos europeos en el segundo curso de las titulaciones de informática impartidas en la Escuela Politécnica Superior de la Universidad de Alicante. Para ello se plantea un estudio comparativo:

- del esfuerzo que debe realizar el estudiante, estimado a priori por el profesor en la programación y planificación de su asignatura (proyecto preacción desarrollado a lo largo del curso 2005-2006).
- y del esfuerzo real dedicado por el estudiante para poder alcanzar los objetivos de la asignatura, estimado a posteriori (proyecto acción con la creación de un grupo piloto para el curso 2006-2007).

Así, el objetivo principal de este grupo de trabajo ha sido la puesta en marcha el curso 2006-2007 de un grupo piloto de segundo curso de Ingeniería Informática en el que se aplique el sistema de créditos ECTS. Para ello a lo largo del curso 2005-2006 se ha propuesto un estudio que contenga recomendaciones sobre:

- Valoración en créditos ECTS de las asignaturas de segundo de Ingeniería Informática.

- Diseño de guías docentes de las asignaturas de segundo de Ingeniería Informática para adecuarse al número de créditos ECTS de un curso.
- Adaptación de la metodología educativa de estas asignaturas a un sistema con uso de créditos ECTS.
- Evaluación de las distintas necesidades para la puesta en marcha de un grupo piloto el curso 2006-2007 en el que se apliquen estas propuestas.

El grupo de trabajo ha estado formado por:

- Un profesor representante de cada una de las asignaturas (subredes) troncales y obligatorias de segundo curso de las tres Ingenierías en Informática.
- El dinamizador de las redes de Informática en la Escuela Politécnica Superior, que actuará de coordinador del grupo de trabajo.
- Además, cada responsable de subred ha organizado su propio grupo de trabajo con el resto de sus miembros para la elaboración de contenidos y documentación.

2.1 TITULACIONES Y ASIGNATURAS VINCULADAS A LA INVESTIGACIÓN DOCENTE

Una de las características más ambiciosas del presente trabajo es que, con el objetivo de cubrir de manera más global la docencia en segundo curso, se han integrado las asignaturas que conforman la troncalidad y obligatoriedad del segundo curso en las tres titulaciones en Informática¹ (Ingeniería Informática, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas). Estas asignaturas, clasificadas por áreas docentes, son:

Área de Arquitectura y Tecnología de Computadores

- ARQUITECTURA DE COMPUTADORES (7,5 créditos)
- ESTRUCTURAS DE COMPUTADORES (6 créditos)
- SISTEMAS OPERATIVOS (6 créditos)
- SISTEMAS OPERATIVOS I (4,5 créditos)

Área de Ciencia de la Computación e Inteligencia Artificial

- COMPUTABILIDAD (4,5 créditos)
- LENGUAJES Y PARADIGMAS DE PROGRAMACIÓN (6 créditos)
- AMPLIACIÓN DE ESTADÍSTICA (6 créditos)

Área de Economía Financiera y Contabilidad

- SISTEMA ECONÓMICO Y EMPRESA (9 créditos)
- TÉCNICAS DE ADMINISTRACIÓN Y CONTABLES (9 créditos)

1 Si bien es cierto que las asignaturas del área docente de Economía Financiera y Contabilidad, por diferentes razones organizativas no pudieron estar finalmente en la presente edición. Confiamos que para futuras convocatorias y proyectos pueda completarse la oferta.

Área de Lenguajes y Sistemas Informáticos

- BASES DE DATOS I (9 créditos)
- LENGUAJES, GRAMÁTICAS Y AUTÓMATAS (4,5 créditos)
- PROGRAMACIÓN Y ESTRUCTURAS DE DATOS (9 créditos)
- HERRAMIENTAS DE PROGRAMACIÓN (6 créditos)
- PROGRAMACIÓN ORIENTADA A OBJETOS (4,5 créditos)

Área de Derecho Administrativo

- RELACIONES JURÍDICAS BÁSICAS (4,5 créditos)²

El siguiente cuadro muestra los detalles de estas asignaturas referidos a su carácter troncal u obligatorio, el número de créditos que representan en el plan de estudios así como el departamento encargado de su impartición.

Asignatura	Tipo	Período	Créditos	Depto.*
Ampliación de Estadística	Troncal	2º	4,5	DCCIA
Arquitectura de Computadores	Troncal	2º	7,5	DTIC
Bases de Datos I	Troncal	Anual	9	DLSI
Computabilidad	Troncal	2º	4,5	DCCIA
Estructuras de Computadores	Troncal	1º	6	DTIC
Herramientas de Programación	Obligatoria	1º	6	DLSI
Lenguajes y Paradigmas de Programación	Obligatoria	2º	6	DCCIA
Lenguajes, Gramáticas y Autómatas	Troncal	1º	4,5	DLSI
Programación Orientada a Objetos	Obligatoria	1º	4,5	DLSI
Programación y Estructuras de Datos	Troncal	Anual	9	DLSI
Sistema Económico y Empresa	Troncal	1º	9	DEFM
Sistemas Operativos	Troncal	2º	6	DTIC
Sistemas Operativos I	Troncal	2º	4,5	DTIC
Técnicas de Administración y Contables	Troncal	2º	9	DEFM
Relaciones jurídicas básicas (1º)	Obligatoria	1º	4,5	DEJE

* DCCIA: Departamento de Ciencia de la Computación e Inteligencia Artificial
 DEJE: Departamento de Estudios Jurídicos del Estado
 DTIC: Departamento de Tecnología Informática y Computación
 DEFM: Departamento de Economía Financiera, Contabilidad y Marketing
 DLSI: Departamento de Lenguajes y Sistemas Informáticos

Cuadro 14. *Asignaturas troncales y obligatorias implicadas en el proyecto*

2 Es necesario mencionar que, si bien la asignatura “Relaciones Jurídicas Básicas” pertenece al primer curso de la Ingeniería Técnica en Informática de Gestión, ha sido considerada en esta red para completar las asignaturas de primer curso, ya que ésta no fue incluida en la propuesta anterior por estar éste centrado en la Ingeniería Informática. Para la integración de esta asignatura en el marco global de trabajo, y dado que no es posible incluirla en el proyecto piloto de implementación del primer curso de este año por razones evidentes, esta asignatura formará parte de este grupo de trabajo con el fin de obtener la madurez necesaria para su implantación durante el siguiente periodo académico.

3. PLAN DE ADECUACIÓN AL EEES

El nuevo sistema de créditos ECTS supone un cambio conceptual muy importante en la planificación docente. De esta forma, el contexto actual en el que los créditos representan exclusivamente los procesos docentes presenciales, se debe sustituir por un nuevo contexto, en el que se consideren tanto las horas presenciales como las no presenciales que conducen al alumno a la consecución de sus objetivos de aprendizaje.

Siguiendo la estructura marcada por el trabajo realizado con las asignaturas de primer curso, se ha definido un escenario de trabajo para la adecuación de los actuales planes de estudio al nuevo contexto ECTS. Este escenario contempla la dedicación en las diferentes dimensiones del aprendizaje, tanto en contenidos teóricos como prácticos, esto es, el tiempo de clase, el de preparación de la materia fuera de clase, los procesos tutoriales o las actividades en pequeños grupos.

En la adaptación de estos escenarios actuales, se ha tenido en cuenta la correspondencia de un crédito (actual) a 10 horas de clase presencial y la asignación de un total de 60 créditos ECTS anuales con 25 a 30 horas por crédito, lo que da como resultado entre 1500 y 1800 horas al año de dedicación total.

A continuación se mostrarán los escenarios actuales y adaptados al espacio europeo para cada una de las tres titulaciones de informática.

3.1 ADAPTACIÓN ECTS EN INGENIERÍA INFORMÁTICA

Para la adaptación del escenario actual al nuevo contexto europeo se ha partido de la carga docente de cada una de las asignaturas de la titulación junto con la carga asignada en función del número de grupos. En ambos casos se ha tenido en cuenta la diferenciación clara entre teoría y práctica.

Para el cálculo del número de grupos teóricos y prácticos, se ha tomado como referencia el número de alumnos matriculados en el curso 2005-2006 (ver Cuadro 15) y se ha realizado la media aritmética.

asignatura	alumnos
BD1	163
COMP	182
EC	212
HP	239
LGA	177
LPP	158
PED	199
POO	167
SOI	135
media	181,3

Cuadro 15. Ingeniería Informática: alumnos matriculados por asignatura (curso 2005-2006)

Despreciando los decimales y tomando como media 181 alumnos se han establecido un total de 3 grupos de teoría. Dada la experimentalidad de la titulación, se han supuesto 9 grupos de prácticas. Para los cálculos de grupos pequeños se ha dividido cada grupo de prácticas en dos, resultando un total de 18. Los cálculos del Plan de Estudios y del Plan de Ordenación docente en el escenario actual quedan reflejados en el siguiente cuadro:

créditos en Ingeniería Informática	Grupos	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	TOTAL
Teoría				6	2,25	3	3	3	3	2,25	4,5			2,25		20,25
Práctica				3	2,25	3	3	3	1,5	2,25	4,5			2,25		24,75
Total PE				9	4,5	6	6	6	4,5	4,5	9			4,5		54
Teoría	3			18	6,75	9	9	9	9	6,75	13,5			6,75		87,75
Práctica	9			27	20,25	27	27	27	13,5	20,25	40,5			20,25		222,75
Total POD				45	27	36	36	36	22,5	27	54			27		310,5

Cuadro 16. *Reparto de créditos según Plan de Estudios y Plan de Ordenación Docente*

De acuerdo a lo anterior, el Cuadro 17 muestra el escenario simulado para la adaptación ECTS en la Ingeniería Informática.

Tal y como se muestra en el escenario simulado, el número total de horas calculado está ligeramente por debajo de las 1400, una cifra inferior a las 1500-1800 horas marcadas por el sistema ECTS. Este dato pone de manifiesto la adecuación de este escenario para su implementación en el espacio europeo y permite al docente un pequeño margen para incorporar nuevos materiales o proponer alguna actividad complementaria. Así, el número de horas por crédito es de 23,2 y el número de docentes a tiempo completo para cubrir las necesidades que marca este escenario sería de algo más de 19.

3.2 ADAPTACIÓN ECTS EN INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Siguiendo el mecanismo anterior, para la adaptación del escenario actual al nuevo contexto europeo se ha partido de la carga docente de cada asignatura de la titulación junto con la carga asignada en función del número de grupos. En ambos casos se ha tenido en cuenta la diferenciación clara entre teoría y práctica.

Para el cálculo del número de grupos teóricos y prácticos, se ha tomado como referencia el número de alumnos matriculados en el curso 2005-2006 y se ha realizado la media aritmética.

Parámetro	Valor
Coefficiente trabajo teoría	1,75
Coefficiente trabajo prácticas	2
Coefficiente grupo pequeño	0,25

DEDICACIÓN ESTUDIANTE																
ACTIVIDAD	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica			60	22,5	30	30	30	30	22,5	45			22,5			22,5
Clase práctica			15	15	15	15	15	7,5	15	30			15			15
Actividades grupo pequeño			15	7,5	15	15	15	7,5	7,5	15			7,5			7,5
Tutoría curso															1	
Estudio teoría			105	39,38	52,5	52,5	52,5	52,5	39,38	78,75			39,38			39,38
Trabajos prácticas			30	30	30	30	30	15	30	60			30			30
Trabajos grupo pequeño			3,75	1,875	3,75	3,75	3,75	1,875	1,875	3,75			1,875			1,875
Exámenes			6	2,25	3	3	3	3	2,25	4,5			2,25			2,25
Horas con profesor			96	47,25	63	63	63	48	47,25	94,5			47,25		1	47,25
Horas sin profesor			138,8	71,25	86,25	86,25	86,25	69,38	71,25	142,5			71,25			71,25
Totales			234,8	118,5	149,3	149,3	149,3	117,4	118,5	237			118,5		1	118,5

DEDICACIÓN DOCENTE																	
ACTIVIDAD	Gr.	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica	3			180	67,5	67,5	90	90	90	67,5	135			67,5			877,5
Clase práctica	9			135	135	135	135	135	67,5	135	270			135			1282,5
Actividades grupo pequeño	18			270	135	135	270	270	135	135	270			135			1890
Tutoría curso	181															181	181
Totales				585	337,5	337,5	495	495	292,5	337,5	675			337,5		181	4231

Profesores a tiempo completo	19,23
Horas crédito	23,22
Aulas grupo pequeño	2,1

Cuadro 17. Escenario de adaptación al sistema ECTS para la Ingeniería Informática

asignatura	Alumnos
AE	92
PED	212
BDI	127
AC	206
SEE	170
TAC	155
POO	194
Media	165,14

Cuadro 18. *Ingeniería Técnica en Informática de Gestión: alumnos por asignatura (2005-2006)*

Para cubrir la demanda de los 165 alumnos se han de crear dos grupos de teoría y sus correspondientes 6 grupos de prácticas y 12 grupos pequeños. Los cálculos del Plan de Estudios y del Plan de Ordenación docente en el escenario actual quedan reflejados en el siguiente cuadro:

créditos en ITIG	Grupos	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	TOTAL
Teoría		1,5	4,5	6						2,25	4,5	6			6	30,75
práctica		3	3	3						2,25	4,5	3			3	21,75
Total PE		4,5	7,5	9						4,5	9	9			9	52,5
Teoría	2	3	9	12						4,5	9	12			12	61,5
Práctica	6	18	18	18						13,5	27	18			18	130,5
Total POD		21	27	30						18	36	30			30	192

Cuadro 19. *Reparto de créditos según Plan de Estudios y Plan de Ordenación Docente*

De acuerdo a lo anterior, el Cuadro 20 muestra el escenario simulado para la adaptación ECTS en la Ingeniería Técnica en Informática de Gestión.

Tal y como ocurría en la Ingeniería Informática, el número total de horas calculado está ligeramente por debajo de las 1400, una cifra inferior a las 1500-1800 horas marcadas por el sistema ECTS. Este dato pone de manifiesto la adecuación de este escenario para su implementación en el espacio europeo y permite al docente un pequeño margen para incorporar nuevos materiales o proponer alguna actividad complementaria. Así, el número de horas por crédito es de 22,65 y el número de docentes a tiempo completo para cubrir las necesidades que marca este escenario sería de algo más de 12.

Parámetro	Valor
Coefficiente trabajo teoría	1,75
Coefficiente trabajo prácticas	2
Coefficiente grupo pequeño	0,25

DEDICACIÓN ESTUDIANTE																
ACTIVIDAD	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica	15	45	60						22,5	45	60			60		307,5
Clase práctica	15	15	15						15	30	15			15		120
Actividades grupo pequeño	15	15	15						7,5	15	15			15		97,5
Tutoría curso															1	1
Estudio teoría	26,25	78,75	105						39,38	78,75	105			105		538,125
Trabajos prácticas	30	30	30						30	60	30			30		240
Trabajos grupo pequeño	3,75	3,75	3,75						1,875	3,75	3,75			3,75		24,375
Exámenes	1,5	4,5	6						2,25	4,5	6			6		30,75
Horas con profesor	46,5	79,5	96						47,25	94,5	96			96	1	587,5
Horas sin profesor	60	112,5	138,8						71,25	142,5	138,8			138,8		802,5
Totales	106,5	192	234,8						118,5	237	234,8			234,8	1	1359,25

DEDICACIÓN DOCENTE																	
ACTIVIDAD	Gr.	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica	3			180	67,5	90	90	90	90	67,5	135			67,5			877,5
Clase práctica	9			135	135	135	135	135	67,5	135	270			135			1282,5
Actividades grupo pequeño	18			270	135	270	270	270	135	135	270			135			1890
Tutoría curso	181															181	181
Totales				585	337,5	495	495	495	292,5	337,5	675			337,5	181		4231

Profesores a tiempo completo	19,23
Horas crédito	23,22
Aulas grupo pequeño	2,1

Cuadro 20. Escenario de adaptación al sistema ECTS para la Ingeniería Técnica en Informática de Gestión

3.3 ADAPTACIÓN ECTS EN INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

De nuevo, para la adaptación del escenario actual al nuevo contexto europeo se ha partido de la carga docente de cada asignatura de la titulación junto con la carga asignada en función del número de grupos. En ambos casos se ha tenido en cuenta la diferenciación clara entre teoría y práctica.

Para el cálculo del número de grupos teóricos y prácticos, se ha tomado como referencia el número de alumnos matriculados en el curso 2005-2006 (ver Cuadro 21) y se ha realizado la media aritmética.

asignatura	alumnos
BD1	155
COMP	149
EC	141
LGA	157
PED	160
POO	136
SO	105
Media	143,29

Cuadro 21. *Ingeniería Técnica en Informática de Sistemas: alumnos por asignatura (2005-2006)*

Los 143 alumnos de media originan un único grupo de teoría y, por tanto, 3 grupos de prácticas y 6 grupos pequeños. Los cálculos del Plan de Estudios y del Plan de Ordenación docente en el escenario actual quedan reflejados en el siguiente cuadro:

créditos en ITIS	Grupos	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	TOTAL
Teoría				6	2,25	3			3	2,25	4,5		3			24
práctica				3	2,25	3			1,5	2,25	4,5		3			19,5
Total PE				9	4,5	6			4,5	4,5	9		6			43,5
Teoría	1			6	2,25	3			3	2,25	4,5		3			24
Práctica	3			9	6,75	9			4,5	6,75	13,5		9			58,5
Total POD				15	9	12			7,5	9	18		12			82,5

Cuadro 22. *Reparto de créditos según Plan de Estudios y Plan de Ordenación Docente*

De acuerdo a lo anterior, el Cuadro 23 muestra el escenario simulado para la adaptación ECTS en la Ingeniería Técnica en Informática de Sistemas.

De nuevo, se aprecia en el cuadro que el número total de horas calculado está ligeramente por encima de las 1100, muy por debajo de las 1500-1800 horas

Parámetro	Valor
Coficiente trabajo teoría	1,75
Coficiente trabajo prácticas	2
Coficiente grupo pequeño	0,25

DEDICACIÓN ESTUDIANTE																
ACTIVIDAD	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica			60	22,5	30			30	22,5	45		30				240
Clase práctica			15	15	15			7,5	15	30		15				112,5
Actividades grupo pequeño			15	7,5	15			7,5	7,5	15		15				82,5
Tutoría curso															1	1
Estudio teoría			105	39,38	52,5			52,5	39,38	78,75		52,5				420
Trabajos prácticas			30	30	30			15	30	60		30				225
Trabajos grupo pequeño			3,75	1,875	3,75			1,875	1,875	3,75		3,75				20,625
Exámenes			6	2,25	3			3	2,25	4,5		3				24
Horas con profesor			96	47,25	63			48	47,25	94,5		63			1	484
Horas sin profesor			138,8	71,25	86,25			69,38	71,25	142,5		86,25				665,625
Totales			234,8	118,5	149,3			117,4	118,5	237		149,3			1	1125,625

DEDICACIÓN DOCENTE																	
ACTIVIDAD	Gr.	AE	AC	BDI	C	EC	HP	LPP	LGA	POO	PED	SEE	SO	SOI	TAC	Otra	TOTAL
Clase teórica	1			60	22,5	30			30	22,5	45		30				240
Clase práctica	3			45	45	45			22,5	45	90		45				337,5
Actividades grupo pequeño	6			90	45	90			45	45	90		90				495
Tutoría curso	143															143	143
Totales				195	112,5	165			97,5	112,5	225		165			143	1215,5

Profesores a tiempo completo	5,53
Horas crédito	18,76
Aulas grupo pequeño	0,55

Cuadro 23. Escenario de adaptación al sistema ECTS para la Ingeniería Técnica en Informática de Sistemas

marcadas por el sistema ECTS. Este dato pone de manifiesto la adecuación de este escenario para su implementación en el espacio europeo y permite al docente un pequeño margen para incorporar nuevos materiales o proponer alguna actividad complementaria. Así, el número de horas por crédito es de 18,76 y el número de docentes a tiempo completo para cubrir las necesidades que marca este escenario sería de unos 6.

Estableciendo una comparativa directa entre los escenarios de POD de las dos titulaciones técnicas, se puede comprobar cómo el sistema de módulos para la creación de grupos de teoría da resultados de adaptación muy distintos ante diferencias no excesivamente grandes (unos 20 alumnos aproximadamente). Evidentemente, esta diferencia debería ser compensada potenciando las actividades complementarias ya mencionadas.

4. OBJETIVOS DEL SEGUNDO CURSO

Si bien en el presente trabajo se han tratado las asignaturas troncales y obligatorias de segundo curso para las actuales titulaciones de informática, dado que se trata de un curso de primer ciclo y con el fin de converger hacia las propuestas del ya mencionado libro blanco, se ha intentado establecer una serie de parámetros comunes al conjunto de las asignaturas tratadas.

Así, los objetivos de este curso se corresponden con los objetivos instrumentales, interpersonales, y sistémicos comunes a las asignaturas de la tres titulaciones, integrables, por su carácter general, a los objetivos generales de un único título en Informática. Estos objetivos son:

4.1 OBJETIVOS INSTRUMENTALES GENERALES (SABER Y SABER HACER)

- cOI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y prácticas de ordenador.
- cOI2: Utilizar con fluidez el software necesario en las prácticas relacionadas con cada asignatura.
- cOI3: Adquirir y utilizar un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso.
- cOI4: Conocer y utilizar la terminología usual de cada asignatura.
- cOI5: Adquirir un buen manejo de la bibliografía existente en la asignatura, de forma que se potencia su autosuficiencia a la hora de completar su formación.
- cOI6: Comprender el ámbito de acción de la asignatura dentro de la titulación de Informática y dentro de los perfiles profesionales.

4.2 OBJETIVOS INTERPERSONALES GENERALES (SER Y ESTAR)

- cOIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo, etc.

- cOIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional.
- cOIP3: Comprometerse de forma ética con el trabajo, con el resto de integrantes del grupo y consigo mismo.

4.3 OBJETIVOS SISTÉMICOS GENERALES

- cOS1: Capacidad de integrar los conocimientos, métodos, algoritmos y destrezas prácticas de cada asignatura para resolver situaciones reales relacionadas con la informática y otras disciplinas relacionadas.
- cOS2: Reforzar el hábito de plantearse interrogantes; ante un problema deben preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría en las condiciones iniciales alguna modificación, etc.
- cOS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de cada asignatura de forma interdisciplinar.
- cOS4: Adquirir una comprensión del método científico, a través de diversas actividades realizadas en la asignatura, y asimilar su importancia como manera de pensar y actuar en su labor de científico e ingeniero, fomentando su capacidad de abstracción y su espíritu crítico.

5. COMPETENCIAS ACADÉMICAS Y PROFESIONALES

Los objetivos antes descritos requieren de la adquisición de un conjunto de competencias comunes a todos los profesionales dentro del marco del título en Informática. Al igual que el caso de los objetivos, estas competencias se pueden clasificar en instrumentales, interpersonales, y sistémicas, si bien dentro de cada grupo podemos realizar una clasificación adicional:

5.1 COMPETENCIAS INSTRUMENTALES (SABER Y SABER HACER)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

5.1.1 Habilidades cognitivas (saber):

- cCIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación de informática a partir de la introducción explícita en clase de ejemplos y comentarios, aprovechando puntos relevantes del temario.
- cCIC2: Conocer y entender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales, ayudándose de la introducción explícita en clase de ejemplos y comentarios en puntos relevantes del temario.

5.1.2 Capacidades metodológicas (saber hacer):

- cCIM1: Ser capaz de tomar decisiones de manera razonada.
- cCIM2: Ser capaz de analizar y sintetizar.
- cCIM3: Ser capaz de manejar la bibliografía relacionada con cada asignatura.

5.1.3 Destrezas tecnológicas (saber hacer):

- cCIT1: Habilidades básicas de consulta de la red informática para la obtención y manejo de información relacionada con la asignatura.

5.1.4 Destrezas lingüísticas (saber hacer):

- cCIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de fenómenos o experimentos.
- cCIL2: Conocer y utilizar la terminología usual de la asignatura, tanto en castellano y/o en valenciano, y conocer dicha terminología en inglés.

5.2 COMPETENCIAS INTERPERSONALES (SER Y ESTAR)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

5.2.1 Competencias para tareas colaborativas:

- cCIPTC1: Ser capaz de realizar opcionalmente un trabajo en equipo de ampliación de la materia estudiada en cada asignatura.
- cCIPTC2: Ser capaz de trabajar en equipo para resolver cuestiones y problemas relacionados con la materia estudiada en cada asignatura.

5.2.2 Compromiso con el trabajo:

- cCIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar.
- cCIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado.
- cCIPTR3: Se debe cumplir el plazo de entrega de dichos trabajos.
- cCIPTR4: Es importante adquirir un compromiso ético entre todos los componentes del grupo.

5.3 COMPETENCIAS SISTÉMICAS

En este caso atendemos a la integración de capacidades cognitivas, destrezas prácticas y disposiciones:

- cCS1: Capacidad de aplicar los conocimientos, métodos y algoritmos vistos en cada asignatura a situaciones y problemas concretos del área de informática y de otras disciplinas relacionadas.

- cCS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura.
- cCS3: Motivación por la calidad y por la creatividad.
- cCS4: Capacidad de adoptar el protocolo dado por el método científico en el planteamiento y realización de trabajos diversos tanto a nivel académico como profesional.
- cCS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de desarrollo profesional (adaptación a nuevas situaciones).

6. METODOLOGÍA DEL APRENDIZAJE

El aprendizaje de una materia debe potenciar, por un lado, los conocimientos sobre la misma y, por otro, sus aspectos más pragmáticos. Ambos niveles, considerados tradicionalmente en la clásica dicotomía teoría/práctica, deben ser reforzados con complementos que mejoren su rendimiento. No obstante, y previamente a la impartición de la materia, sería deseable asegurarse de que los conocimientos previos supuestos en el alumno se cumplen razonablemente a través del correspondiente análisis de nivel. Para ello, el docente debería:

- Establecer mecanismos de control del nivel mínimo.
- Completar los materiales de la asignatura con referencias y contenidos que refuercen los conocimientos previos no contemplados propiamente en la materia pero necesarios para su superación.
- Transmitir al alumno la necesidad de su implicación en el proceso de aprendizaje para cubrir cualquier carencia existente reforzando además su confianza.
- Hacer efectiva esta implicación a través de ejercicios que permitan desarrollar y potenciar los conceptos de los que carece.
- Fomentar en cualquier la técnica de trabajo que facilite la integración del alumno en una dinámica activa, participativa, de trabajo en grupo y de aprendizaje continuado.

A continuación se detallarán los aspectos principales vinculados a las diferentes técnicas metodológicas implicadas en el proceso de aprendizaje, desde la clase magistral, como espacio común para grandes grupos, hasta el proceso tutorial individualizado.

6.1 LECCIÓN MAGISTRAL

Dado que la lección magistral es, sin duda, la técnica formativa más utilizada en la universidad española, especialmente debido a la existencia de grandes grupos de alumnos, ha demostrado, bajo estas circunstancias, ser un sistema eficaz de enseñanza, aunque quizás no de aprendizaje. Sin duda se trata de un sistema directo, rápido y que ofrece al alumno un contexto cómodo, pero que al mismo tiempo fomenta su pasividad y carece de mecanismos de control sobre lo aprendido.

Bajo la perspectiva de los nuevos modelos educativos, dirigidos más hacia el pensamiento creativo que hacia la simple impartición de conceptos, la clase magistral, base de las competencias conceptuales y procedimentales, debe ser un elemento fundamental de la enseñanza que debe, bajo cualquier circunstancia, complementarse con otras técnicas que la refuercen, como las prácticas de laboratorio o las actividades en grupos pequeños, que mejorarán las competencias instrumentales y metodológicas.

Las fases básicas de una clase magistral podrían ser las siguientes:

- Introducción sobre los objetivos y los contenidos de la clase.
- Referencias bibliográficas necesarias para el seguimiento de dichos contenidos.
- Referencias a los contenidos previos ya vistos y relacionados con lo que se va a exponer con el fin de dotarle de continuidad y de contexto.
- Desarrollo de contenidos a través de una exposición clara que ponga de manifiesto lo relevante frente a lo periférico, con aspectos formales y ejemplos que ilustren los conceptos más complejos.
- Resumen y conclusiones respecto al cumplimiento de los objetivos planteados inicialmente.

Con el fin de evitar los principales inconvenientes de clase magistral, entre los que se encuentra el ya mencionado fomento de la pasividad del alumno, es posible favorecer un ambiente dinámico y participativo con el uso de técnicas complementarias:

- Uso de material de apoyo tales como apuntes o transparencias: esta técnica ha de usarse con una cierta moderación ya que, pretendiendo favorecer la participación del alumno, puede ocasionar el efecto completamente contrario. Es muy común que un alumno considere las transparencias como el material de trabajo único, en vez de cómo un mecanismo de apoyo al aprendizaje. El docente ha de tener en cuenta estos aspectos para aprovechar el uso de este material (habitualmente en consonancia con un proyector y con la propia pizarra) sin perder la capacidad de mantener al alumno en el hilo conductor de la explicación.
- Fomentar un clima de confianza que posibilite la fluidez en la comunicación entre el profesor y el alumno. Dar confianza para hacer preguntas, que tienden a parecer al alumno irrelevantes, ante la errónea concepción de la duda como una limitación individual. Conseguir este clima pasa en ocasiones por la organización de tutorías para grupos reducidos
- Responder a las dudas conocidas a partir de la experiencia docente también ayuda a aprender de los errores ajenos.
- Fomentar el diálogo entre compañeros, colocando al docente en una posición neutral y de moderación, ayuda a reforzar posturas y hace más interesante el proceso de aprendizaje.

- Fomentar el resumen de conceptos vistos y la introducción de conceptos venideros, con el fin de incentivar al alumno, mostrándole las ventajas competitivas del estudio anticipado.

6.2 RESOLUCIÓN DE PROBLEMAS Y PRÁCTICAS DE LABORATORIO

Es evidente la necesidad de complementar el estudio teórico con la resolución de problemas específicos, algo que constituirá sin duda uno de los elementos esenciales en el desarrollo laboral del alumno. Para ello, la organización de clases de problemas es fundamental y permitirá al alumno reforzar los conocimientos adquiridos en la clase magistral y, sobre todo, encontrar su aplicación (cuestionada en ocasiones) en casos concretos.

En lo referente a materias altamente tecnológicas, como ocurre en muchas de las asignaturas tratadas, las prácticas de laboratorio juegan un papel fundamental. Si bien sería deseable que las prácticas se realizaran de manera individual, no siempre la distribución de grupos lo permite. Esto, aun pudiéndose considerar un aspecto negativo, puede manejarse como algo positivo en el aprendizaje ya que se combina con las habilidades relacionadas con el trabajo en equipo. Las clases de laboratorio no deben ser simples horas de uso libre del material, sino que deben estar estructuradas como ocurría en las lecciones teóricas. Así, el docente debería:

- Introducir los conceptos que se tratarán en la práctica.
- Especificar los resultados que se desean obtener y la manera de presentarlos, fomentando el uso de técnicas de entrega mixtas con elaboración de informes y presentaciones telemáticas.
- Establecer los paralelismos necesarios con los aspectos teóricos de la materia para evitar juicios erróneos sobre la conexión entre teoría y práctica. Para ello, la planificación de la asignatura debe ser especialmente meticulosa en su estructuración, respetando y previendo los tiempos de desarrollo teórico-práctico y la consecución de objetivos convenientemente sincronizada.

6.3 PROCESO TUTORIAL

Éste es sin duda uno de los mecanismos más interesantes y menos utilizados por los alumnos en el proceso de aprendizaje. El proceso tutorial no debería ser concebido como un sustitutivo de las anteriores metodologías, sino como un complemento de refuerzo para afianzar los conocimientos que, por unas o otras razones, no han conseguido desarrollar las competencias del alumno y cumplir los objetivos convenientemente.

Al clásico proceso tutorial presencial de carácter individual o en pequeños grupos de dos o tres personas, que se realiza habitualmente en horario de atención al alumno y en el propio despacho del docente, hay que unir los mecanis-

mos telemáticos que, gracias a las nuevas tecnologías, han facilitado la comunicación profesor-alumno al prescindir de cualquier carácter presencial. Sin duda, estos procesos virtuales, como es el caso de las tutorías que se realizan en nuestra universidad a través de la herramienta corporativa Campus Virtual³, han demostrado un enorme éxito en el alumnado, que no sólo dispone de una potente plataforma de comunicación, sino que le ayuda a estrechar la brecha inicial existente con el profesor y facilita la creación de ese clima de confianza que mencionábamos anteriormente.

Además de estos procesos tutoriales, se propone la posibilidad de desarrollar sesiones planificadas, con carácter algo menos individual y más cercano al ya denominado trabajo en pequeños grupos. Esta metodología añade nuevas dimensiones al proceso tutorial y genera grupos colaborativos y foros de discusión que fomentan el aprendizaje.

6.4 RECURSOS, MATERIALES Y HERRAMIENTAS DE APRENDIZAJE

Ya se ha comentado las ventajas que la plataforma Campus Virtual ofrece en los procesos tutoriales. Sin duda, esta herramienta proporciona la base de la metodología docente de nuestra universidad en lo que a recursos y herramientas se refiere. Las fichas de las asignaturas, los materiales docentes, los tests de evaluación, la introducción de anuncios, bibliografía, cuestiones frecuentes o debates son algunos ejemplos que ponen de manifiesto las enormes posibilidades de aplicación de metodologías de aprendizaje.

6.5 CRITERIOS DE CALIFICACIÓN

Si bien es cierto que tradicionalmente la calificación ha sido el resultado numérico de un conjunto más o menos numeroso de pruebas de distinto tipo, sería deseable que en función de la calificación obtenida por el alumno, se cumplieran los criterios marcados en el siguiente cuadro:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento y comprensión de la materia se extiende más allá del trabajo cubierto por el programa. • Los problemas relacionados con la asignatura son resueltos con eficiencia y precisión. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de resultados. • La participación en las clases teóricas y prácticas y en las tutorías ha sido correcta y muy satisfactoria

3 Para más información sobre esta herramienta, visitar <http://www.ua.es/es/univirtual/index.html>.

Calificación	Rango	Pautas
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento y comprensión de la materia es satisfactorio. • Los problemas relacionados con la asignatura son resueltos con precisión. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de resultados aceptables. • La participación en las clases teóricas y prácticas y en las tutorías ha sido correcta y bastante satisfactoria.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y comprensión del contenido del curso es básico. • Los problemas relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio son usualmente desarrolladas con éxito razonable. • La participación en las clases teóricas y prácticas y en las tutorías ha sido correcta pero no siempre satisfactoria.
Suspense	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y comprensión contenido del curso no ha sido aceptable. • Los problemas relacionados con la asignatura no son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio son usualmente desarrolladas de forma no satisfactoria, y el significado y análisis de los resultados no son entendidos generalmente. • La participación en las clases teóricas y prácticas y en las tutorías ha sido escasa y deficiente.

Cuadro 24. *Criterios de calificación*

6.6 ESTRATEGIAS DE APRENDIZAJE DEL ALUMNO

En la medida en que una de las novedades que plantea el sistema ECTS es la medición del tiempo de aprendizaje frente al tradicional tiempo de enseñanza, el alumno ha de estar especialmente integrado en el proceso de aprendizaje para que los objetivos se cumplan de manera más solvente. Así, el alumno debería:

- Responsabilizarse en la gestión y control de su propio aprendizaje.
- Desarrollar un espíritu crítico y una actitud abierta ante los cambios científico-técnicos de su especialidad, con curiosidad intelectual y rigor científico.
- Estimular el perfeccionamiento profesional y la formación continuada.
- Fomentar actitudes y adquirir técnicas para un eficaz trabajo en equipo.
- Basar en criterios deontológicos su futuro comportamiento profesional.
- Valorar el sentido humanístico de la ciencia y la técnica como resultado del esfuerzo de generaciones anteriores, que debe servir para resolver los problemas del hombre actual y futuro.

La forma de conseguir estos objetivos pasa por la forma de enfrentarse a cada uno de los procesos docentes y de aprendizaje comentados anteriormente.

6.6.1 Lección Magistral

La asimilación de contenidos expuestos en las lecciones magistrales debería ser asegurada a través del repaso tanto individualmente como con el uso de técnicas de estudio en grupo. La frecuencia de este estudio tiene que permitir al alumno mantener la dinámica en clase y evitar así las típicas situaciones de abandono prematuro. Esta frecuencia garantizará, además, la necesidad o no de los procesos tutoriales.

Es vital que la actitud del alumno en el desarrollo de la lección magistral sea lo más dinámica posible. Esto facilitará (y disminuirá notablemente) el estudio posterior.

6.6.2 Clases de problemas y prácticas de laboratorio

Durante las clases de problemas o las prácticas de laboratorio, el alumno deberá resolver, a partir de ejemplos clarificadores, los ejercicios que se le planteen, sin esperar a que sean resueltos en clase o a que las soluciones sean proporcionadas. La correcta realización de estos ejercicios garantiza el proceso de autoevaluación y, por tanto, el mantenimiento de la evaluación continua, además de proporcionar una base metodológica de solución de ejercicios.

En el caso de que existan ejercicios sin una solución concreta o de los que no se proporcione solución, el alumno tendrá que desarrollar las estrategias necesarias que garanticen la solución a partir de lo conocido, preparándoles, a través de las correspondientes competencias, para abordar determinados casos reales que pueden darse en su vida laboral.

El alumno deberá también resolver problemas propuestos y no realizados en el periodo de clase.

En las entregas de soluciones de problemas o de prácticas de laboratorio, el alumno deberá conocer los objetivos propuestos y tendrá que elaborar un informe o una memoria de prácticas. Si además, este trabajo se realiza por parte de varios compañeros, el alumno se familiarizará con dinámicas de grupo propias del trabajo en equipo, teniendo que asumir la responsabilidad de la entrega total y no de la fracción realizada individualmente.

6.6.3 Recursos de aprendizaje

El alumno debe conocer todas las herramientas a su alcance para el correcto desarrollo del proceso de aprendizaje. Tendrá que estar al tanto de las novedades de la asignatura a través de los anuncios así como de los materiales distribuidos por parte del docente.

6.6.4 Proceso tutorial

El alumno podrá consultar dudas bien de manera presencial en el despacho del profesor o bien virtualmente a través de mecanismos telemáticos como el Campus Virtual.

Es importante que el estudiante adquiera conciencia del valor del tiempo tutorial, por lo que es fundamental que las dudas se planteen de forma concreta y, de tratarse de algún tipo de “recuperación docente”, se acuda al profesor con los conceptos estudiados a falta de concretar aspectos específicos.

El alumno tendrá también la posibilidad de asistir a las tutorías en pequeños grupos fomentando en la medida de lo posible el debate y el dinamismo.

6.6.5 Evaluación

Si el alumno ha seguido los pasos anteriores, la preparación para el examen consistirá en un exhaustivo repaso de los contenidos ya estudiados, haciendo especial hincapié en lo expuesto a lo largo del proceso docente y garantizando el conocimiento de la materia, si es posible, más allá de lo estrictamente contenido en ella.

7. BIBLIOGRAFÍA

- [ACM01] Association for Computing Machinery. *ACM code of ethics and professional conduct*. New York: The Association for Computing Machinery, May 2001. <http://www.acm.org/constitution/code.html>
- [ACM01b] Computing Curricula 2001. *Computer Science, The Joint Task Force on Computing Curricula. IEEE Computer Society y Association for Computing Machinery*, Final Report, December 15, 2001.
- [ACM04] ACM/IEEE Joint Task Force on Computer Engineering Curricula. *Computing Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*. 2004.
- [ACM68] ACM Curriculum Committee on Computer Science. Curriculum 68: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 11(3):151-197, March 1968.
- [ACM78] ACM Curriculum Committee on Computer Science. Curriculum '78: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3):147-166, March 1979.
- [ACM79] ACM (March 1979). “Curriculum '78: Recommendations for the Undergraduate program in Computer Science”, *Communications of the ACM*, vol. 22, no. 3, pp. 147-166, March 1979.
- [ACM91] ACM/IEEE-CS Joint Curriculum Task Force Report. Computing Curricula 1991, *Communications of the ACM*, 34(6):69-84, June 1991.
- [Austing79] R.H. Austing, B.H. Barnes, D.T. Bonnette, G.L. Engel y G. Stokes. Curriculum 78. Recommendations for the undergraduate program in computer science. *ACM Curriculum Committee on Computer Science*, 1979.

- [Bennett86] W. Bennett. A position paper on guidelines for electrical and computer engineering education. *IEEE Transactions in Education*, E-29(3):175-177, August 1986.
- [Campos02] Javier Campos, Josep Casanovas, José Manuel Colom, Gregori Martín, Javier Martínez, Ana Pont, Ramón Puigjaner, Antonio Robles y María Ribera Sancho, *Informe sobre la Adaptación de los Estudios de las Ingenierías en Informática a la Declaración de Bolonia*. Barcelona, Junio 2002.
- [Clocksin87] Clocksin, W. F. y Mellish, C. S. (1987). *Programación en Prolog*. Gustavo Pili.
- [COPIITI03] COPIITI, *Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática. Perfil de la profesión de Ingeniero en Informática y definición del currículo académico*, 2003.
- [CS04] Consorcio Career-Space (CS) - Centro Europeo para el Desarrollo de la Formación Profesional (CEDEFOP). *Directrices para el desarrollo curricular. Nuevos currículos de TIC para el siglo XXI: el diseño de la educación del mañana*. (Oficina de Publicaciones Oficiales de las Comunidades Europeas, Lu.
- [CSTB94] *Computing Science and Telecommunications Board. Realizing the information future*. Washington DC: National Academy Press, 1994.
- [CSTB99] *Computing Science and Telecommunications Board. Being fluent with information technology*. Washington DC: National Academy Press, 1999.
- [Deming82] W.E. Deming, *Quality, Productivity and Competitive Position*. Massachusetts Institute of Technology, 1982
- [Denning89] Peter J. Denning, Douglas E. Comer, David Gries, Michael C. Mulder, Allen B. Tucker, A. Joe Turner, and Paul R. Young. *Computing as a discipline. Communications of the ACM*, 32(1):9-23, January 1989.
- [Dodd90] Dodd, T. (1990). *Prolog. A Logical Approach*. Oxford University Press.
- [EAB86] Educational Activities Board. Design education in computer science and engineering. *Technical Report 971, Computer Society of the IEEE*, October 1986.
- [EC77] Education Committee of the IEEE Computer Society. A curriculum in computer science and engineering. *Publication EHO119-8, Computer Society of the IEEE*, January 1977.
- [EPS1] F. Llopis Pascual y F. Llorens Largo (Eds). (2005) *Adecuación del primer curso de los estudios de informática al Espacio Europeo de Educación Superior*. Vicerrectorado de Calidad y Armonización Europea. Instituto de Ciencias de la Educación. Serie Docencia Universitaria – EEES. Universidad de Alicante.
- [Gibbs86] Norman E. Gibbs and Allen B. Tucker. Model curriculum for a liberal arts degree in computer science. *Communications of the ACM*, 29(3):202-210, March 1986.

- [IEEE83] Educational Activities Board IEEE. *The 1983 Model Program in Computer Science and Engineering*. IEEE Computing Society Press.1983
- [IF-U00] *Informatics Curriculum Framework 2000 for Higher Education*, 2000.
- [IF-U84] *A Modular Curriculum in Computer Science*, 1984.
- [Martin96] C. Dianne Martin, Chuck Huff, Donald Gotterbarn, Keith Miller. *Implementing a tenth strand in the CS curriculum. Communications of the ACM*, 39(12):75-84, December 1996.
- [Myers98] J. Paul Myers, Jr. and Henry M. Walker. The state of academic hiring in computer science: *An interim review. SIGCSE Bulletin*, 30(4):32a-35a, December 1998.
- [Phadke83] M. Phadke, R. Kackar, D. Speeney y M. Grieco. Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental design. *The Bell System Technical Journal*, 1983.
- [Ralston80] Anthony Ralston and Mary Shaw. Curriculum 78. Is computer science really that unmathematical. *Communications of the ACM* (23)2:67-70, February 1980.
- [Shaw85] Mary Shaw. *The Carnegie-Mellon curriculum for undergraduate computer science*. New York: Springer-Verlag, 1985.
- [Sterling94] Sterling, L. y Shapiro, E. (1994). *The Art of Prolog. Advanced Programming Techniques*. 2ª, The MIT Press.
- [Tucker91] Allen B. Tucker, Bruce H. Barnes, Robert M. Aiken, Keith Barker, Kim B. Bruce, J. Thomas Cain, Susan E. Conry, Gerald L. Engel, Richard G. Epstein, Doris K. Lidtke, Michael C. Mulder, Jean B. Rogers, Eugene H. Spafford, and A. Joe Turner. *Computing Curricula '91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991*.

8. ADAPTACIÓN DE ASIGNATURAS: GUÍAS DOCENTES

A continuación se presenta el trabajo realizado por todas y cada una de las asignaturas que han participado en el presente trabajo y contenido en la guía docente de la asignatura. Cada guía detalla, entre otros, los objetivos, las competencias, los prerrequisitos, la metodología docente y las estrategias de aprendizaje para cada asignatura concreta. No obstante, al tratarse de asignaturas enmarcadas en un mismo contexto y curso, es lógico encontrar puntos en común no solo a todas las asignaturas sino al perfil de un Ingeniero en Informática. La relación entre objetivos y competencias de cada una de las asignaturas se muestra a través de la tabla de coherencia al final de cada una de las guías.

La última de las guías docentes corresponde a la asignatura Relaciones Jurídicas Básicas que, si bien no se enmarca en el segundo curso, como ya se ha comentado, ha participado en la elaboración de guías en el periodo actual completando así el trabajo realizado por el resto de asignaturas troncales y obligatorias de primer curso.

GUÍA DOCENTE DE AMPLIACIÓN DE ESTADÍSTICA

Violeta Migallón Gomis; José Penadés Martínez; Rafael Álvarez Sánchez

*Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
(violeta; jpenades; ralvarez)@dccia.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

A la hora de elaborar la propuesta docente que nos ocupa, se ha de tener en cuenta cuál ha sido la evolución de las distintas recomendaciones curriculares de informática publicadas por instituciones de prestigio internacional, desde las propuestas iniciales de la ACM (Association for Computing Machinery) [ACM68] y el IEEE (Institute for Electrical and Electronic Engineers) [EC77], de 1968 y 1977, respectivamente -donde por primera vez se intenta dar un carácter autónomo a la informática- hasta las tendencias actuales recogidas en el Computing Curricula 2001 [ACM01b], realizado conjuntamente por el IEEE y la ACM. Analizando esta evolución se observa que los avances técnicos han hecho que muchas de las materias de la Ingeniería Informática de la década pasada hayan ganado importancia dando lugar a cuatro grandes perfiles genéricos, tal y como se refleja en el Computing Curricula 2001: Computer Science, Computer Engineering, Software Engineering e Information Systems. Las actuales titulaciones de Informática están relacionadas principalmente con los tres primeros perfiles. En dichos perfiles, la Estadística es considerada materia fundamental, aunque con contenidos algo distintos. Así, en el perfil de Computer Science aparecen dentro del área de estructuras discretas las materias de Fundamentos Básicos de Conteo y Probabilidad Discreta, además de incluir contenidos de razonamiento probabilístico y teoría de la decisión en el área de Sistemas Inteli-

gentes. Respecto al perfil Software Engineering, se hace hincapié en la importancia que tiene en este perfil la estadística y métodos empíricos proponiendo complementar los contenidos anteriores con temas más avanzados de estadística en el contexto de la informática, diseño de experimentos y análisis de resultados. Sin embargo, es el perfil de Computer Engineering donde se definen más claramente los contenidos básicos relacionados con la Estadística, necesarios para dicho perfil. Concretamente, estos contenidos son: probabilidad discreta, probabilidad continua, esperanza, distribuciones muestrales, estimación, contrastes de hipótesis, correlación y regresión.

Por otra parte, si nos situamos en el contexto español, debemos observar primero cuáles son las directrices generales propias de las titulaciones de Informática (Reales decretos 1459/1990, 1460/1990, 1461/1990 de 26 de octubre, BOE 1990). En las titulaciones de Ingeniería en Informática e Ingeniería Técnica en Informática de Sistemas aparece como materia troncal de obligatoria inclusión, en todos los planes de estudio, la materia “Estadística”, cuyos contenidos son: estadística descriptiva, probabilidades, y métodos estadísticos aplicados, y asociados a ellos, 6 créditos troncales; mientras que la Ingeniería Técnica en Informática de Gestión tiene asociados 9 créditos troncales.

Siguiendo necesariamente estas directrices propias, en la Universidad de Alicante se han distribuido los contenidos de Estadística en dos asignaturas: Estadística y Ampliación Estadística. La primera es troncal en las tres titulaciones, mientras que la segunda es troncal en la Ingeniería Técnica en Informática de Gestión y optativa en las otras dos titulaciones.

En los planes de estudio de Informática de la Universidad de Alicante la asignatura Ampliación de Estadística pretende complementar la asignatura Estadística cubriendo parte de las necesidades, que sobre estadística tendrá el futuro informático, especialmente en métodos estadísticos aplicados, ya que estos son sus descriptores.

La Estadística ha evolucionado, a lo largo de los últimos 150 años, desde su antiguo propósito de elaborar la contabilidad de los estados, a ser la metodología científica de las ciencias empíricas y que permite resolver gran cantidad de problemas en las organizaciones modernas. Los campos de aplicación de la Estadística son tan numerosos como los que precisen del análisis de datos. Por ello, su expansión es continua. Centrándonos en el análisis de la adecuación del perfil de la asignatura a los perfiles de la titulación, diremos que la Estadística juega un papel fundamental a la hora de testar y evaluar hardware, software o sistemas de comunicación donde la calidad, fiabilidad y productividad deben estar garantizadas. Además los descriptores de esta asignatura son básicos para la formación de un informático ya que proporcionan técnicas que ayudan a entender o resolver problemas que se presentan en distintos ámbitos de la informática:

- técnicas de minería de datos para el análisis de bases de datos,
- técnicas de compresión de datos,
- técnicas para el reconocimiento de la voz,
- técnicas para el reconocimiento de imágenes y visión computarizada,
- evaluación de sistemas informáticos para optimizar su gestión,
- análisis de las conexiones de Internet y del flujo de información en la red,
- análisis textual del contenido de las páginas de Internet, de cara a su clasificación según una ontología.

De todo lo expuesto se deduce que, aunque los tópicos relacionados con esta asignatura van a ser básicos en la formación de cualquier ingeniero informático, sea cual sea su perfil académico y su futuro perfil profesional, toman especial relevancia en los siguientes perfiles, englobados a su vez en tres perfiles profesionales generales.

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: para el diseño de software es esencial saber analizar las distintas rutinas y módulos en términos de eficiencia y correcto funcionamiento. El control estadístico de calidad y la teoría de la fiabilidad proporcionan técnicas adecuadas para dicho análisis.
- Arquitectura y diseño de software: en el diseño de software para permitir y controlar el uso de redes informáticas y en los sistemas de gestión de bases de datos es muy útil el control estadístico de calidad y la teoría de la fiabilidad.
- Diseño multimedia: entre las capacidades profesionales técnicas se encuentra la ingeniería del software y la programación informática, las cuales requieren de conocimientos relacionados con el control de calidad y la teoría de la fiabilidad, con el fin de diseñar un producto multimedia competitivo.

Perfil profesional de sistemas:

- Ingeniería de comunicación de datos: en dicho perfil profesional es esencial tener una base fuerte sobre los distintos modelos estadísticos de estimación.
- Diseño de redes de comunicación: es esencial tener una base fuerte en modelos estadísticos.
- Asistencia técnica: en este perfil se debe tener conocimientos básicos sobre control de calidad.
- Ingeniería de integración y pruebas e implantación y pruebas: en la realización de pruebas, para asegurar que el sistema funcione según las especificaciones se usan, entre otras, técnicas relacionadas con la inferencia estadística y la estadística descriptiva.

Perfil profesional de gestión y explotación de tecnologías de la información:

- Consultoría de empresas de TI: la estrategia empresarial requiere de técnicas de planificación, basadas en muchas ocasiones en estudios estadísticos.
- Especialista en sistemas: en dicho perfil es necesaria una base fuerte en matemáticas y análisis estadístico. En particular, la estadística descriptiva multivariante y el control estadístico van a ser útiles a la hora de realizar y analizar los estudios comparativos para demostrar las capacidades de dichos sistemas.
- Desarrollo de investigación y tecnología: la participación y gestión de proyectos, ya sean de pequeña envergadura o proyectos de investigación internacionales, requieren de estrategia, planificación y control. Algunas de las técnicas las proporciona la estadística.
- Dirección de TIC (marketing, proyectos, dirección general): este perfil requiere un conocimiento amplio del resto de áreas tecnológicas de las Tecnologías de la Información y las Comunicaciones (TIC) y, por tanto, también de aquellas que requieren de la estadística.

Esta adecuación de los tópicos relacionados con la asignatura Ampliación de Estadística a los distintos perfiles profesionales queda resumida en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad del software.
Arquitectura y diseño de software	Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad del software.
Diseño multimedia	Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad del software.
Ingeniería de comunicación de datos	Conocimiento y habilidad para el estudio y aplicación de los distintos modelos estadísticos de estimación.
Diseño de redes de comunicación	Conocimiento y habilidad para el estudio y aplicación de los distintos modelos estadísticos.
Asistencia técnica	Conocimiento y habilidad en la aplicación de la estadística al control de sistemas informáticos
Ingeniería de integración y pruebas e implantación y pruebas	Conocimiento y habilidad para el estudio y aplicación de la estadística descriptiva y la inferencia estadística.
Consultoría en empresas de TI	Conocimiento y habilidad en técnicas de planificación.
Especialista en sistemas	Conocimiento y habilidad en la aplicación de la estadística al control de sistemas informáticos. Conocimiento y habilidad en técnicas de planificación y estadística multivariante.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de investigación y tecnología	Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad. Conocimiento y habilidad en técnicas de planificación.
Dirección de TIC	Conocimiento y habilidad en la aplicación de la estadística al control de sistemas informáticos. Conocimiento y habilidad en técnicas de planificación. Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad.

Cuadro 1. *Adecuación de los tópicos relacionados con la asignatura Ampliación de Estadística*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Ampliación de Estadística forma parte del segundo curso de Ingeniería Técnica en Informática de Gestión, como asignatura troncal, y de la Ingeniería Técnica en Informática de Sistemas y la Ingeniería Informática, como asignatura optativa. Dicha asignatura tiene una relación clara y estrecha con varias materias de primer curso donde se imparten conceptos relacionados con la Estadística y cuyo entendimiento, por lo menos a nivel básico, es necesario para abordar con éxito la materia que nos ocupa. Concretamente, dichas asignaturas son:

- **Álgebra:** forma parte del primer cuatrimestre del primer curso de las tres titulaciones de Informática como asignatura troncal. Dentro de sus descriptores aparecen la teoría de conjuntos, básica para el entendimiento de la asignatura.
- **Estadística:** forma parte del segundo cuatrimestre del primer curso de las tres titulaciones de Informática como asignatura troncal. Los descriptores de dicha asignatura son: estadística descriptiva, probabilidades, y métodos estadísticos aplicados. Esta asignatura es imprescindible para entender la asignatura que nos ocupa. De hecho, Ampliación de Estadística es incompatible con Estadística.
- **Matemática Discreta:** forma parte del primer curso de las titulaciones de informática como asignatura troncal que se imparte en el segundo cuatrimestre. Los descriptores son: aritmética entera y modular, combinatoria y grafos. La combinatoria es básica para la asignatura Ampliación de Estadística.

Además de estas relaciones, en cursos más avanzados, existen otras asignaturas que entre sus tópicos incluyen temas más avanzados en los que se necesita la estadística:

- **Razonamiento:** se ofrece como asignatura optativa para la Ingeniería en Informática. Los descriptores de dicha asignatura son: métodos de razona-

miento artificial, razonamiento condicional, razonamiento con incertidumbre y razonamiento temporal.

- Fundamentos de Inteligencia Artificial: asignatura troncal de cuarto curso que se imparte en el primer cuatrimestre. Sus descriptors son: heurística y sistemas basados en el conocimiento.
- Aplicaciones Industriales del Reconocimiento Automático. Asignatura optativa cuyos descriptors son: técnicas de reconocimiento de formas y sus aplicaciones.
- Lenguajes, Gramáticas y Autómatas: es una asignatura troncal de segundo curso de la titulación de Ingeniería Informática. Los descriptors de dicha asignatura son: máquinas secuenciales y autómatas finitos, gramáticas y lenguajes formales, y redes neuronales.
- Técnicas de Inteligencia Artificial. Asignatura troncal de cuarto curso cuyos descriptors son: aprendizaje y percepción.
- Algoritmia Avanzada. Asignatura obligatoria de cuarto curso para Ingeniería Informática. Los descriptors de dicha asignatura son: búsqueda exhaustiva y estocástica, programación dinámica y algoritmos de codificación y compresión.

La siguiente figura resume las relaciones anteriormente descritas:

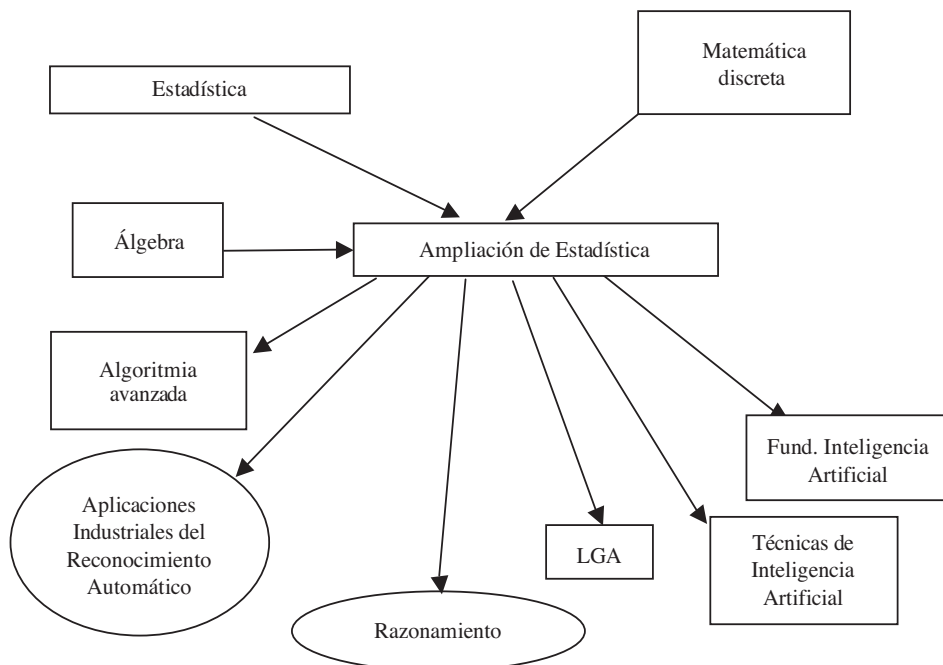


Ilustración 1. Relaciones de Ampliación de Estadística en el plan de estudios.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes:

- OI1: Comprender, interpretar y analizar los métodos estadísticos tratados en esta asignatura, relacionados con el muestreo y la estadística descriptiva.
- OI2: Comprender, interpretar y analizar los métodos estadísticos tratados en esta asignatura, relacionados con la inferencia estadística.
- OI3: Ser capaz de discernir aquellas situaciones en las que es posible y necesario un análisis estadístico.
- OI4: Adquirir aquellos conceptos básicos, resultados y métodos asociados con la estadística que son correquisitos o prerrequisitos de otras asignaturas.
- OI5: Conocer la enorme importancia de la Estadística en la Informática y ser capaz de aplicarla especialmente en dicho ámbito.
- OI6: Desarrollar la capacidad de asimilar nuevas técnicas estadísticas, que puedan ser necesarias en la vida profesional.

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber):

Las habilidades cognitivas se han agrupado por bloques temáticos de la asignatura. Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2, desarrolladas en el artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque 1: Introducción al software estadístico.

- CIC1: Comprender la forma de definir variables atendiendo al software estadístico utilizado SPSS o Statgraphics, para su posterior análisis.
- CIC2: Comprender la utilidad de los distintos operadores: matemáticos, relacionales y lógicos, de generación, de transformación y de selección, siendo capaz de construir variables nuevas o subconjuntos de ellas que sean necesarias para el análisis estadístico susceptible de realizar.

Bloque 2: Introducción al muestreo y Estadística Descriptiva.

- CIC3: Entender la diferencia entre población y muestra.
- CIC4: Comprender la metodología básica para poder realizar un análisis estadístico fiable.
- CIC5: Entender la diferencia entre muestreo aleatorio y no aleatorio.
- CIC6: Entender los distintos tipos básicos de muestreo aleatorio y saber identificar cuál sería la técnica de muestreo más apropiada ante un problema real sencillo.
- CIC7: Entender las distintas técnicas de la estadística descriptiva que permiten analizar y sintetizar la información aportada por un conjunto de datos, a partir de representaciones gráficas, mediante tablas o mediante la obtención de estadísticos de centralización, dispersión o posición.
- CIC8: Saber identificar, a partir de una base de datos real, qué gráficos, tablas o parámetros sintetizarán de forma más apropiada la información que aporta dicha base de datos.

Bloque 3: Inferencia Estadística.

- CIC9: Entender los distintos modelos de distribuciones, sabiendo identificar ante un problema concreto sencillo si se trata de un modelo discreto o continuo.
- CIC10: Entender las diferencias básicas existentes entre los modelos de distribución discretos y continuos a la hora de calcular valores críticos o probabilidades.
- CIC11: Comprender y saber analizar las características principales de un modelo de distribución ya sea discreto o continuo a través de la representación gráfica de su función de cuantía o de densidad, respectivamente, y a través de las funciones de distribución.
- CIC12: Entender el concepto de intervalo de confianza y saber identificar, a partir de una situación real, cuál de los intervalos de confianza vistos en clase será más conveniente en cada caso.
- CIC13: Entender el concepto de contraste de hipótesis y los distintos pasos necesarios para realizar un contraste de hipótesis paramétrico.

- CIC14: Saber identificar, a partir de una situación concreta, cuál de los contrastes de hipótesis vistos en clase para parámetros será más conveniente en cada caso.
- CIC15: Entender el significado de un contraste de homogeneidad o contingencia y los pasos necesarios para realizarlo.
- CIC16: Saber identificar, a partir de una situación concreta, la necesidad de realizar un contraste de homogeneidad o contingencia.
- CIC17: Entender el significado de un test de bondad de ajuste y los pasos necesarios para realizarlo.
- CIC18: Saber identificar, a partir de una situación concreta, la necesidad de realizar un test de bondad de ajuste.
- CIC19: Entender el significado de la realización de un análisis de la varianza (ANOVA) y los pasos necesarios para realizarlo.
- CIC20: Saber identificar, a partir de una situación concreta, la necesidad de realizar un análisis ANOVA.

Capacidades metodológicas (saber hacer):

Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3 introducidas a nivel general en el artículo introductorio de este libro.

Bloque 1: Introducción al software estadístico.

- CIM1: Ser capaz de sintetizar los datos de las fichas de una encuesta en una base de datos para su posterior análisis estadístico con el software Statgraphics.
- CIM2: Ser capaz de sintetizar los datos de las fichas de una encuesta en una base de datos para su posterior análisis estadístico con el software SPSS.
- CIM3: Ser capaz de construir variables nuevas o subconjuntos de ellas que sean necesarias para un análisis estadístico concreto tanto con el SPSS como con el Statgraphics.

Bloque 2: Introducción al muestreo y Estadística Descriptiva.

- CIM4: Ante una situación real concreta, ser capaz de aplicar la metodología a seguir para poder realizar posteriormente un análisis estadístico.
- CIM5: Ser capaz de aplicar las distintas técnicas de muestreo aleatorias vistas en la asignatura.
- CIM6: Ser capaz de aplicar los métodos de la estadística descriptiva, interpretando cualquier patrón significativo, bien sea tratando las variables de una única población o bien mediante comparación entre poblaciones.

Bloque 3: Inferencia Estadística.

- CIM7: Ser capaz de calcular probabilidades y valores críticos ante un modelo de distribución concreto ya sea discreto o continuo, con la ayuda de software estadístico.
- CIM8: Ser capaz de aplicar las técnicas de obtención de intervalos de confianza e interpretar su significado.
- CIM9: Ser capaz de aplicar las técnicas de obtención de contrastes de hipótesis paramétricos e interpretar su significado.
- CIM10: Ser capaz de aplicar los contrastes de homogeneidad y contingencia y los de bondad de ajuste e interpretar su significado.
- CIM11: Ser capaz de aplicar el análisis de la varianza (ANOVA) e interpretar su significado.
- CIM12: Ante una situación real concreta susceptible de tratar mediante un análisis estadístico, se debe ser capaz de interpretar los datos, utilizando en cada caso las técnicas más apropiadas para resumir información o tomar decisiones, sabiendo interpretar los resultados en el contexto que nos ocupe.

Destrezas tecnológicas (saber hacer):

Además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se considera la siguiente:

- CIT1: Manejar con fluidez los paquetes de software SPSS y Statgraphics, que servirán de guía para la realización de actividades relacionadas con la asignatura.

Destrezas lingüísticas (saber hacer):

Además de las destrezas lingüísticas cCIL1 y cCIL2, introducidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje matemático, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso ya sea algorítmico o teórico.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas: las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo: las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

Consideramos las siguientes competencias y contenidos mínimos que el alumnado debe conocer para el buen entendimiento de la asignatura:

- Tener nociones básicas sobre estadística descriptiva.
- Entender el concepto de suceso aleatorio.
- Entender las técnicas básicas de conteo y ser capaz de aplicarlas para obtener la probabilidad de un suceso aleatorio.
- Comprender el concepto de variable aleatoria y saber distinguir los distintos tipos de variables aleatorias.
- Entender el concepto de función de cuantía para variables discretas y su relación con la función de distribución.
- Entender el concepto de función de densidad para variables continuas y su relación con la función de distribución.
- Saber calcular probabilidades en los distintos modelos de distribución, con el uso de tablas.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerrequisitos necesarios para el estudio y entendimiento de esta asignatura se cubren en la asignatura Estadística que se imparte en primero de las tres titulaciones de informática. Sin embargo, el alumnado de esta asignatura no es un conjunto totalmente homogéneo ya que la materia Ampliación de Estadística es troncal en la Ingeniería Técnica en Informática de Gestión, pero optativa en la Ingeniería Técnica en Informática de Sistemas y en la Ingeniería Informática. Esto da lugar a que estén cursando esta asignatura alumnos tanto de segundo como de tercero, cuarto y quinto, dependiendo de la titulación en la que estén matriculados y el curso en el que decidan cursar la asignatura. Esto implica que muchos de ellos tengan ciertos conceptos bastante olvidados o incluso que no

hayan conseguido ciertas competencias mínimas en el curso académico en el que superaron la asignatura Estadística. Tengamos en cuenta que es necesario actualizar periódicamente los contenidos de las asignaturas y en particular el software estadístico. Para paliar estas carencias, tanto en el campus virtual como en la primera clase de la asignatura, se asesorará al alumnado indicándole una bibliografía no muy extensa que debe conocer, previa al entendimiento de la asignatura, así como una lista de problemas relativos a la misma. Además se intenta, en la medida de lo posible, dar a los temas un carácter bastante autocontenido, recordando al principio de cada tema aquellos conceptos que van a necesitar para entenderlo. De esta forma aquellos estudiantes que hayan olvidado estos conceptos podrán recordarlos en clase y si lo creen necesario profundizar en ellos con la ayuda del material bibliográfico.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Introducción al software estadístico.

- Tema 1: Introducción al Statgraphics.
- Tema 2: Introducción al SPSS.

Bloque 2: Introducción al muestreo y Estadística Descriptiva

- Tema 3: Introducción al muestreo.
- Tema 4: Estadística Descriptiva.

Bloque 3: Inferencia Estadística.

- Tema 5: Modelos de distribuciones discretos y continuos.
- Tema 6: Estimación de parámetros poblacionales.
- Tema 7: Contrastes de hipótesis para parámetros poblacionales.
- Tema 8: Análisis inferencial de datos categóricos.
- Tema 9: Análisis de la varianza.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1: Introducción al software estadístico.

Tema 1: Introducción al Statgraphics.

- 1.1. Introducción.
- 1.2. Primeros pasos a través del Statgraphics.
- 1.3. Introducción de datos usando el Statgraphics.
- 1.4. Los operadores del Statgraphics.
- 1.5. Bases de datos a analizar.
- 1.6. Práctica.

Tema 2: Introducción al SPSS.

- 2.1. Introducción.
- 2.2. Primeros pasos a través del SPSS.
- 2.3. Introducción de datos usando el SPSS.
- 2.4. Los operadores del SPSS.
- 2.5. Bases de datos a analizar.
- 2.6. Práctica.

Bloque 2: Introducción al muestreo y Estadística descriptiva

Tema 3: Introducción al muestreo.

- 3.1. Introducción.
- 3.2. Población y Muestra.
- 3.3. Determinación de los objetivos.
- 3.4. Determinación de la muestra.
- 3.5. Diseño del instrumento de recolección.
- 3.6. Obtención de los datos.
- 3.7. Análisis de la información.
- 3.8. Práctica.

Tema 4: Estadística Descriptiva.

- 4.1. Introducción.
- 4.2. Distribución de frecuencias.
- 4.3. Parámetros y estadísticos de centralización.
- 4.4. Parámetros y estadísticos de posición.
- 4.5. Parámetros y estadísticos de dispersión.
- 4.6. Forma.
- 4.7. Gráficos caja.
- 4.8. Práctica.

Bloque 3: Inferencia Estadística.

Tema 5: Modelos de distribuciones discretos y continuos.

- 5.1. Introducción.
- 5.2. Distribuciones de probabilidad discretas.
- 5.3. Distribuciones de probabilidad continuas.
- 5.4. Práctica.

Tema 6: Estimación de parámetros poblacionales.

- 6.1. Introducción.
- 6.2. Intervalos de confianza para la media y la desviación típica.
- 6.3. Intervalos de confianza para proporciones.
- 6.4. Intervalos de confianza para la diferencia de medias y el cociente de varianzas.
- 6.5. Tamaño de la muestra y error de estimación.
- 6.6. Práctica.

Tema 7: Contrastes de hipótesis para parámetros poblacionales.

- 7.1. Introducción.
- 7.2. Pasos a realizar en un contraste de hipótesis.
- 7.3. Contrastes de hipótesis bajo normalidad.
- 7.4. Contrastes de hipótesis para muestras grandes.
- 7.5. Práctica.

Tema 8: Análisis inferencial de datos categóricos.

- 8.1. Introducción.
- 8.2. Contraste de homogeneidad y contingencia.
- 8.3. El test de bondad de ajuste Ji-cuadrado.
- 8.4. Práctica.

Tema 9: Análisis de la varianza.

- 9.1. Introducción.
- 9.2. Análisis de la varianza de un factor.
- 9.3. Robustez de las hipótesis del modelo.
- 9.4. Método de la diferencia mínima de Fisher.
- 9.5. Práctica.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente se ha desarrollado en el artículo introductorio de este libro desde el punto de vista general para el segundo curso de la titulación de informática. Dicho desarrollo, aunque genérico, se considera válido para el caso particular de la asignatura Ampliación de Estadística. Resumiendo, podemos decir que aunque los nuevos modelos educativos propician el pensamiento creativo, enseñando a aprender por encima de enseñar conocimientos, no hay que menospreciar un modelo en el que la clase magistral tiene un papel importante pero no exclusivo en la transmisión de conocimientos, de forma que este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas de laboratorio y las actividades en grupos pequeños que jugarán un papel fundamental. Las actividades que se proponen son:

- **Clases de teoría con apoyo de material audiovisual:** en lo que se refiere a las clases de teoría, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía sobre los contenidos más importantes de la asignatura. Además, algunos profesores de la asignatura hemos editado un libro que incluye todos los contenidos de la misma. De esta forma, el alumnado no tiene que estar tan preocupado por la toma de apuntes en las clases teóricas, pero sí en entender y asimilar lo que se le está explicando. Dichas explicaciones teóricas se intercalarán con la realización de problemas, ejemplos prácticos y aplicaciones, siempre que el contenido lo requiera.

- **Actividades en grupos pequeños / tutorías docentes:** estas actividades estarán relacionadas con la realización de problemas y cuestiones teórico-prácticas vinculadas con la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos a situaciones reales concretas y fomentar la capacidad de análisis, síntesis y autoevaluación del alumnado.
- **Prácticas de laboratorio:** en cuanto a las prácticas de laboratorio, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía para aprender a utilizar el correspondiente software estadístico e interpretar los resultados obtenidos. Actualmente se utilizan dos paquetes de software estadístico: el Statgraphics y el SPSS; sin embargo, se intenta incidir en que lo más importante no es el software que se utilice sino el saber interpretar los resultados obtenidos, ya que el software estadístico está en constante cambio y desarrollo.
- **Trabajos complementarios:** en cuanto a los trabajos complementarios, comentar que dichos trabajos incidirán en la nota final de la asignatura y pueden ser de índole teórica, de índole práctica o de índole teórico-práctica, y deberán realizarse de forma individual.
- **Tutorías de atención al alumnado:** el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como cuestiones concretas sobre el contenido de la asignatura. Además de dichas tutorías individualizadas, se programarán varias tutorías en grupo, al menos una para cada bloque de la asignatura correspondiente.

5.2 ESTRATEGIAS DE APRENDIZAJE

Las estrategias de aprendizaje se han establecido a nivel común en el artículo introductorio de este libro. Tal y como se indicó, los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador no son los únicos sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web y el campus virtual ofrecen innumerables posibilidades que no hay que dejar pasar. Así, se ha elaborado una página de la asignatura, que incluye toda la información que el alumno necesita. El uso de la misma ha sido mayoritario en las experiencias llevadas a cabo hasta el momento.

Entre otras cosas, en dicha página podemos encontrar:

- **Tablón de anuncios:** desde aquí, el alumnado puede estar perfectamente informado de cualquier novedad relacionada con la asignatura. Además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, etc.
- **Tutorías:** indica el horario de atención al alumnado.
- **Objetivos:** resume los objetivos que se pretenden alcanzar en esta asignatura.
- **Temario:** especifica el temario de esta asignatura.

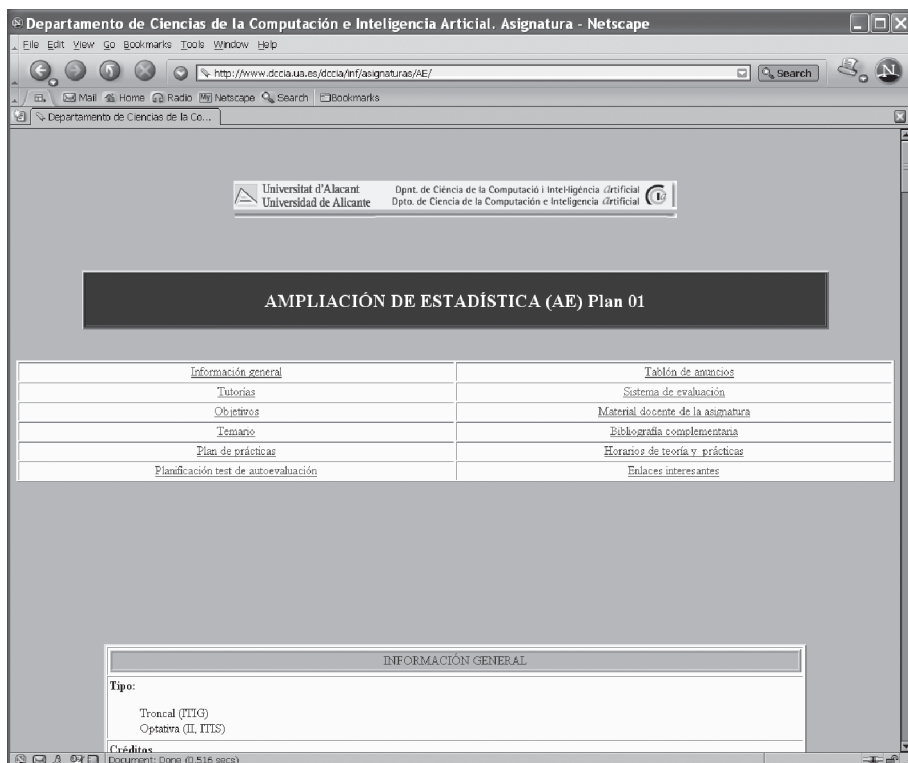


Ilustración 2. *Página de la asignatura Ampliación de Estadística*

- Material docente de la asignatura: aquí se puede encontrar todo el material que como mínimo va a ser necesario para el seguimiento de las clases teóricas y prácticas.
- Bibliografía básica y complementaria: aparece un listado de bibliografía que pueden consultar para profundizar en la asignatura o preparar los trabajos complementarios.
- Plan de prácticas: se especifica semana a semana qué trabajo debe realizar el alumnado tanto en las horas presenciales como en las no presenciales de forma autónoma.
- Horarios de teoría y prácticas: contiene información sobre los grupos de teoría y prácticas, así como el profesorado que los imparte.
- Enlaces de interés: aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en algunos contenidos de la materia.
- Planificación de los tests de autoevaluación: en el campus virtual se publicará un ejercicio de autoevaluación por cada tema mediante el cual se podrá medir el grado de asimilación obtenido. En este apartado se recuerda el periodo en el que estará activo cada uno de los tests.

En base a todo esto, la estrategia de aprendizaje que se propone se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura.

Planificación de las clases teóricas:

- Lectura previa del guión correspondiente a la sesión de teoría que se trate.
- Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar primero contrastarlo con otros compañeros o utilizar la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida entre todos en el aula, o por el profesor fuera del aula.
- Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Planificación de las clases prácticas:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- Parte de las prácticas se realizarán en los laboratorios y parte en horas no presenciales. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá con bastante celeridad dichas prácticas, indicando, una vez corregidas, los fallos más comunes. Cada estudiante de forma individual debe analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario se pedirá ayuda al profesor correspondiente.

Autoevaluación: una vez realizadas todas las actividades previas relacionadas con un tema concreto, el estudiante debe discernir si cree que dicho tema ha sido totalmente entendido. En caso de no ser así, debe incidir en el estudio de los contenidos que crea tener más flojos, utilizando si lo cree conveniente las tutorías y

realizando algunos problemas de ampliación, bien de los propuestos en las hojas de problemas o bien haciendo uso de la bibliografía. Cuando crea estar preparado, puede realizar el ejercicio de autoevaluación del tema correspondiente, publicado en el campus virtual.

Evaluación final: si el resultado de los ejercicios de autoevaluación y de las prácticas propuestas ha sido satisfactorio, el estudiante habrá superado la asignatura. En caso contrario, tendrá que profundizar más en los contenidos de la asignatura y realizar el examen final de la asignatura.

De forma opcional se podrá hacer trabajos complementarios de manera individual, para subir la nota siempre y cuando el trabajo realizado a lo largo del curso se considere satisfactorio.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial, y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES				
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos pequeños/tutorías docentes
Presentación		1	0	1
BLOQUE 1	Tema 1	0	1	0
	Tema 2	0	1	0
BLOQUE 2	Tema 3	2	1	1
	Tema 4	1	2	2
BLOQUE 3	Tema 5	2	2	2
	Tema 6	2	2	2
	Tema 7	2	2	2
	Tema 8	2	2	2
	Tema 9	1	1	1
Preparación del examen final		2	1	2
Examen final:		1,5	0	0
TOTAL: 46,5		16,5	15	15

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES				
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños/ tutorías docentes
Presentación		1,75	0	0,25
BLOQUE 1	Tema 1	0	2	0
	Tema 2	0	2	0
BLOQUE 2	Tema 3	3,5	2	0,25
	Tema 4	1,75	4	0,5
BLOQUE 3	Tema 5	3,5	4	0,25
	Tema 6	3,5	4	0,25
	Tema 7	3,5	4	0,25
	Tema 8	3,5	4	0,25
	Tema 9	1,75	0	0
Preparación del examen final		3,5	4	1,75
Tutorías:				
TOTAL: 60		26,25	30	3,75

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

Las horas no presenciales de la sesión de presentación estarán dedicadas a la recopilación de la documentación de la asignatura y, en su caso, al repaso de aquellos prerrequisitos que no se hayan alcanzado. La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y entendimiento de los enunciados de los problemas, y revisión de los problemas ya corregidos por el profesorado. De las horas no presenciales dedicadas a las clases de teoría se utiliza una por cada tema para la realización del ejercicio de autoevaluación.

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

- Cochran, W.G. (1987) *Técnicas de muestreo*. CECSA.
- DeGroot, M.H.; Addison-Wesley (1988) *Probabilidad y Estadística Iberoamericana*.
- Hanke, J. y Reitsh, A. (1997) *Estadística para negocios* Mc Graw-Hill.
- Lipschutz, S. y Schiller, J. (1999) *Introducción a la Probabilidad y Estadística* McGraw-Hill.
- López Cachero, M. (1992) *Fundamentos y métodos de Estadística*. Pirámide.
- Mate, C. (1995) *Curso general sobre Statgraphics*, Vol. 1 y 2.. UPCO.
- Migallón, V. y Penadés, J. (2000) *Estadística II*. Ramón Torres.

Montgomery, D.C. y Runger, G.C. (1998) Probabilidad y Estadística aplicadas a la Ingeniería McGraw-Hill.

Pérez, C. (2004) *Técnicas Estadísticas con SPSS*. Prentice Hall.

Walpole, R.E.; Myers, R.H. y Myers, S. (1999) *Probabilidad y Estadística para Ingenieros*. Prentice Hall.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Casas, J.M.; García, C.; Rivera, L. y Zamora, A. (1998) *Problemas de Estadística (Descriptiva, Probabilidad e Inferencia)* Ediciones Pirámide.

Cuadras, C.M. (1995) *Problemas de Probabilidades y Estadística*. Vol. 1.EUB, 2ª ed.

García Ferrando, M. (1999) *Socioestadística*. Alianza Universidad Textos.

Peña, D. (1991). *Estadística. Modelos y Métodos*, vol. 1 Alianza Universidad.

Peña, D. y Romo, J. (1997) Introducción a la Estadística para las Ciencias Sociales McGraw-Hill.

Pérez, C. (2002) *Estadística aplicada a través de excel* Prentice Hall.

Ruiz-Maya, L. (1986) *Problemas de Estadística* Editorial AC.

7.3 OTROS RECURSOS

Como se ha mencionado ya, en esta asignatura el alumnado dispone de un guión y de un libro de texto, del que se ayuda en las clases de teoría. Además, dispone en la red de un cuaderno con todas las actividades que debe realizar y unas diapositivas de apoyo explicando cómo utilizar el software estadístico para aplicar los distintos métodos estadísticos tratados en esta asignatura, así como su interpretación. Pero además de estos recursos básicos, incluimos aquí otros recursos de especial interés, tanto para el profesorado como el alumnado.

Publicaciones periódicas:

- American Statistical Association Journals
- Applied Mathematics and Computation
- Biometrics
- Biometrika
- Computational Mathematics and Modeling
- Computational Statistics
- Computers and Industrial Engineering
- Journal of Applied Probability
- Journal of Applied Statistics
- Journal of Computational and Graphical Statistics
- Journal of the Royal Statistical Society
- Journal of Statistical Computation and Simulation
- Journal of Statistical Software

- Journal of Theoretical Probability
- SIAM Journals
- SIAM Journal on Scientific Computing
- Statistics and Computing
- Statistics & Probability Letters
- Uncertainty in Artificial Intelligence
- Videre: A Journal of Computer Vision Research

Grupos de noticias:

- sci.math
- sci.math.research
- sci.stat.math
- sci.stat.edu

Los dos primeros tratan temas relacionados con las matemáticas en general, mientras que el resto están dedicados a la estadística en particular.

Asociaciones y grupos de interés:

- AMS: American Mathematical Society. <http://www.ams.org>
- NCTM: National Council of Teachers of Mathematics. <http://www.nctm.org>
- MAA: Mathematical Association of America. <http://www.maa.org>
- SIAM: Society for Industrial and Applied Mathematics. <http://www.siam.org>
- RSME: Real Sociedad Matemática Española. <http://www.mat.ucm.es/rsme>
- EMS: European Mathematical Society. <http://www.emis.de>
- The Math Forum. <http://forum.swarthmore.edu>
- ACM: Association for Computing Machinery. <http://www.acm.org>
- IEEE: Institute of electrical and Electronics Engineers. <http://www.ieee.org>
- IEEE Computer Society. <http://www.computer.org>

Páginas web de interés:

Algunas páginas Web muy interesantes, que proporcionan enlaces o recopilaciones de enlaces a recursos estadísticos (organizaciones, revistas electrónicas, conferencias, bibliografía, tutoriales, software estadístico, etc.) son las siguientes:

- <http://es.geocities.com/riotorto/> (Biomates)
- <http://www.analyse-it.com/download/dl.asp> (Software estadístico para excel)
- <http://www8.madrid.org/iestadis/fijas/otros/comic.htm> (La estadística en cómic)

- <http://www.york.ac.uk/depts/maths/histstat/people/welcome.htm> (Portraits of Statisticians)
- <http://www.clarkson.edu/~dobrowb/probweb/probweb.html> (the Probability Web)
- <http://members.aol.com/johnp71/javastat.html> (Web Pages that Perform Statistical Calculations)
- <http://www.stat.ufl.edu/vlib/statistics.html> (The World Wide Web Virtual Library: Statistics)
- <http://www.loria.fr/equipes/polka> (Polynomials, Combinatorics, Arithmetic)
- <http://mally.eco.rug.nl/biblio/SPlist.html> (Stochastic Programming Bibliography)

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces es fácil la resolución de dudas de esta forma. No obstante, cuando la duda es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

8.1 PROCEDIMIENTOS DE EVALUACIÓN

Para evaluar esta asignatura intentamos acercarnos lo más posible a la evaluación continua a partir de las distintas actividades propuestas a lo largo del curso. Concretamente, a aquellos alumnos que asistan a clase se les realizará un seguimiento de los conocimientos adquiridos y de la realización de las distintas actividades y tests de autoevaluación, en el horario de clase, así como de la actitud presentada en las clases. Atendiendo a esto, la nota de la asignatura se desglosa de la siguiente forma:

NOTA DE AMPLIACIÓN DE ESTADÍSTICA	
Prácticas de los temas 1, 2, 3 y 4	20%
Prácticas de los temas 5 y 6	30%
Prácticas de los temas 7 y 8	30%
Test de autoevaluación del tema 3	5%
Test de autoevaluación del tema 4	5%
Test de autoevaluación del tema 5	5%
Test de autoevaluación del tema 6	5%

Cuadro 4. *Procedimientos de evaluación*

Siempre que la nota obtenida con los trabajos anteriores sea mayor o igual que 5, podrá ser aumentada hasta en un punto atendiendo a la actitud presentada en la asignatura: se valorará entre otros, la asistencia a prácticas y a las tutorías en grupo, llevar el trabajo al día, presentación de las prácticas entregadas y realización de los tests o trabajos voluntarios.

Los alumnos que no asistan a clase deberán hacer un examen final y entregar todas las prácticas y actividades realizadas a lo largo del curso al comienzo del mismo. La nota del examen tendrá un peso del 80% y la del resto de actividades un 20%. Pero sólo se superará la asignatura si en ambas partes se obtiene una nota mayor o igual que 5 sobre 10.

8.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas y procedimientos estadísticos relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos estadísticos y de resolución de problemas se ajustan a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre estadística cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. Los problemas y procedimientos estadísticos relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos estadísticos y de resolución de problemas son generalmente ajustados a la naturaleza del problema. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de los resultados aceptables. • La actuación en las destrezas transferibles es generalmente buena. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.

Calificación	Rango	Pautas
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas y procedimientos estadísticos relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio estándares son usualmente desarrolladas con éxito razonable aunque el significado y análisis de los resultados pueden no ser entendidos completamente. • Las destrezas transferibles están a un nivel básico. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspense	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas y procedimientos estadísticos relacionados con la asignatura no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente y el significado y análisis de los resultados no son entendidos generalmente. • Las destrezas transferibles están a un nivel deficiente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 5. *Criterios de evaluación*

Queremos hacer notar que para obtener una matrícula de honor es necesario obtener un sobresaliente alto y hacer un trabajo complementario de calidad.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Ampliación de Estadística. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI1	De CIC1 a CIC8 cCIC1 cCIC2	1,2	3,4		Enseñanza presencial (Lección magistral/Trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al muestreo y la estadística descriptiva.
OI2	De CIC9 a CIC20 cCIC1 cCIC2	1,2		5, 6,7, 8,9	Enseñanza presencial (Lección magistral/trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a la inferencia estadística.
OI3	De CIC1 a CIC20	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.

Cuadro 6. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI4	De CIC1 a CIC20 cCIC1	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
OI6	De CIC1 a CIC20 cCIC1	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
cOI6	De CIC1 a CIC20	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral /trabajo de aula en grupos). Enseñanza no presencial (Aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI5	CIM1 CIM2 CIM3 CIM6 CIM12 cCIM1 cCIM2 CIT2 CIT4 CIT5		3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral). Enseñanza no presencial (Aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOI1	De CIM1 a CIM12 CIT1 cCIT1	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental., eficiencia y precisión en la resolución de procesos estadísticos y la resolución del problema.
cOI2	CIT1 cCIT1	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/ realización de prácticas propuestos). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI3	CIL1 cCIL1	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral /trabajo de aula en grupos). Enseñanza no presencial (Aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de rigurosidad en las explicaciones de los procedimientos aplicados.

Cuadro 8. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
cOI4	cCIL2	1,2	3,4	5, 6,7, 8,9	Enseñanza presencial (Lección magistral). Enseñanza no presencial (Aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de conocimiento de los distintos términos estadísticos, en castellano y/o valenciano y en inglés.
cOI5	De CIM1 a CIM12 cCIM3	1,2	3,4	5, 6,7, 8,9	Enseñanza no presencial (Aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 9. *Análisis de coherencia: competencias instrumentales (saber hacer)*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
cOIP1	CIPTC1 cCIPTC1 cCIPTC2		3,4	5, 6,7, 8,9	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3		3,4	5, 6,7, 8,9	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4		3,4	5, 6,7, 8,9	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 10. Análisis de coherencia: competencias instrumentales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
cOS1	cCS1 cCS4 cCS5		3,4	6,7, 8,9	Enseñanza presencial (Lección magistral/trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3		3,4	6,7, 8,9	Enseñanza presencial (Lección magistral/trabajo de aula en grupos). Enseñanza no presencial (Aprendizaje on-line/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos estadísticos aplicados.
cOS3	cCS1 cCS2 cCS4 cCS5		3,4	6,7, 8,9	Enseñanza presencial (Lección magistral/trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2		3,4	6,7, 8,9	Enseñanza presencial (Lección magistral/trabajo de aula en grupos). Enseñanza no presencial (Aprendizaje on-line/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos estadísticos aplicados.

Cuadro 11. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE ARQUITECTURAS DE COMPUTADORES

José García Rodríguez; Antonio Soriano Payá; Andrés Fuster Guillo;
José Antonio Serra Pérez

*Departamento de Tecnología Informática y Computación
Universidad de Alicante
(jgarcia; soriano; fuster; jaserra)@dtic.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

En el desarrollo de esta propuesta docente se han considerado como referencias principales el libro blanco del título de grado de Ingeniería Informática y las recomendaciones del Computing Curricula 2001 [ACM01b], realizado conjuntamente por el IEEE y la ACM como resumen de diversos documentos previos elaborados por estas y otras organizaciones de prestigio en las dos décadas previas. Los tres perfiles profesionales que responden a las tendencias profesionales y que engloban diferentes propuestas son: perfil de desarrollo software, perfil de sistemas y perfil de gestión y explotación de tecnologías de la información. En el perfil de desarrollo software se puntualiza la necesidad de conocimientos hardware debido a la influencia del hardware en las soluciones software de diversas áreas de aplicación. El perfil de sistemas es inherente a los objetivos de capacitación en el análisis, diseño, construcción e implementación de sistemas basados en computadoras de propósito general. Finalmente, en el perfil de gestión y explotación de tecnologías de la información, el profesional debe dirigir el diseño de soluciones de sistemas informáticos para sus clientes con los productos hardware y software disponibles, estando los sistemas informáticos formados por productos muy diversos como procesadores, redes, software de aplicaciones y sistemas por lo que debe ser experto en alguno de estos productos y a menudo colaborar en equipos que ofrezcan una solución completa al cliente.

En lo que respecta a la legislación vigente en España, debemos observar las directrices generales propias de las titulaciones de Informática (Reales decretos 1459/1990, 1460/1990, 1461/1990 de 26 de octubre, BOE 1990). En el caso de la Universidad de Alicante, y en la titulación de Ingeniería Técnica en Informática de Gestión, aparecen las asignaturas de Informática básica como troncal con una asignación de 6 créditos teóricos y 6 créditos prácticos. También, como materia troncal, tenemos la asignatura de Arquitecturas de Computadores, con una asignación de 4,5 créditos prácticos y 3 créditos teóricos.

En lo que respecta a los contenidos, el Computing Curricula 2001 recomendaba 36 horas lectivas para la impartición del área de Arquitectura y Organización, dividiéndola en 9 unidades de conocimiento: lógica digital y sistemas digitales, representación de datos a nivel máquina, organización de la máquina a nivel ensamblador, organización y arquitectura de los sistemas de memoria, interfaces y comunicación, organización funcional, multiprocesamiento y arquitecturas alternativas, mejoras de rendimiento y arquitecturas para redes y sistemas distribuidos.

Es por tanto evidente la necesidad de cubrir los conceptos de estructuras y arquitecturas de computadores en cualquiera de los perfiles profesionales propuestos, en particular, para los tres perfiles que apunta el libro blanco de grado de Ingeniería Informática.

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: para el diseño de software es necesario conocer la estructura y arquitectura del computador para obtener el máximo rendimiento de las máquinas sobre las que se ejecutan las aplicaciones.
- Arquitectura y diseño de software: en el diseño de software para sistemas compartidos es necesario conocer cómo funcionan las arquitecturas hardware distribuidas.
- Diseño multimedia: la programación multimedia se apoya en instrucciones específicas de lenguaje máquina que ofrecen los procesadores orientados a acelerar operaciones multimedia.

Perfil profesional de sistemas:

- Ingeniería de comunicación de datos: en dicho perfil profesional es básico conocer las técnicas de transferencia de datos entre computadores.
- Diseño de redes de comunicación: es necesario conocer los dispositivos hardware base de estas redes.
- Asistencia técnica: en este perfil se debe tener conocimientos generales sobre configuraciones hardware de equipos informáticos.

- Ingeniería de integración y pruebas e implantación y pruebas: en la realización de pruebas para elegir la mejor opción de entre diversas arquitecturas hardware sobre las que ejecutar las aplicaciones.

Perfil profesional de gestión y explotación de tecnologías de la información:

- Consultoría de empresas de TI: es necesario saber analizar las necesidades hardware que conlleva la implantación de las TIC en una empresa.
- Especialista en sistemas: en dicho perfil es necesaria la habilidad en la gestión y manejo de los sistemas informáticos de la empresa.
- Desarrollo de investigación y tecnología: los cambios continuos en la tecnología disponible requieren que el profesional disponga de una base sólida sobre las arquitecturas para analizar y si es preciso desarrollar o adaptar nuevos productos.
- Dirección de TIC (marketing, proyectos, dirección general): este perfil requiere un conocimiento amplio del conjunto de áreas tecnológicas de las Tecnologías de la Información y las Comunicaciones (TIC).

Esta adecuación de los tópicos relacionados con la asignatura Arquitecturas de computadores a los distintos perfiles profesionales queda resumida en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimiento y habilidad en la optimización del código para un aprovechamiento óptimo de los recursos hardware. • Conocimiento y habilidad en el manejo de arquitecturas distribuidas para un diseño adecuado del software. • Conocimiento y habilidad en el uso de las instrucciones máquina específicas para multimedia de proporcionan los procesadores. • Conocimiento y habilidad en el manejo de la entrada/salida de los computadores y los métodos de transferencia de datos. • Conocimiento y habilidad para la configuración y uso de los dispositivos de comunicaciones entre ordenadores. • Conocimiento y habilidad en la configuración de equipos informáticos. • Conocimiento y habilidad para el análisis de rendimiento de los sistemas sobre diferentes arquitecturas. • Conocimiento y habilidad en el análisis de necesidades hardware para el acceso a las TIC de las empresas.
Arquitectura y diseño de software	
Diseño multimedia	
Ingeniería de comunicación de datos.	
Diseño de redes de comunicación	
Asistencia técnica	
Ingeniería de integración y pruebas e implantación y pruebas	
Consultoría en empresas de TI	

Cuadro 1. *Adecuación de los tópicos relacionados con a asignatura Arquitectura de Computadores*

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimiento y habilidad en la gestión y mantenimiento de los sistemas informáticos.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento y habilidad en el análisis y desarrollo de arquitecturas de alto rendimiento.
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento y habilidad en la consultoría y administración de sistemas informáticos.

Cuadro 2. *Adecuación de los tópicos relacionados con la asignatura Arquitectura de Computadores*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Arquitecturas de computadores forma parte del segundo curso de Ingeniería Técnica en Informática de Gestión como asignatura troncal. Tiene una relación directa con la asignatura **Informática básica** de primer curso donde se imparten conceptos sobre sistemas combinatorios y secuenciales, representación de la información y de tecnología de computadores básicos para poder abordar el estudio de las estructuras y arquitecturas de computadores.

Además de estas relaciones, en cursos más adelantados existen otras asignaturas que entre sus tópicos incluyen temas más avanzados en los que se necesitan conocimientos sobre las arquitecturas de computadores:

- **Sistemas operativos:** asignatura troncal de tercer curso donde se abordan los conceptos fundamentales de los sistemas operativos, relacionándolos con las arquitecturas de computadores.
- **Redes:** asignatura obligatoria de tercer curso donde se introducen los conceptos de redes de computadores y transmisión de datos.
- **Administración de servicios de Internet:** asignatura optativa donde se introducen conceptos sobre tecnologías y arquitecturas de red, así como el diseño, implantación y configuración de servicios de red como correo, web, sistemas de archivo remoto o servidores de aplicaciones.
- **Administración e instalación de redes de computadores:** es una asignatura optativa en la que se abordan conceptos sobre infraestructuras de red.
- **Domótica y edificios inteligentes:** asignatura optativa en la que se muestran conceptos sobre edificios inteligentes y su ámbito de aplicación y se estudian estándares y tecnologías de comunicación en edificación.
- **Sistemas industriales:** asignatura optativa en la que se estudian tecnologías basadas en computador para la resolución de tareas de control y monitorización de procesos industriales.
- **Mantenimiento del computador:** asignatura optativa en la que se presentan técnicas de instalación, mantenimiento y testeo de computadores.

- **Tecnología y arquitectura robótica:** asignatura optativa que introduce al alumno al diseño de robots para diferentes aplicaciones.

La siguiente figura resume las relaciones anteriormente descritas:

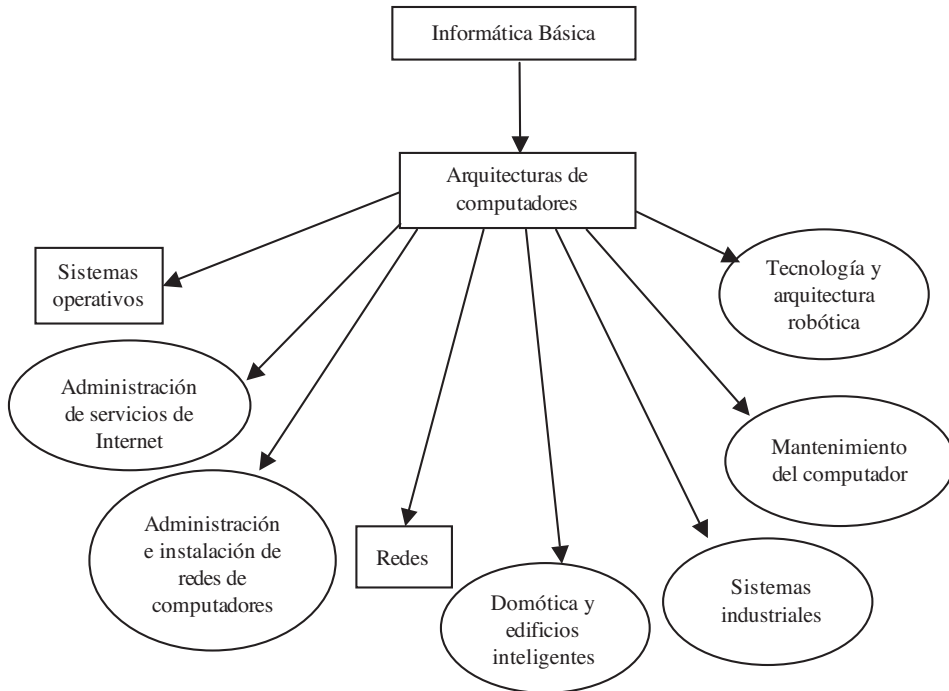


Ilustración 3. Relaciones de Arquitectura de Computadores en el plan de estudios

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro, como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos:

- OI1: Identificar y comprender la organización interna de un computador, estructurándolo en sus unidades funcionales.
- OI2: Analizar el funcionamiento de las diferentes unidades funcionales de un computador.
- OI3: Identificar los elementos funcionales que intervienen en la ejecución de las instrucciones determinando cómo deben conectarse entre sí dichos elementos para que fluya la información.

- OI4: Identificar y evaluar los diferentes sistemas de almacenamiento que es posible encontrar en un computador, estableciendo una clasificación adecuada.
- OI5: Comprender la necesidad del sistema de entrada/salida y analizar las funciones que realiza.
- OI6: Comprender la necesidad de los buses y analizar los tipos más adecuados, dependiendo de las unidades a conectar.
- OI7: Comprender el concepto de arquitectura y evaluar las técnicas de mejora de rendimiento.

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber): además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2 desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque 1: Estructura de computadores.

- CIC1: Comprender la diferencia entre los conceptos de estructura y arquitectura de computadores.
- CIC2: Profundizar en el conocimiento de los elementos funcionales que componen el computador e identificar las funciones que realizan.
- CIC3: Identificar los elementos funcionales y las conexiones que se requieren en la ejecución de una instrucción.
- CIC4: Entender el proceso de ejecución de una instrucción, las fases en la que se divide y las señales que debe activar la unidad de control
- CIC5: Comprender el funcionamiento y las diferencias entre unidades de control microprogramadas y cableadas.

- CIC6: Identificar los bloques estructurales de una unidad aritmético-lógica, así como sus entradas y salidas generales.
- CIC7: Evaluar diferentes circuitos aritméticos para enteros, haciendo hincapié en la importancia de los retardos de propagación.
- CIC8: Comprender el funcionamiento y la problemática de los operadores en coma flotante.
- CIC9: Analizar la necesidad de los sistemas de memoria, comprendiendo los conceptos fundamentales y reflexionando sobre la necesidad de establecer una jerarquía.
- CIC10: Comprender la necesidad de disponer de un sistema de entrada/salida en un computador y valorar las funciones que realiza.
- CIC11: Evaluar las diferentes técnicas de transferencia de datos.
- CIC12: Comprender la necesidad de disponer de diferentes buses para unir los diferentes elementos del computador y valorar la necesidad de una jerarquía de buses
- CIC13: Analizar el impacto de las diferentes topologías de conexión y protocolos de comunicación usados en buses en el rendimiento de los sistemas

Bloque 2: Arquitectura de computadores y mejora de rendimiento.

- CIC14: Comprender las implicaciones del diseño de un repertorio de instrucciones adecuado al propósito de la máquina en términos de coste y rendimiento.
- CIC15: Analizar diferentes alternativas de diseño y valorar la necesidad de utilizar programas para la evaluación del rendimiento de diferentes propuestas de arquitectura

Capacidades metodológicas (saber hacer): Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3, introducidas a nivel general en artículo introductoria de este libro.

Bloque 1: Estructura de computadores.

- CIM1: Aplicar los conocimientos sobre las unidades funcionales del computador al caso concreto de computadores comerciales o de propósito educacional.
- CIM2: Ser capaz de construir una unidad de control capaz de ejecutar un conjunto de instrucciones sobre una ruta de datos cualquiera, definiendo las fases de ejecución y permitiendo la activación de las señales necesarias.
- CIM3: Aplicar los conocimientos de sobre operadores aritméticos básicos para el diseño de operaciones más sofisticadas.

- CIM4: Ser capaz de extrapolar los conocimientos de aritmética entera para resolver problemas de aritmética en coma flotante.
- CIM5: Ser capaz de diseñar mapas de memoria principal y diseñar la circuitería necesaria para una correcta decodificación del sistema de memoria.
- CIM6: Aplicar los conocimientos de entrada/salida y técnicas de transferencia de datos para el análisis de rendimiento de diferentes alternativas de gestión de la entrada/salida.
- CIM7 Ser capaz de diseñar un sistema de buses adecuado para un computador estableciendo una jerarquía adecuada.

Bloque 2: Arquitectura de computadores y mejora de rendimiento.

- CIM8: Ser capaz de analizar el rendimiento de diferentes arquitecturas y valorar la mejor opción para el problema planteado, apoyándose en el conocimiento de diferentes benchmarks.

Destrezas tecnológicas (saber hacer): además de la destreza tecnológica cCIT1 introducida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar cualquier entorno de programación en ensamblador para desarrollar programas en código máquina.

Destrezas lingüísticas (saber hacer): además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir rigor en las exposiciones y demostraciones, tanto a nivel oral como escrito, siendo riguroso en las explicaciones de cualquier proceso ya sea algorítmico o teórico.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas: las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo: las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Conocer los diferentes sistemas de numeración y la conversión de números entre diferentes bases.
- Tener nociones generales de puertas lógicas y saber emplearlas adecuadamente para construir circuitos combinacionales que implementen una función determinada.
- Saber operar en aritmética binaria sobre números enteros y en coma flotante.
- Comprender las relaciones entre tiempo y frecuencia y entre las diferentes unidades de medida de capacidad de almacenamiento.
- Comprender la arquitectura von Neumann y la relación entre las diferentes unidades funcionales de un computador.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerrequisitos necesarios para el estudio y entendimiento de esta asignatura se cubren en la asignatura Informática básica. Sin embargo, por la entidad de esta asignatura y por tratarse de una asignatura de primer curso, los alumnos no llegan a asimilar en el grado esperado diversos conceptos básicos como las conversiones de sistemas de numeración binario a hexadecimal, operaciones de aritmética binaria, puertas lógicas, conversiones entre unidades de medida de almacenamiento o de frecuencias de funcionamiento, o conocimiento básico sobre los bloques funcionales del computador. Esto implica la necesidad de repasar, al menos de manera rápida, determinados conceptos que debían suponerse adquiridos en el primer curso en el momento de aplicar los citados conceptos en la asignatura.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Estructura de computadores.

- Tema 1: Introducción a la estructura de computadores.

- Tema 2: Unidad de control.
- Tema 3: Unidad aritmético-lógica.
- Tema 4: Unidad de memoria.
- Tema 5: Unidad de entrada/salida.
- Tema 6: Buses.

Bloque 2: Arquitectura de computadores y análisis de rendimiento.

- Tema 7: Introducción a la arquitectura de computadores. Parámetros de rendimiento.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1: Estructura de computadores.

- Tema 1: Introducción a la estructura de computadores.
 - 1.1. Conceptos generales.
 - 1.2. Estudio de computador ejemplo.
- Tema 2: Unidad de control.
 - 2.1. Introducción.
 - 2.2. Fases en la ejecución de una instrucción.
 - 2.3. Unidad de control.
 - 2.4. Unidad de control cableada.
 - 2.5. Unidad de control microprogramada
- Tema 3: Unidad aritmético-lógica.
 - 3.1. Estructura general de la ALU.
 - 3.2. Sumadores y restadores.
 - 3.3. Multiplicadores.
 - 3.4. Divisores.
 - 3.5. Otros operadores.
 - 3.6. Operaciones en coma flotante.
- Tema 4: Unidad de memoria.
 - 4.1. Conceptos básicos.
 - 4.2. Jerarquía de memoria.
 - 4.3. Memoria principal. Diseño de mapas.
 - 4.4. Caché.
 - 4.5. Otras memorias.
- Tema 5: Unidad de entrada/salida.
 - 5.1. Sistema de entrada/salida.
 - 5.2. Dispositivos.
 - 5.3. Controladores.
 - 5.4. Técnicas de transferencia de datos.
 - 5.5. Procesadores.

- Tema 6: Buses.
 - 6.1. Introducción y conceptos básicos.
 - 6.2. Temporización y protocolos de arbitraje.
 - 6.3. Jerarquía de buses.
 - 6.4. Descripción de buses comerciales.

Bloque 2: Arquitectura de computadores y análisis de rendimiento.

- Tema 7: Introducción a la arquitectura de computadores. Parámetros de rendimiento.
 - 7.1. Introducción.
 - 7.2. Diseño de computadores.
 - 7.3. Análisis de rendimiento.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente se ha desarrollado en el artículo introductorio de este libro desde el punto de vista general para el segundo curso de la titulación de informática. Dicho desarrollo, aunque genérico, se considera válido para el caso particular de la asignatura Arquitecturas de computadores. Resumiendo, podemos decir que aunque los nuevos modelos educativos propician el pensamiento creativo, enseñando a aprender por encima de enseñar conocimientos, no hay que menospreciar un modelo en el que la clase magistral tiene un papel importante pero no exclusivo en la transmisión de conocimientos, de forma que este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas de laboratorio y las actividades en grupos pequeños que jugarán un papel fundamental. Las actividades que se proponen son:

- **Clases de teoría con apoyo de material audiovisual:** en lo que se refiere a las clases de teoría, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía sobre los contenidos más importantes de la asignatura. Además, algunos profesores de la asignatura hemos editado un libro que incluye todos los contenidos de la misma. De esta forma, el alumnado no tiene que estar tan preocupado por la toma de apuntes en las clases teóricas, pero sí en entender y asimilar lo que se le está explicando. Dichas explicaciones teóricas se intercalarán con la realización de problemas, ejemplos prácticos y aplicaciones siempre que el contenido lo requiera.
- **Actividades en grupos pequeños / tutorías docentes:** estas actividades estarán relacionadas con la realización de problemas y cuestiones teórico-prácticas relacionadas con la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos a situaciones reales concretas y fomentar la capacidad de análisis, síntesis y autoevaluación del alumnado.

- **Prácticas de laboratorio:** en cuanto a las prácticas de laboratorio, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía para aprender a utilizar el correspondiente software de simulación utilizado. Actualmente se realizan prácticas con el simulador Manotas que dispone de un sencillo lenguaje ensamblador y permite trazar la ejecución de las instrucciones de manera que el alumno, mediante la realización de programas sencillos, comprenda todos los conceptos explicados en teoría.
- **Tutorías de atención al alumnado:** el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con su organización y planificación de la asignatura, así como sobre su contenido. Además de dichas tutorías individualizadas, se programarán varias tutorías en grupo, al menos una para cada bloque de la asignatura correspondiente. Asimismo, los alumnos pueden utilizar la modalidad de tutorías virtuales para problemas puntuales que puedan surgir durante el estudio en casa.

5.2 ESTRATEGIAS DE APRENDIZAJE

Las estrategias de aprendizaje se han establecido a nivel común en el artículo introductorio de este libro. Tal y como se indicó, los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador no son los únicos medios sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web y el campus virtual ofrecen innumerables posibilidades que no hay que dejar pasar. Así, se ha elaborado en el campus virtual, una página de la asignatura, que incluye toda la información que el alumno necesita. El uso de la misma ha sido mayoritario en las experiencias llevadas a cabo hasta el momento.

Entre otras cosas, en dicha página podemos encontrar:

- **Anuncios:** desde aquí, el alumno puede estar perfectamente informado de cualquier novedad relacionada con la asignatura. Además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, etc.
- **Tutorías:** Indica el horario de atención al alumnado.
- **Objetivos:** resume los objetivos que se pretenden alcanzar en esta asignatura.
- **Temario:** especifica el temario de esta asignatura.
- **Material docente de la asignatura:** aquí se puede encontrar todo el material que como mínimo va a ser necesario para el seguimiento de las clases teóricas y prácticas.
- **Bibliografía complementaria:** aparece un listado de bibliografía complementaria que pueden consultar para profundizar en la asignatura o preparar los trabajos complementarios.

- Horarios de teoría y prácticas: Contiene información sobre los grupos de teoría y prácticas, así como el profesorado que los imparte.
- Enlaces de interés: Aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en algunos contenidos de la materia.

En base a todo esto, la estrategia de aprendizaje que se propone, se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura.

Planificación de las clases teóricas:

- Lectura previa del guión correspondiente a la sesión de teoría que se trate.
- Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y, en caso de no entender algo, intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida en la propia aula entre todos o por el profesor fuera del aula.
- Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario se pedirá ayuda al profesor correspondiente.

Planificación de las clases prácticas:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- Parte de las prácticas se realizarán en los laboratorios y parte en horas no presenciales. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá con bastante celeridad dichas prácticas, indicando una vez corregidas los fallos más comunes. Cada estudiante de forma individual debe analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario se pedirá ayuda al profesor correspondiente.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES					
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos pequeños / tutorías docentes	
Presentación:		1	0	0	
BLOQUE 1	Tema 1	1	2	0	
	Tema 2	6	6	4	
	Tema 3	4	6	2	
	Tema 4	4	4	2	
	Tema 5	4	6	2	
	Tema 6	2	2	1	
BLOQUE 2	Tema 7	3	4	2	
Preparación del examen final:		2	0	3	
Examen final:		3	0	0	
TOTAL: 76		30	30	16	

Cuadro 3. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES					
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños / tutorías docentes	
Presentación:		0	0	0	
BLOQUE 1	Tema 1	1,75	4	0	
	Tema 2	10,5	12	1	
	Tema 3	9	12	0,5	
	Tema 4	9	8	0,5	
	Tema 5	9	12	0,5	
	Tema 6	3,5	4	0,25	
BLOQUE 2	Tema 7	5,25	8	0,5	
Preparación del examen final:		3,5	0	1,75	
Tutorías: 1					
TOTAL: 117,5		52,5	60	5	

Cuadro 4. Plan de trabajo de la asignatura, número de horas no presenciales

La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y entendimiento de los enunciados de los problemas, y revisión de los problemas ya corregidos por el profesorado.

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

- Hennesy, J.L. ; y Patterson, D.A. (2002) *Arquitectura de computadores: un enfoque cuantitativo*. McGraw-Hill.
- Patterson, D.A.; Hennesy, J.L. (2000) *Estructura y diseño de computadores. Interficie, circuitería/programación*. Reverté.
- Rico, A.; Soriano y Párraga, A.(1999) *Estructuras de computadores. Un computador ejemplo: MANOTAS*. A. Grediaga, M:L., Servicio de publicaciones de la UA.
- Stallings, W. (2000) *Organización y arquitectura de computadores..* Prentice Hall.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

- Angulo, J.M.; García, J. y Angulo, I. (2003) *Fundamentos y estructura de computadores..* Thomson.
- De Miguel Anasagasti, P. (2000) *Fundamentos de computadores* Paraninfo.
- Soriano, A.; Grediaga, A.; García, J. y Mora, F.J. (2002) *Estructuras de computadores: Problemas resueltos*. Servicio de Publicaciones de la UA.
- Tanenbaum, A.S. (2000) *Organización de computadoras: Un enfoque estructurado..* Prentice Hall.

7.3 OTROS RECURSOS

Como se ha mencionado ya, en esta asignatura, el alumnado dispone de un guión de la asignatura y de un libro de texto, del que se ayudan en las clases de teoría. Además, el alumnado dispone en la red de todas las actividades que deben realizar. Pero además de estos recursos básicos, incluimos aquí otros recursos de especial interés, tanto para el profesorado como el alumnado.

Publicaciones periódicas:

- Integration, the VLSI journal
- IEEE Transactions on Computers
- IEEE Transactions on VLSI Systems
- IEE Proceedings - Computers and Digital Techniques
- IEE Proceedings - Circuits, Devices and Systems
- Journal of Circuits, Systems and Computers
- Circuits, Systems, and Signal Processing

- TCS (Theoretical Computer Science)
- SIAM Journal on Computing
- Journal of the ACM
- Journal of Systems Architecture
- The Journal of Instruction-Level Parallelism
- IPL (Information Processing Letters)
- Parallel Computing
- IEEE Concurrency
- Journal of Parallel and Distributed Computing
- International Journal of High Speed Computing
- Networks
- Cluster Computing
- Journal of Supercomputing
- The International Journal of Supercomputer Applications and High Performance Computing
- International Journal of Parallel and Distributed Systems and Networks
- Parallel Algorithms and Applications
- Concurrency: Practice and Experience
- Distributed Computing

Asociaciones y grupos de interés:

- ANSI: American National Standards Institute. <http://www.ansi.org/>
- EAPLS: European Association for Programming Languages and Systems. <http://danae.uni-muenster.de/eapls/>
- EUROMICRO: <http://www.euromicro.org/>
- IASTED: International Association of Science and Technology for Development. <http://www.iasted.com/>
- ISTECS: Iberoamerican Science and Technology Education Consortium: <http://www.istec.org/>
- ACM: Association for Computing Machinery. <http://www.acm.org>
- IEEE: Institute of electrical and Electronics Engineers. <http://www.ieee.org>
- IEEE Computer Society. <http://www.computer.org>
- ACM SIGCSE: Special Interest Group on Computer & Science Education. <http://www.acm.org/sigcse>

Páginas web de interés:

Algunas páginas Web muy interesantes, que proporcionan enlaces o recopilaciones de enlaces a recursos estadísticos (organizaciones, revistas electrónicas, conferencias, bibliografía, tutoriales, etc.) son las siguientes:

- <http://atc.ugr.es/~acanas/arquitectura.html><http://www.analyse-it.com/download/dl.asp> (enlaces de interés)

- <http://www.cs.wisc.edu/~arch/www/> (Web de AC de la Universidad de Wisconsin)
- <http://csjava.occ.cccd.edu/~pharao/CS116Links.html> (Diccionarios, software, etc sobre AC)
- <http://www.ee.iastate.edu/~acar/archlink.html> (lista de grupos de investigación sobre AC)

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces la resolución de dudas de esta forma es fácil. No obstante, cuando la duda es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

8.1 PROCEDIMIENTOS DE EVALUACIÓN

Para evaluar al alumno, en esta asignatura, se considerarán lo siguiente:

NOTA DE ARQUITECTURA DE COMPUTADORES	
Examen de teoría	50%
Prácticas con entorno de simulación	20%
Examen de prácticas	10%
Asistencia y participación	10%
Realización de ejercicios propuestos	10%

Cuadro 5. *Procedimientos de evaluación*

La asistencia a las clases de prácticas y resolución de ejercicios propuestos en grupos pequeños será obligatoria, valorándose la participación de los alumnos. Será necesario superar por separado las componentes teóricas y prácticas de la asignatura.

8.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento de la asignatura es superior al exigido en el programa. • La comprensión conceptual es sobresaliente. • Los problemas propuestos se resuelven con gran solvencia. • La capacidad para programar en lenguaje ensamblador es muy alta. • La actitud en clase es de participación muy activa.

Calificación	Rango	Pautas
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento de la asignatura es correcto. • La comprensión conceptual es notable. • Los problemas propuestos se resuelven adecuadamente. • La capacidad para programar en lenguaje ensamblador es adecuada. • La actitud y participación en clase es buena.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento de los contenidos de la asignatura es básico. • La comprensión conceptual es aceptable. • Los problemas propuestos se resuelven adecuadamente en la mayor parte de los casos. • La capacidad para programar en lenguaje ensamblador es básica pero suficiente. • La actitud y participación en clase es aceptable.
Suspenso	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento de la asignatura no es aceptable. • No se han resuelto la mayor parte de los problemas propuestos. • La capacidad para programar en ensamblador no llega al mínimo deseable. • La actitud y participación en las clase es escasa o nula.

Cuadro 6. *Criterios de evaluación*

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Arquitecturas de Computadores. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo de propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI1	De CIC1 a CIC15 cCIC1 cCIC2	1-6	7	Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI2	De CIC1 a CIC11 cCIC1 cCIC2	1-6		Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios.. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI3	De CIC1 a CIC13 cCIC1 cCIC2	1-6		Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI3	CIC9 cCIC1	1-6		Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI5	De CIC10 a CIC11 cCIC1	1-6		Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI6	De CIC12 a CIC13 cCIC1	1-6		Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI7	De CIC1 a CIC15 cCIC1 cCIC2	1-6	7	Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
cOI6	De CIC14 a CIC15	1-6	7	Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 8. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)		
cOI1	De CIM1 a CIM8 CIT1 cCIT1	1-6	7	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (Aprendizaje on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Examen. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI2	CIT1 cCIT1	1-6	7	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (Realización de prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Examen. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI3	CIL1 cCIL1	1-6	7	Enseñanza presencial (Lección magistral/trabajo de aula en grupos). Enseñanza no presencial (Aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de rigurosidad en las explicaciones de los procesos descritos.
cOI4	cCIL2	1-6	7	Enseñanza presencial (Lección magistral). Enseñanza no presencial (Aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de conocimiento de los distintos términos relacionados, en castellano y/o valenciano y en inglés.

Cuadro 9. Análisis de coherencia: competencias interpersonales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)		
cOIS	De CIM1 a CIM8 cCIM3	1-6	7	Enseñanza no presencial (Aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
cOIP1	CIPTC1 cCIPTC1 cCIPTC2	1-6	7	Enseñanza presencial (Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3	1-6	7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4	1-6	7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 10. Análisis de coherencia: competencias interpersonales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)		
cOS1	cCS1 cCS4 cCS5	1-6	7	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3	1-6	7	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los conocimientos adquiridos
cOS3	cCS1 cCS2 cCS4 cCS5	1-6	7	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2	1-6	7	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos estadísticos aplicados.

Cuadro 11. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE BASES DE DATOS I

Armando Suárez Cueto; Patricio Martínez Barco; Andrés Montoyo Guijarro;
Cristina Cachero Castro; Eva Gómez Ballester; Paloma Moreda Pozo;
Rafael Muñoz Guillena; Manuel Palomar Sanz;
Juan Carlos Trujillo Mondéjar;
Sonia Vázquez Pérez

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
(armando;patricio;montoyo;ccachero;eva;moreda;rafael;mpalomar;
jtrujillo;svazquez}@dlsi.ua.es*

1. CONTEXTUALIZACIÓN

Toda organización, sea pequeña o grande, tiene unas necesidades de información, bien en la forma tradicional de datos administrativos, bien en sistemas avanzados de tratamiento de información de todo tipo. De todos los datos que entran y salen de esa organización, en el formato que sea, unos son importantes y otros no tanto.

El objetivo de un analista es identificar la información importante y estructurarla de forma que sea útil para todos los miembros de la organización. Ese sistema de información puede ser mecanizado mediante herramientas informáticas y servir así a la productividad de la entidad.

En un principio, los sistemas de información a mecanizar eran sencillos y reflejaban más o menos exactamente el flujo administrativo de papel del exterior hacia la empresa, dentro de la misma empresa, y de la empresa hacia el exterior, nuevamente. Para ello se utilizaban los lenguajes de programación disponibles, más o menos adecuados para la tarea, que manejaban ficheros organizados según lo permitía la tecnología del momento.

Pero pronto nuevas necesidades y expectativas hicieron que el mantenimiento y creación de aplicaciones informáticas, junto con el incremento masivo de la cantidad de datos a almacenar y tratar, se convirtiera en un cuello de botella debido a problemas de redundancia (e inconsistencia) de datos, deficientes medidas de seguridad, baja calidad de la información almacenada, y pérdidas de información por diversas causas. La tecnología del momento no era adecuada para sistemas de información en constante evolución y con unos requerimientos de rendimiento y fiabilidad cada vez más exigentes.

La aparición de las técnicas de bases de datos vino a solucionar gran parte de estos problemas.

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

Siguiendo la propuesta del Libro Blanco sobre las titulaciones universitarias de informática en el nuevo espacio europeo de educación superior [EICE04], hay una sola titulación de Grado denominada Ingeniería en Informática, que recoge los contenidos generales y básicos de la enseñanza superior informática. Los titulados en Ingeniería Informática deben ser profesionales con una formación amplia y sólida que les prepare para dirigir y realizar tareas en todas las fases del ciclo de vida de los sistemas de información mediante la aplicación de su conocimiento científico y de los métodos y técnicas propios de la ingeniería. Por ello, se establecen los siguientes objetivos generales:

- Estar preparados para ejercer la profesión, teniendo una conciencia clara de su dimensión humana, económica, social, legal y ética.
- Estar preparados para, a lo largo de su carrera profesional, asumir tareas de responsabilidad en las organizaciones, tanto de contenido técnico como directivo, y contribuir en la gestión de la información y en la gestión del conocimiento.
- Tener las capacidades requeridas en la práctica profesional de la ingeniería: ser capaces de dirigir proyectos, de comunicarse de forma clara y efectiva, de trabajar en equipos multidisciplinarios y conducirlos adecuadamente, de adaptarse a los cambios y de aprender autónomamente a lo largo de la vida.
- Estar preparados para aprender y utilizar de forma efectiva técnicas y herramientas que surjan en el futuro. Esta versatilidad les hace especialmente valiosos en organizaciones en las que sea necesaria una innovación permanente.
- Ser capaces de especificar, diseñar, construir, implantar, verificar, auditar, evaluar y mantener sistemas informáticos que respondan a las necesidades de sus usuarios.
- Tener la formación de base suficiente para poder continuar estudios, nacionales o internacionales, de Máster y Doctorado.

Con el fin de lograr estos objetivos, los Contenidos Formativos Comunes (CFC) que representa el 60% de la carga de los estudios (al menos 144 ECTS), teniendo en cuenta las recomendaciones ACM/IEEE, quedan distribuidos de la siguiente forma:

- Fundamentos científicos (10% - 15%).
- Fundamentos matemáticos de la Informática.
- Fundamentos físicos de la Informática.
- Contenidos específicos de la Ingeniería en Informática (5% - 10%).

- Programación.
- Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes.
- Sistemas operativos, Sistemas Distribuidos y Redes.
- Ingeniería de Computadores.
- Contenidos Generales de la Ingeniería (35% - 40%).
- Gestión de las organizaciones.
- Ética, legislación y profesión.
- Destrezas profesionales.
- Proyecto de fin de carrera (6%).

Así, la materia de las Bases de Datos incluye dentro de la categoría Contenidos específicos de la Ingeniería en Informática y más concretamente en la sub-categoría de Ingeniería del Software, cuyo detalle se muestra a continuación:

- Programación (P): fundamentos y metodología de la programación, algoritmia, computabilidad, lenguajes de programación, paradigmas de programación, estructuras de datos.
- Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes (IS): desarrollo de software, procesos, requisitos, especificación y diseño, gestión de proyectos, calidad del software, interacción persona-computadora, bases de datos, inteligencia artificial.
- Sistemas Operativos, Sistemas Distribuidos y Redes (SO): sistemas operativos, sistemas distribuidos, sistemas de tiempo real, arquitectura e infraestructura de redes y servicios telemáticos, seguridad.
- Ingeniería de los computadores (IC): fundamentos, estructura y arquitectura de computadores, tecnología de computadores.

De todo lo expuesto se deduce que, aunque los tópicos relacionados con esta asignatura van a ser básicos en la formación de cualquier ingeniero informático, sea cual sea su perfil académico y su futuro perfil profesional, toman especial relevancia en los siguientes perfiles, englobados a su vez en tres perfiles profesionales generales.

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: la mayor parte de los sistemas software, en mayor o menor medida, precisan de un proceso sofisticado de la información, que estructura en una base de datos que facilita su gestión. Para ello es imprescindible conocer y manejar sistemas de gestión de base de datos, y disponer de las herramientas adecuadas para su acceso usando cualquier lenguaje de programación.
- Arquitectura y diseño de software: los sistemas de información mecanizados son muy a menudo el objeto del desarrollo de software específico y aplicaciones para la gestión y presentación de los datos, siendo las bases de datos la tecnología adecuada sobre todo en grandes volúmenes de datos.

- **Diseño multimedia:** el almacenamiento y acceso eficiente a información heterogénea es fundamental en aplicaciones multimedia. Es cada vez más incuestionable la necesidad de bases de datos específicamente diseñadas para este propósito.

Perfil profesional de sistemas:

- **Ingeniería de comunicación de datos:** gran parte del tráfico de datos generado por las aplicaciones proviene de las órdenes enviadas a sistemas de gestión de bases de datos y de la recepción de datos, con problemas específicos derivados de los conceptos de integridad y seguridad de datos.
- **Diseño de redes de comunicación:** comercio electrónico, bibliotecas, portales corporativos e institucionales en Internet, etc. Todos precisan del mantenimiento y accesibilidad de sus bases de datos.
- **Asistencia técnica:** la complejidad de los sistemas hardware y software que soportan bases de datos necesitan apoyo técnico que permita elegir las mejores opciones dentro del amplio abanico de soluciones generales y específicas.
- **Ingeniería de integración y pruebas e implantación y pruebas:** los entornos de base de datos generan problemas concretos que afectan tanto al hardware como al software y determinan el éxito o el fracaso de la puesta en marcha del sistema.

Perfil profesional de gestión y explotación de tecnologías de la información:

- **Consultoría de empresas de TI:** las bases de datos son parte fundamental de las Tecnologías de la Información, siendo de especial importancia el análisis y diseño de las mismas, así como resolver satisfactoriamente aspectos relevantes como la seguridad y recuperación de datos, la elección de las arquitecturas y del software, especialmente sistemas de gestión de bases de datos, y ofrecer un servicio eficaz y eficiente.
- **Especialista en sistemas:** los sistemas basados en el manejo intensivo de datos son muchos y muy variados, estratégicamente fundamentales para el éxito de la puesta en marcha de un negocio o un servicio, y el conocimiento de todas las posibles opciones es imprescindible. Particularmente, la complejidad de los sistemas de gestión de bases de datos hace de estos profesionales un personal altamente demandado.
- **Desarrollo de investigación y tecnología:** las líneas de investigación en tecnologías de bases de datos abarcan múltiples aspectos del almacenamiento y gestión de datos.
- **Dirección de TIC (marketing, proyectos, dirección general):** este perfil requiere un conocimiento amplio del resto de áreas tecnológicas de las Tecnologías de la Información y las Comunicaciones (TIC) y por tanto también de la importancia e influencia de los sistemas de bases de datos.

Perfil Titulación	Perfil Asignatura
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de las tecnologías de bases de datos en la gestión de la información.
Arquitectura y diseño de software	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de las tecnologías de bases de datos en el análisis y diseño de sistemas de información mecanizados.
Diseño multimedia	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de las tecnologías de bases de datos multimedia.
Ingeniería de comunicación de datos.	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio y aplicación de los distintos aspectos necesarios para una gestión de bases de datos eficiente y segura.
Diseño de redes de comunicación	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio y aplicación de los distintos sistemas de acceso y manipulación de bases de datos.
Asistencia técnica	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de los sistemas de gestión de bases de datos y del diseño correcto de los sistemas de información.
Ingeniería de integración y pruebas e implantación y pruebas	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio y aplicación de las tecnologías de bases de datos.
Consultoría en empresas de TI	<ul style="list-style-type: none"> • Conocimiento y habilidad en análisis y diseño de sistemas de información mecanizados y sistemas de gestión de bases de datos.
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimiento y habilidad en la puesta en marcha y optimización de sistemas de gestión de bases de datos.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento y habilidad de las tecnologías de bases de datos actuales y de las líneas de investigación que intentan dar solución a las nuevas necesidades en
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la estadística al control de sistemas informáticos. • Conocimiento y habilidad en técnicas de planificación. • Conocimiento y habilidad en la aplicación de la estadística en el control de calidad y fiabilidad. • Conocimiento y habilidad en la aplicación de las bases de datos al control de sistemas informáticos, técnicas de planificación y control de calidad y fiabilidad.

Cuadro 1. Adecuación de los tópicos relacionados con la asignatura Bases de Datos I

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Bases de Datos I (BD1) recoge gran parte de los Contenidos Formativos Comunes asignados a la materia de bases de datos en el Plan de Estudios de Ingeniero en Informática. En este contexto, la asignatura de BD1, con descriptor ESTRUCTURA DE INFORMACIÓN, FICHEROS, BASES DE DATOS, supone el primer contacto que el alumno tiene con esta materia y, en general, con la subcategoría IS. En concreto, se centra en la presentación de los fundamentos básicos de las bases de datos.

Dentro de los Contenidos Formativos Comunes, esta asignatura tiene su continuación en Bases de Datos II (BD2) completando así los contenidos asignados a la materia. BD2, la cual se centra ya en aspectos de diseño y gestión de las bases de datos y enlaza con los contenidos de las asignaturas de Análisis y Especificación de Sistemas de Información, Ingeniería del Software I, y Diseño y Programación Avanzada de Aplicaciones. Entre las materias ofertadas discrecionalmente por cada Universidad, la materia de Bases de Datos se completa con las asignaturas Bases de Datos Multidimensionales, Bases de Datos Avanzadas, Bases de Datos Distribuidas, y Administración de Bases de Datos.

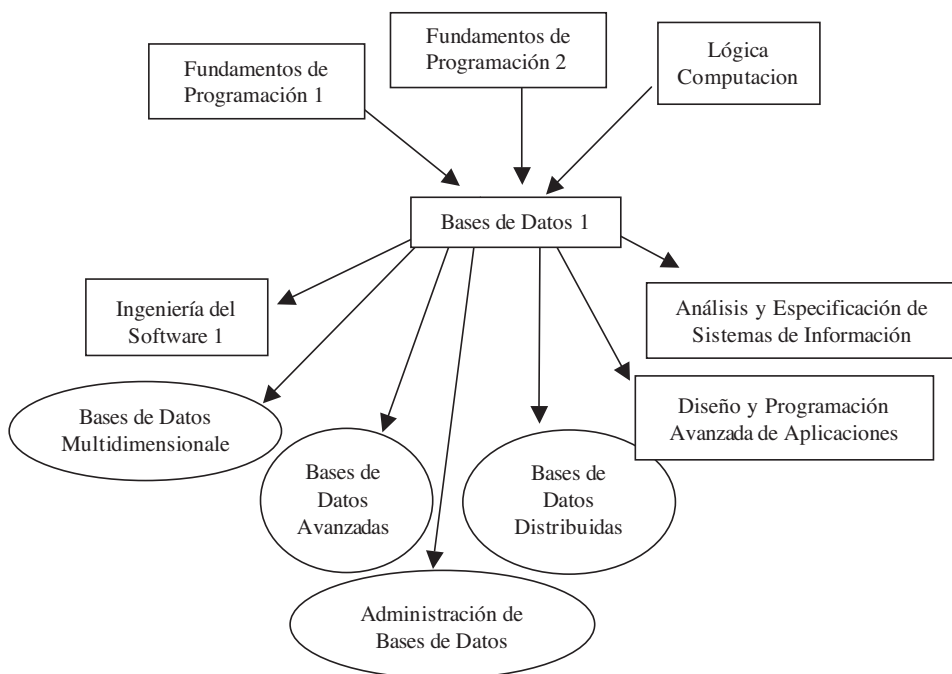


Ilustración 4. Relaciones de Bases de Datos I en el plan de estudios.

1.3 BASES DE DATOS I EN LOS CURRÍCULOS INTERNACIONALES

La asignatura Bases de Datos I, como asignatura básica en la enseñanza de la materia de bases de datos, ha sido recomendada en los currículos internacionales de mayor prestigio, en concreto, los propuestos en ACM/IEEE (1991, 2001), ACM/AIS/AITP en IS (1997) y UNESCO-IFIP (1994, 2000), que analizaremos a continuación.

1.3.1 Bases de Datos I en el modelo curricular ACM/IEEE (1991, 2001)

En concreto, en la propuesta de ACM/IEEE'91, hay una selección de cursos correspondientes a la materia o requisito común DB: Bases de datos y recuperación de la información. Esta se compone de dos unidades de conocimiento básicas “**DB1: Visión global, modelos y aplicaciones de los sistemas de bases de datos**” y “**DB2: El modelo de datos relacional**”, junto con el curso avanzado de “**Bases de datos y recuperación de información**”.

Estas 3 unidades están totalmente relacionadas con la mayoría de la propuesta para Bases de Datos I. En concreto, **DB1** está relacionada con los contenidos del temario de Bases de Datos I correspondientes a la “Introducción a las bases de datos” y a “Modelos de datos”. La segunda unidad, **DB2**, está relacionada con los contenidos sobre el “Modelo Relacional” y el “Diseño de bases de datos relacionales”. Por último, el curso avanzado contiene la relación con los contenidos que tratan “La perspectiva lógica del modelo relacional”, la “Organización física de las bases de datos”, los “Sistemas de gestión de bases de datos” y el “Modelo entidad-relación”.

Sin embargo, en general, el contenido de bases de datos en esta propuesta resultaba evidentemente escaso. Apenas nueve horas lectivas quedaban propuestas para desarrollar un temario bastante ambicioso lo que evidentemente **generaría** una enseñanza insuficiente en materia de bases de datos.

Esta deficiencia fue revisada en la propuesta de ACM/IEEE'2001 donde se destaca un crecimiento en el contenido sobre bases de datos a través del área “**IM: Gestión de la información**”. De ella se destacan en concreto las unidades de conocimiento que se comentan a continuación: “**IM1: Modelos y sistemas de información**” y “**IM2: Sistemas de bases de datos**”, “**IM3: Modelado de datos**”, “**IM4: Bases de datos relacionales**” “**IM5: Lenguajes de interrogación a bases de datos**”, “**IM6: Diseño de Bases de Datos relaciones**” y “**IM9: Diseño físico de la base de datos**”, todos ellos fuertemente ligados a la propuesta realizada en esta guía para BD1.

Además, cabe resaltar que el resto de unidades presentadas en el área **IM** forman parte, en menor o mayor medida, del resto de asignaturas que conforman la materia de Bases de Datos en la Universidad de Alicante (Bases de Datos II -obligatoria de 3er. curso-, Administración de Bases de Datos -optativa-, Bases de Datos Avanzadas -optativa-, Bases de Datos Distribuidas -optativa-, y Bases de Datos Multidimensionales).

1.3.2 Bases de Datos I en el modelo curricular ACM/AIS/AITP en IS (1997)

Respecto a la propuesta del modelo curricular planteado en el informe IS'97, cabe destacar que la primera parte del apartado **1.6 Bases de Datos**, de su cuerpo de conocimientos corresponde básicamente al contenido de la asignatura BD1, tal y como se propone en esta guía. Así los apartados **1.6.1 (SGBD: características, funciones, arquitectura)**, **1.6.2 (Modelos de datos: relacional, jerárquico, red, objetual, semántico-objetual)**, **1.6.4 (Integridad)**, **1.6.5 (Lenguajes de definición de datos)**, y **1.6.3 (Normalización)** están perfectamente recogidos en el temario de la asignatura propuesta.

1.3.3 Bases de Datos I en el modelo curricular de UNESCO-IFIP (1994, 2000)

En la propuesta de currículo de UNESCO-IFIP, el módulo **5.6. Diseño y gestión de bases de datos** está perfectamente relacionado con los contenidos del programa propuesto para BD1 en cuanto a los fundamentos tecnológicos de las bases de datos, aunque este último profundiza más el modelo de datos relacional.

2. Objetivos

2.1 Objetivos generales

Los objetivos generales de la asignatura de Bases de Datos I son:

2.1.1 Instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos:

- OI1: Comprender los fundamentos tecnológicos de las BD y su influencia en los sistemas de información.
- OI2: Comprender el concepto de modelo de datos.
- OI3: Aplicar los conceptos del modelo relacional a la creación de bases de datos y la manipulación de datos.
- OI4: Dominar el Modelo Relacional.
- OI5: Comprender el Modelo Entidad-Relación Extendido.
- OI6: Experimentar con los sistemas de gestión de bases de datos relacionales.

2.1.2 Interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3 detallados en el artículo introductorio de este libro.

2.1.3 Sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

A continuación se enumeran las competencias que, a partir de los objetivos desarrollados en el punto anterior, consideramos que se deben adquirir en la asignatura Bases de Datos I [MS04].

2.2.1 Competencias Instrumentales Específicas

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber)

Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2, desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque I: Introducción

- CIC1: Comprender el concepto de sistema de información y sus componentes.
- CIC2: Conocer las fases del desarrollo de un sistema de información mecanizado.
- CIC3: Comprender que son las propiedades estáticas y dinámicas de un sistema de información.
- CIC4: Conocer la evolución histórica de la mecanización de los sistemas de información con ordenador.
- CIC5: Comprender el concepto de base de datos y sus características principales.
- CIC6: Comprender los conceptos de independencia, integridad y seguridad de datos.
- CIC7: Comprender y diferenciar correctamente los conceptos: modelo de datos y esquema de bases de datos.
- CIC8: Estudiar los componentes fundamentales de los modelos de datos.
- CIC9: Estudiar la evolución de los modelos de datos y sus características principales.
- CIC10: Conocer varios de los modelos de datos existentes.

Bloque II: El modelo relacional

- CIC11: Conocer el desarrollo del Modelo Relacional y su importancia en los sistemas de gestión de Bases de Datos comerciales.

- CIC12: Conocer los fundamentos matemáticos del Modelo Relacional: concepto de relación.
- CIC13: Conocer estructuras básicas del Modelo Relacional, la tupla, la relación , y sus operadores.
- CIC14: Comprender el concepto de valor nulo en el modelo relacional.
- CIC15: Estudiar el concepto de clave candidata.
- CIC16: Estudiar el concepto de integridad referencial y su aplicación en los SGBD relacionales.
- CIC17: Conocer las posibilidades de representación del MR: restricciones de correspondencia entre clases.
- CIC18: Estudiar los lenguajes de manipulación teóricos propuestos para este modelo de datos (Álgebra relacional, Cálculo relacional de tuplas y Cálculo relacional de dominios) (cognitiva).
- CIC19: Comprender la diferencia entre lenguajes de manipulación relacionalmente completos y computacionalmente completos.
- CIC20: Comprender la teoría de la normalización, su relación con el diseño de bases de datos relacionales y la necesidad de normalizar las relaciones
- CIC21: Estudiar las formas normales básicas, primera, segunda y tercera formas normales.
- CIC22: Comprender los problemas que solucionan formas normales más avanzadas

Bloque III: Los sistemas de gestión de las bases de datos

- CIC23: Familiarizarse con el concepto del sistema de gestión de base de datos, sus componentes y servicios que proporciona, incluyendo el mantenimiento de la independencia, integridad y seguridad de los datos en Bases de Datos y los mecanismos asociados.
- CIC24: Conocer cuáles son los objetivos básicos de los sistemas de gestión de bases de datos.
- CIC25: Conocer la arquitectura de un sistema de gestión de bases de datos.
- CIC26: Comprender los niveles de independencia de datos que induce la arquitectura de un SGBD.
- CIC27: Diferenciar entre los distintos esquemas de un SGBD, la función de cada uno y los modelos de datos aplicables.
- CIC28: Estudiar los métodos de acceso del SGBD a la organización física de la base de datos.
- CIC29: Conocer la arquitectura cliente/servidor como mecanismo de acceso al sistema de gestión de base de datos.
- CIC30: Conocer varios SGBD.

Bloque IV: Fundamentos del modelo E/R

- CIC31: Conocer estructuras básicas del Modelo Entidad-Relación Extendido: la entidad, los atributos y las relaciones.
- CIC32: Conocer las distintas versiones existentes del modelo.

Bloque V: Otras tecnologías de BD

- CIC33: Conocer nuevas tecnologías en bases de datos y sistemas de gestión de bases de datos.
- CIC34: Comprender el impacto de internet en la gestión de datos y conocer las nuevas necesidades y posibilidades.
- CIC35: Conocer la minería de datos y los almacenes de datos.

Capacidades metodológicas (saber hacer)

Además de tener en cuenta las habilidades cognitivas cCIM1, cCIM2, y cCIM3 desarrolladas en artículo introductorio de este libro. Tendremos en cuenta las siguientes.

Bloque I: Introducción

- CIM1: Evaluar la necesidad de mecanizar un sistema de información, con bases de datos un sistema de información.
- CIM2: Utilizar un modelo de datos correcto según la fase de diseño y el software elegido.
- CIM3: Abstracta los datos esenciales y relevantes para un sistema de información sencillo.

Bloque II: El modelo relacional

- CIM4: Interpretar un esquema de base de datos relacional.
- CIM5: Definir relaciones a partir de un esquema de diseño para bases de datos relacionales.
- CIM6: Definir correctamente las claves candidatas.
- CIM7: Definir correctamente las restricciones de correspondencia entre clases (claves ajenas y su combinación con las demás restricciones: claves candidatas y valores no nulos).
- CIM8: Modificar adecuadamente un esquema de BDR, atendiendo a nuevos requisitos del sistema de información.
- CIM9: Comprobar la correcta forma normal de un conjunto de relaciones y efectuar los cambios que sean necesarios atendiendo a los requisitos previos del sistema de información que representan.
- CIM10: Definir las estrategias para mantener la integridad referencial, necesarias para cada clave ajena.
- CIM11: Consultar una BDR con álgebra relacional, cálculo relacional de tuplas y cálculo relacional de dominios.

- CIM12: expresar restricciones de integridad de una BDR con AR, CRT y CRD

Bloque III: Los sistemas de gestión de las bases de datos

- CIM13: Manejar una selección de Sistemas de Gestión de Bases de Datos (componentes y servicios que proporcionan).

Bloque IV: Fundamentos del modelo E/R

- CIM15: Interpretar esquemas conceptuales de base de datos empleando el Modelo Entidad-Relación Extendido.

Bloque V: Otras tecnologías de BD

Destrezas tecnológicas (saber hacer)

Además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se consideran las siguientes:

Bloque I: Introducción

Bloque II: El modelo relacional

- CIT1: definir correctamente un esquema de BDR con SQL: creación, modificación y eliminación de relaciones (tablas).
- CIT2: insertar, modificar y borrar datos con SQL.
- CIT3: consultar una BDR con SQL.

Bloque III: Los sistemas de gestión de las bases de datos

- CIT4: Crear interfaces para BDR sencillas, publicación en Internet.

Bloque IV: Fundamentos del modelo E/R.

Bloque V: Otras tecnologías de BD

Destrezas lingüísticas (saber hacer)

Las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro.

2.2.2 Competencias Interpersonales (ser y estar)

Las competencias interpersonales hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCIPTC1, cCIPTC2, cCIPTR1, cCIPTR2, cCIPTR3, y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Entender los elementos básicos de los lenguajes de programación de tercera generación: variables, funciones, tipos de datos estructurados, punteros, estructuras de datos dinámicas y ficheros.
- Entender los conceptos de registro y campo, organización y métodos de acceso de los ficheros.
- Entender los conceptos de la lógica de primer orden: lenguaje de primer orden, interpretación, fórmula bien formada y su evaluación.
- Entender el álgebra de conjuntos: concepto de conjunto, operadores unión, intersección, diferencia y producto cartesiano.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerequisites necesarios para el estudio y entendimiento de esta asignatura se cubren en las asignaturas Fundamentos de Programación 1 y 2, y Lógica Computacional, todas ellas asignaturas de primero, de carácter troncal y obligatorio según las titulaciones, excepto el álgebra de conjuntos cuyo contenido, al menos en los aspectos básicos que se necesitan en Bases de Datos 1, se desarrolla en la etapa preuniversitaria. No obstante, aquellos temas de contenido que precisen de estos conocimientos siempre incluirán un breve resumen dada la especificidad de los conceptos necesarios para bases de datos en general y el modelo relacional en particular.

4. BLOQUES Y TEMAS DE CONTENIDO

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

En este punto y, siguiendo los objetivos y competencias concretos de las bases de datos, enunciamos el contenido del programa de teoría y práctica de la asignatura de Bases de Datos I.

4.1.1 Programa de Teoría

Bloque I: Introducción

- Unidad 0: Presentación y objetivos.
- Unidad 1: Introducción a las bases de datos.
 - Sistemas de información.

- Bases de datos.
- Sistemas de gestión de bases de datos.
- Unidad 2: *Modelos* de datos.
 - Introducción a los modelos de datos.
 - Conceptos básicos de los modelos de datos.
 - Cualidades principales de un modelo de datos.
 - Desarrollo generacional y clasificación de los modelos de datos.

Bloque II: El modelo relacional

- Unidad 3: El modelo relacional.
 - Introducción al modelo relacional.
 - Estructuras de datos.
 - Esquema relacional.
 - Restricciones de integridad en el modelo relacional.
- Unidad 4: El álgebra relacional.
 - Operadores del álgebra relacional.
 - Interrogación de una BD en AR.
- Unidad 5: Introducción al diseño de bases de datos relacionales.
 - Introducción al diseño de bases de datos.
 - Teoría de la normalización.
 - Formas normales.
 - Procedimiento de normalización adicional.
- Unidad 6: La perspectiva lógica del modelo relacional.
 - Formalización lógica de una base de datos relacional.
 - Cálculo relacional de tuplas.
 - Cálculo relacional de dominios.

Bloque III: Los sistemas de gestión de las bases de datos

- Unidad 7: Organización física de las bases de datos.
 - Introducción.
 - Conceptos previos.
 - El tipo de datos fichero.
 - Implementación de bases de datos relacionales.
- Unidad 8: Sistemas de gestión de bases de datos.
 - Sistemas de gestión de bases de datos.
 - Arquitectura de los sistemas de gestión de bases de datos.
 - Independencia, integridad y seguridad de datos.
 - Esquema interno de la base de datos.
 - Arquitectura cliente/servidor.

Bloque IV: Fundamentos del modelo E/R

- Unidad 9: Introducción al modelo entidad-relación.
 - Introducción a los modelos semánticos.
 - El modelo entidad-relación y sus extensiones.
 - Mecanismos de abstracción en el modelo entidad-relación extendido (EER).
 - Comparación EER y modelo relacional.

Bloque V: Otras tecnologías de BD

- Unidad 10: Bases de Datos Avanzadas.
 - Nuevas tendencias en Bases de Datos.
 - Bases de Datos Distribuidas.
 - Administración de Bases de Datos.
 - Almacenes de Datos.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE**5.1 METODOLOGÍA DOCENTE**

Dentro de la disciplina de informática existen diversos campos de aplicación entre los cuales las bases de datos es uno de los más extendidos. Esto se debe a la progresiva informatización y automatización de muchas tareas anteriormente manuales o que requerían un esfuerzo humano en cuanto a recogida y análisis de información. Es necesario que el alumno comprenda la importancia de manejar adecuadamente la información, de proteger datos privados, de obtener información implícita, etc. Mediante una correcta comprensión del campo de las bases de datos y con unas adecuadas directrices de trabajo, el alumno será capaz de abordar situaciones que requieran nuevas habilidades.

Con el fin de acercar todos estos conceptos a los alumnos, proponemos una metodología docente basada en cinco puntos:

- Ideas previas: dada la popularidad de esta materia es frecuente que los alumnos tengan ideas preconcebidas. Será necesario indagar sobre las mismas.
- Transmisión de conocimientos: partiendo de la información recogida sobre esas ideas preconcebidas, se determinará la forma más adecuada de exponer los contenidos de cada tema en las clases de teoría y de prácticas. En todo momento, se mostrará su aplicación en el mundo laboral.
- Afianzamiento de conocimientos: los alumnos analizarán y estudiarán, tanto de forma individual como mediante trabajos en equipo, los conocimientos que le han sido transmitidos.
- Proceso de tutorías: para un mayor y mejor contacto con el alumno se proponen procesos de tutorías en grupos reducidos que permitan descubrir si los conocimientos han sido asimilados de forma adecuada.

- Actividades complementarias: conferencias, seminarios, búsquedas bibliográficas, discusión electrónica, prospección, análisis de productos comerciales, etc.

Ideas previas

Para un aprendizaje adecuado de cualquier materia es necesario comenzar por indagar en las teorías previas y preconcepciones que los alumnos tienen sobre ella. Las expectativas de los estudiantes deben ser debatidas y, en su caso, incorporadas total o parcialmente al programa de la asignatura. Conseguir que perciban que su voz cuenta y ha sido escuchada, y alcanzar un consenso y un compromiso con ellos en el planteamiento del curso, es sustancial para el establecimiento y desarrollo de una comunidad de discurso.

Por ello, la manera de afrontar cualquier tema de la asignatura comenzará con la puesta en común de los conocimientos previos de los alumnos sobre los aspectos fundamentales tratados en dicho tema. De esta manera será posible cubrir algunos objetivos básicos. Por un lado, el profesor estará en situación de identificar concepciones erróneas y no erróneas, lo que le permitirá reorganizar e incluso modificar la exposición de los contenidos de cada tema de acuerdo a tales teorías previas. Por otro lado, los alumnos obtendrán un doble beneficio ya que podrán compartir sus experiencias unos con otros, situación que ofrece muchas oportunidades y genera altas disposiciones para aprender. Además percibirán que asistiendo a clase participan en un debate constructivo, que les atañe personalmente, y que tiene que ver con sus conflictos y sus dudas.

Sin embargo, junto con este planteamiento surge el problema de la comunicación profesor-alumno y alumno-alumno. De todos es conocido el temor que existe en una gran parte del alumnado a la hora de dirigirse al profesor, o incluso al resto de alumnos, cuando los grupos están formados por un elevado número de estudiantes.

Frente a los métodos tradicionales de interacción, la incorporación de las Nuevas Tecnologías de la Información y la Comunicación en el ámbito educativo ayudan a superar tales barreras de comunicación. Hoy en día, Internet, con sus poderosos instrumentos -la World Wide Web, el correo electrónico y las listas de debate electrónico- es una red popular mundial para la comunicación y el intercambio de información. El uso de este medio facilitará que el alumno pueda expresar libremente sus opiniones.

Entre las diferentes opciones que Internet nos ofrece, la tarea será afrontada mediante la participación en debates electrónicos. Más concretamente, “el debate” del Campus Virtual de la Universidad de Alicante, ofrece la posibilidad de realizar dichos intercambios de información de forma fácil y cómoda.

Transmisión de conocimientos

El campo de las bases de datos ha tenido un desarrollo paulatino desde sus inicios, de forma que, actualmente, muchas de las ideas iniciales han evolucionado

nado mejorando las prestaciones. Para una correcta comprensión por parte del alumnado, es importante reflejar todos los aspectos de esta evolución desde el marco científico. De esta forma, tendrán una visión global de la importancia de las bases de datos a través de su evolución en el tiempo.

Uno de los principales objetivos perseguidos es que el alumnado conozca que dentro del campo de las bases de datos existe una terminología específica que debe manejar con destreza, junto con una serie de conceptos básicos, métodos, modelos, sistemas, lenguajes, etc. Todos estos conocimientos serán impartidos de forma interrelacionada, mostrando al alumnado la necesidad de comprender y saber manejar todas las posibles variaciones y aplicaciones. Este planteamiento viene determinado por la certeza de que en un futuro el alumnado tendrá que enfrentarse a nuevas propuestas, nuevas aplicaciones, nuevos lenguajes, etc, y su familiarización con las nuevas aplicaciones vendrá determinada por las capacidades adquiridas a lo largo de su trayectoria académica.

Es por tanto muy importante, la incorporación de consideraciones metodológicas que faciliten el desarrollo de un aprendizaje experimental y fomente las actitudes críticas y creativas en el alumno. Los nuevos paradigmas docentes propugnan los modelos educativos que propician el pensamiento creativo, enseñando a aprender por encima de enseñar conocimientos. Nos proponemos diseñar un modelo en el que la clase teórica tiene un papel importante, pero no exclusivo en la transmisión de conocimientos. Este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas de laboratorio y las actividades en grupos pequeños, que jugarán un papel fundamental.

Cada aspecto teórico tratado en clase tendrá una relación directa con el mundo laboral y por tanto, se mostrará al alumnado ejemplos reales donde los conceptos adquiridos pueden aplicarse. De esta forma, el estudiante tendrá una motivación para adquirir esos conocimientos y comprender su utilidad. Para ello, se realizarán clases expositivas-interactivas donde se motive a los alumnos a intervenir en cualquier momento, exponiendo sus dudas, realizando comentarios que se intentarán aprovechar para realizar de forma más dinámica la clase, y facilitar el aprendizaje deseado por parte de los alumnos. Por último, es interesante dedicar siempre un período de la clase para mostrar las conclusiones más importantes a modo de resumen del tema, haciendo uso de la información recopilada mediante el debate electrónico.

Para incentivar la motivación del alumnado y aumentar el interés por la asignatura, en las clases prácticas, se propondrán diversas actividades, trabajos de investigación, etc, para que el alumno busque distintas aplicaciones de bases de datos utilizadas en empresas reales, proponga, a partir de los conocimientos adquiridos, mejoras en algún sistema real proporcionado por el profesor, cree un sistema a partir de una propuesta determinada, etc.

Un último punto a tratar dentro de la metodología, y no menos importante, es que el alumnado comprenda la necesidad de integración en grupos de trabajo, ya que, en un futuro podría desarrollar un proyecto conjunto, y para ello, es necesario tener una buena coordinación, comunicación y planificación con todos los miembros del proyecto. Unido a esta consideración, también es necesario que el alumnado comprenda las exigencias del usuario y sepa cómo responder a sus requerimientos mostrando diferentes alternativas y en cualquier caso, asesorar al usuario mostrándole la alternativa más ventajosa para sus necesidades.

Afianzamiento de conocimientos

Los alumnos analizarán y estudiarán, tanto de forma individual como mediante trabajos colaborativos con otros alumnos, los conocimientos que le han sido transmitidos. Para el desarrollo de estos trabajos se utilizará de nuevo la herramienta para debates electrónicos proporcionada por el Campus Virtual de la Universidad. Se plantearán cuestiones y/o ejercicios que inviten al debate entre las posibles soluciones y a la colaboración entre alumnos.

Proceso de tutorías

Además de las clases de teoría y prácticas en las que se abordarán desde los conceptos científicos hasta la resolución de ejercicios, es conveniente seguir un proceso de tutorías. Este proceso abarcaría dos aspectos distintos. Por un lado, las consultas individuales que el alumno pueda plantear al profesor durante el desarrollo de un tema y, por otro, reuniones en grupos reducidos al acabar cada tema con el objetivo de intercambiar opiniones, afianzar los conceptos vistos en clase y asegurarnos de que los alumnos no se queden con dudas sobre los contenidos del tema. Además, se pueden reforzar los conceptos con la resolución de ejercicios.

Las consultas individuales son prácticas para el alumno; sin embargo, al tratarse en su mayoría de consultas concretas, no permiten apreciar totalmente al profesor si los conceptos han quedado lo suficientemente claros. Las tutorías en grupos reducidos permiten recoger más opiniones y tener una visión más detallada sobre la asimilación de los distintos conceptos por parte de los alumnos. Estas tutorías en grupo se pueden plantear en un número de dos tutorías por tema y grupo de trabajo. Los grupos podrían ser de unas 10 personas, un número no muy grande para que, por un lado, el alumno no se pierda en el anonimato que muchas veces supone un grupo grande, y por otro lado, que sean los suficientes estudiantes para que surjan diferentes opiniones y provoque un pequeño debate entre ellos, dejando la figura del profesor como guía y moderador.

En la primera de las tutorías el profesor debería plantear diversas cuestiones dirigidas a indagar sobre la comprensión, por parte de los alumnos, de los conceptos teóricos vistos en clase. Además, se plantearían cuestiones dirigidas a

conocer si, tras las clases, los conceptos previos que ellos creían tener sobre el tema han quedado bien reforzados, si eran correctos, o rechazados, en el caso de estar equivocados. Por otro lado, habría que intentar comprobar el nivel de dificultad del tema, y el tiempo que se necesita para la asimilación de todos estos conceptos. Por tanto, se trata de un proceso, que además de ser un apoyo al alumno, nos debe servir para descubrir fallos en la estructura de contenidos, que podamos corregir en un futuro.

En la segunda de las reuniones de tutorías, se trataría de reforzar todos los conceptos vistos a través de su aplicación a ejercicios prácticos. A ser posible, se deberían plantear ejercicios que admitan varios análisis y vías de solución para fomentar el intercambio de ideas entre los alumnos, lo que siempre enriquecerá sus conocimientos sobre el tema.

En definitiva, en estos grupos de trabajo se trataría de proporcionar al alumno un ambiente propicio, un ambiente seguro, en el se sienta libre de expresar sus opiniones sin miedo a equivocarse, para que en este ambiente de confianza se sienta parte del proceso de aprendizaje y responsable de su propio crecimiento.

Actividades complementarias

Dada la importancia de esta materia en el mundo laboral, e intentando aproximar al alumno al mismo, y siempre que el tema lo permita, se realizarán coloquios/conferencias con expertos en el sector.

5.2 ESTRATEGIAS DE APRENDIZAJE

Las estrategias de aprendizaje se han establecido a nivel común en el artículo introductorio de este libro. Tal y como se indicó, los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador, no son los únicos medios sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web y el campus virtual ofrecen innumerables posibilidades que no hay que dejar pasar.

Teniendo en cuenta el marco metodológico anterior, se plantea una estrategia didáctica basada en los siguientes mecanismos:

Actividades presenciales

- Trabajo en aulas. Clases teóricas y prácticas.
- Tutorías organizadas. Proceso de tutorías para resolución de dudas tanto de teoría como de prácticas.
- Actividades complementarias. Conferencias, coloquios, etc.

Actividades no presenciales

- Aprendizaje autónomo. Estudio individual.

Aprendizaje colaborativo.

- Debates electrónicos tanto para la recopilación de ideas previas y el afianzamiento de conceptos.

En base a todo esto, la estrategia de aprendizaje que se propone se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura.

Planificación de las clases teóricas:

- Lectura previa del guión correspondiente a la sesión de teoría que se trate.
- Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en la comprensión del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida en el aula, entre todos, o por el profesor, fuera del aula.
- Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Planificación de las clases prácticas:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- Parte de las prácticas se realizarán en los laboratorios y parte en horas no presenciales. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá con bastante celeridad dichas prácticas, indicando los fallos más comunes. Cada estudiante de forma individual debe analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Autoevaluación: una vez realizadas todas las actividades previas relacionadas con un tema concreto, el estudiante debe discernir si cree que dicho tema ha sido

totalmente entendido. En caso de no ser así, debe incidir en el estudio de los contenidos que crea tener más flojos, utilizando, si lo cree conveniente, las tutorías y realizando algunos problemas de ampliación, bien de los propuestos en las hojas de problemas o bien haciendo uso de la bibliografía. Cuando crea estar preparado puede realizar el ejercicio de autoevaluación del tema correspondiente, publicado en el campus virtual.

Evaluación final: si el resultado de los ejercicios de autoevaluación y de las prácticas propuestas ha sido satisfactorio, el estudiante habrá superado la asignatura. En caso contrario tendrá que profundizar más en los contenidos de la asignatura y realizar el examen final.

De forma opcional se podrán hacer trabajos complementarios individuales para subir la nota, siempre y cuando el trabajo realizado a lo largo del curso se considere satisfactorio.

6. Plan de trabajo de los alumnos

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES				
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos pequeños / tutorías docentes
Bloque 1	Unidad 0	2		
	Unidad 1	2		
	Unidad 2	4		
Bloque 2	Unidad 3	16	12	3
	Unidad 4	6		2
	Unidad 5	8		2
	Unidad 6	6	2	2
Bloque 3	Unidad 7	2		
	Unidad 8	2	1	
Bloque 4	Unidad 9	8		6
Bloque 5	Unidad 10	4		
Examen parcial 1		1		
Preparación del examen final:		1		
Examen final: 2		1		
TOTAL		64	17	15

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES				
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de asignatura	Desarrollo de las actividades en grupos pequeños/ tutorías docentes
Bloque 1	Unidad 0	3,5		
	Unidad 1	3,5		
	Unidad 2	7		
Bloque 2	Unidad 3	28	24	0,75
	Unidad 4	10,5		0,5
	Unidad 5	14		0,5
	Unidad 6	10,5	4	0,5
Bloque 3	Unidad 7	3,5		
	Unidad 8	3,5	2	
Bloque 4	Unidad 9	14		1,5
Bloque 5	Unidad 10	7		
Examen parcial				
Preparación del examen final:				
Examen final:				
TOTAL		105	30	3,75

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

Las horas no presenciales de la sesión de presentación estarán dedicadas a la recopilación de la documentación de la asignatura y, en su caso, al repaso de aquellos prerrequisitos que no se hayan alcanzado. La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y entendimiento de los enunciados de los problemas, y revisión de los problemas ya corregidos por el profesorado. De las horas no presenciales dedicadas a las clases de teoría se utiliza una por cada tema para la realización del ejercicio de autoevaluación.

7. BIBLIOGRAFÍA Y MATERIALES RECOMENDADOS

7.1 BIBLIOGRAFÍA RECOMENDADA

Celma Giménez, M.; Casamayor Ródenas, J.C.; Mota Herranz, L. (1998). *Bases de dades relacionals*. Valencia: Edición: Universitat Politècnica, Servei de Publicacions.

Date, C.J., *Introducción a los sistemas de bases de datos*. Edición: Addison-Wesley Publishing Company

Elmasri & Navathe, *Sistemas de bases de datos. Conceptos Fundamentales*. Edición: Addison Wesley Publishing Company

Mota, Celma y Casamayor *Bases de datos relacionales*. Autor(es):. Edición: Servicio de Publicaciones de la Universidad Politécnica de Valencia Real Academia Española de la Lengua (2002) *Diccionario de la Lengua Española*.

Silberschatz & Korth, *Fundamentos de bases de datos*. Edición: Mc Graw-Hill

7.2 MATERIAL COMPLEMENTARIO RECOMENDADO

- Apuntes de Bases de Datos I editados por los profesores de la asignatura
- Cuaderno de ejercicios de Bases de Datos I editado por los profesores de la asignatura
- Guía de prácticas de Bases de Datos I editada por los profesores de la asignatura
- Recursos: www.oracle.com

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE

8.1 SISTEMA DE EVALUACIÓN

Según [MS04] el proceso evaluativo se ubica, al menos, en cuatro espacios de aprendizaje:

Los contenidos: cuestiones y redes conceptuales

Las prácticas: trabajos realizados dentro y fuera del aula durante el desarrollo de los créditos teóricos y prácticos

La participación en el trabajo de aula, aprendizaje virtual y tutorías

El proyecto o memoria final, portfolio, etc.

Para la evaluación del alumno en Bases de Datos I se sigue un tipo criterial, i.e. una evaluación que pretende determinar las competencias que posee el estudiante tras el desarrollo del proceso de aprendizaje. En este tipo de evaluación se recoge información mediante un instrumento, prueba, procedimiento o actividad para poder describir e identificar las competencias adquiridas por los estudiantes acerca de un dominio de referencia, descrito en la planificación docente inicial. Así, los criterios de evaluación deben correlacionarse con los objetivos/competencias y especificar principalmente un dominio conceptual claro, la adquisición de procedimientos, técnicas, instrumentos, habilidades y destrezas de ejecución profesional y académica, que concretan el nivel de capacidades personales y profesionales necesarias para el ejercicio profesional.

8.2 Criterios de evaluación

Teniendo en cuenta estas premisas, los criterios de evaluación aplicados a Bases de Datos I son:

- El estudiante ha de superar, en una prueba final, al menos al 30% de los compañeros del curso.

- Al final del proceso de aprendizaje, el estudiante habrá realizado correctamente el 70% de los ejercicios propuestos durante el curso.
- Al final del proceso de aprendizaje, el alumno habrá demostrado en los debates a través del Campus Virtual su capacidad de exposición significativa, consistente y localizada en el marco teórico del debate. Asimismo, habrá demostrado el dominio de la terminología usual de la asignatura.
- El alumno habrá asistido a las sesiones tutoriales, donde habrá demostrado su capacidad de colaborar en grupo y su capacidad de integrar los conocimientos, métodos y destrezas prácticas adquiridos para afrontar con éxito la resolución de casos reales propuestos por el profesor.
- Durante el proceso de aprendizaje, el alumno habrá entregado sus prácticas a tiempo.
- El alumno habrá participado un mínimo de 3 veces en clase durante el curso.
- El alumno habrá realizado al menos un trabajo en equipo de los propuestos en la asignatura, y habrá sido capaz de exponerlo durante 15 minutos en clase con claridad.
- El alumno habrá asistido por lo menos a una conferencia de las planteadas durante el curso con expertos del sector.

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas y procedimientos estadísticos relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos estadísticos y de resolución de problemas se ajustan a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. Los problemas y procedimientos estadísticos relacionados con la asignatura son resueltos con eficiencia y precisión. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de los resultados aceptables. • La actuación en las destrezas transferibles es generalmente buena. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.

Calificación	Rango	Pautas
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas y procedimientos estadísticos relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio estándares son usualmente desarrolladas con éxito razonable, aunque el significado y análisis de los resultados pueden no ser entendidos completamente. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspenseo	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas y procedimientos estadísticos relacionados con la asignatura no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente y el significado y análisis de los resultados no son entendidos generalmente. • Las destrezas transferibles están a un nivel deficiente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 4. *Procedimientos de evaluación*

Queremos hacer notar que para la obtención de matrícula de honor es necesario obtener un sobresaliente alto y hacer un trabajo complementario de calidad.

8.3 EVALUACIÓN DEL PROCESO DOCENTE

La evaluación del proceso docente se llevará a cabo tanto por procesos internos como externos. Internamente, los profesores de la asignatura realizarán encuestas periódicas sobre los conocimientos adquiridos por el alumnado, así como sobre el funcionamiento general de la asignatura. En ellas se pondrán de manifiesto las dificultades de aprendizaje y de transmisión de conocimientos experimentadas tanto por los alumnos como por los profesores.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Bases de Datos I. Aquí se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo de propuesto para el alumnado y el sistema y criterio de evaluación.

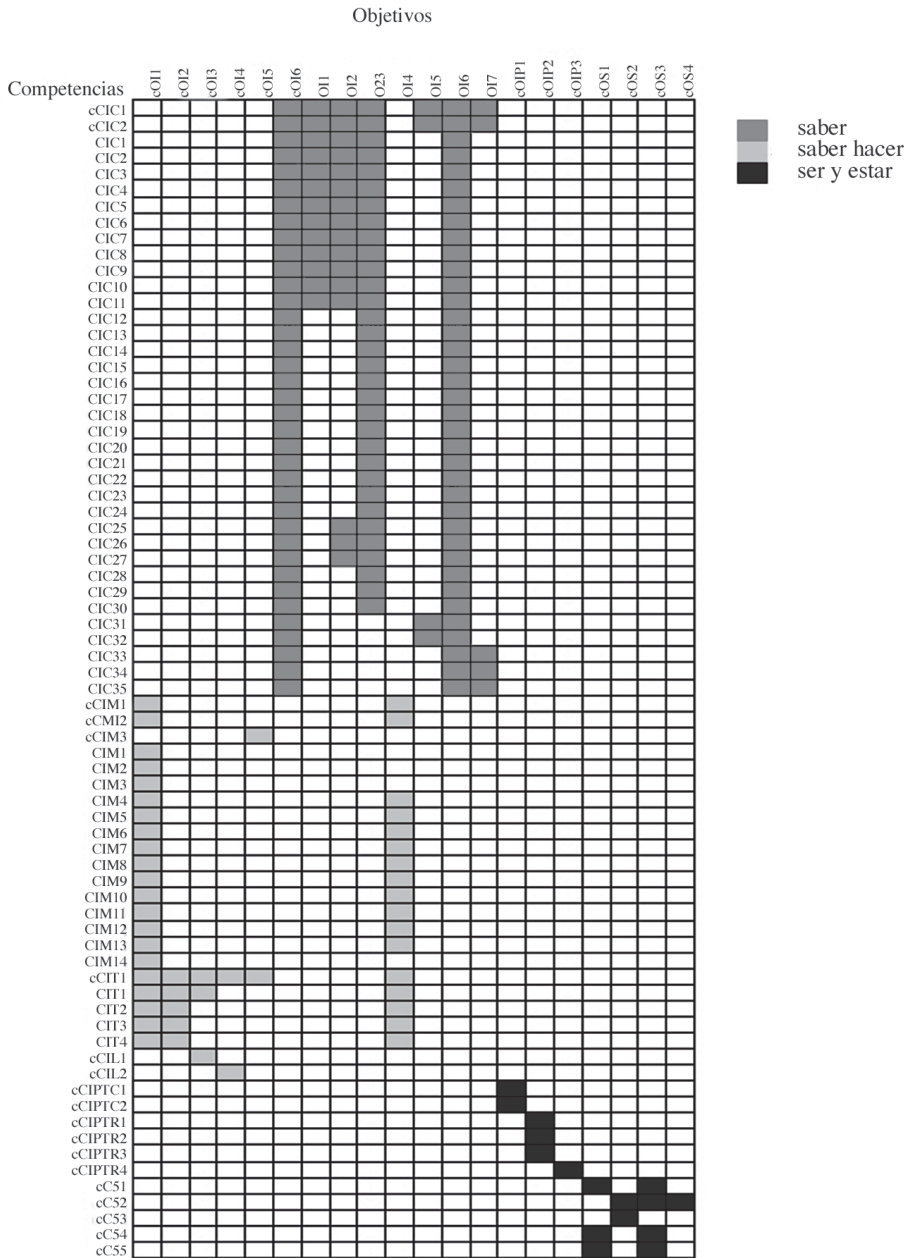


Ilustración 5. Análisis de coherencia de la guía docente de Bases de Datos I

9.1 Objetivos y unidades temáticas

A continuación, se muestra la correspondencia entre unidades temáticas y objetivos

OBJETIVOS	BLOQUES				
	B1	B2	B3	B4	B5
cOI1		3,4,5,6		9	
cOI2		3			
cOI3		3,4,5,6		9	
cOI4	1,2	3,4,5,6	7,8	9	10
cOI5	1,2	3,4,5,6	7,8	9	10
cOI6	0,1,2				
OI1	1,2	3,4,5,6	7,8	9	10
OI2	2	3		9	
OI3		3,4,5,6			
OI4	2	3,5			
OI5	2			9	
OI6		3	8		
OI7					10
cOIP1		3,4,5,6	8	9	
cOIP2		3,4,5,6	8	9	
cOIP3		3,4,5,6	8	9	
cOS1		3,4,5,6	8	9	
cOS2		3,4,5,6	8	9	
cOS3		3,4,5,6	8	9	
cOS4		3,4,5,6	8	9	

Cuadro 5. tabla de correspondencia de unidades temáticas y objetivos

10. BIBLIOGRAFÍA

[EICE04] *Libro Blanco EICE v1.1.4.*

[MS04] Martínez, M.A. & N. Sauleda Parés (2004) *Redes para Investigar el Currículo. Diseño del Aprendizaje en el EEES.*

ACM/IEEE (1991). Joint Currículo Task Force. *Computing Curricula 91.* ACM Press.

ACM/IEEE (2001). Joint Currículo Task Force. *Computing Curricula 2001.*
<http://www.acm.org/sigcse/cc2001>

ACM/AIS/AITP (1997). ACM-AIS-AITP Joint Task Force. *IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems.* ACM Press.

UNESCO-IFIP (1994). *A modular curriculum in Computer Science.* UNESCO.

UNESCO-IFIP (2000). *Informatics Curriculum Framework.* UNESCO.

GUÍA DOCENTE DE COMPUTABILIDAD

Pilar Arques Corrales; Rafael Molina Carmona; Ignacio Viché Clavel

*Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
(arques;rmolina;jviche)@dccia.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

El planteamiento de una propuesta docente coherente en el ámbito de la Informática precisa de una reflexión previa sobre los aspectos conceptuales y metodológicos globales (propuestas curriculares, naturaleza de la disciplina y metodología docente). El análisis de las recomendaciones curriculares planteadas por las instituciones más relevantes del mundo de la Informática (ACM (Association for Computing Machinery), IEEE (Institute for Electrical and Electronic Engineers)) es un paso indispensable para diseñar adecuadamente los contenidos de una asignatura concreta. Es importante captar aspectos y enfoques metodológicos globales tales como la relación entre el contenido teórico y práctico, el papel de los laboratorios y el perfil del profesional a formar, cuando se revisan dichas recomendaciones.

Estas sugerencias han sido elaboradas por medio de un proceso continuado de revisión, redefinición y asentamiento de esta joven disciplina como respuesta al rápido avance tecnológico al que se ve sometida. Mediante este proceso se ha realizado una progresiva maduración de la Informática como ciencia. Desde las propuestas iniciales de ACM e IEEE donde se le intenta dar por primera vez un carácter autónomo a la informática hasta las tendencias actuales recogidas en el Computing Curricula 2001 podemos destacar la visión de la Informática como una disciplina ingenieril con una dosis elevada de componente teórico y matemático. También es conveniente destacar que el ritmo de aparición de nuevas teorías, metodologías o herramientas es frenético.

La Informática Teórica aparece reflejada en Computing Curricula de 2001 dentro de una de las áreas básicas que deben contener todos los estudios superiores de informática, denominada Algoritmos y Complejidad. Esta área está dividida en los múltiples apartados que incluyen: Análisis algorítmico básico, Estrategias algorítmicas, Algoritmos de computación básica, Las clases de complejidad P y NP, Análisis avanzado de algoritmos, Algoritmos criptográficos o Algoritmos geométricos. Además se incluyen dos apartados íntimamente relacionados con la Informática Teórica: Teoría de la computabilidad básica (con descriptores como Máquinas de estado finito, Gramáticas de contexto libre, Funciones no computables, Problema de la parada, Implicaciones de la no computabilidad) y Teoría de Autómatas (con los descriptores Autómatas finitos deterministas, Autómatas finitos no deterministas, Equivalencia entre los modelos anteriores, Expresiones regulares, Lema del bombeo aplicado a expresiones regulares, Autómatas de pila, Relación entre los autómatas de pila y las gramáticas de contexto libre, Propiedades de las gramáticas de contexto libre, Máquinas de Turing, Máquinas de Turing no deterministas, Conjuntos y lenguajes, Jerarquía de Chomsky).

En el contexto español, debemos observar primero cuáles son las directrices generales propias de las titulaciones de Informática (Reales decretos 1459/1990 1460/1990, 1461/1990 de 26 de octubre, BOE 1990). En las titulaciones de Ingeniería en Informática e Ingeniería Técnica en Informática de Sistemas aparece como materia troncal de obligatoria inclusión en todos los planes de estudio, la materia “Teoría de Autómatas y Lenguajes Formales”, de 9 créditos, cuyos los descriptores son los siguientes:

- Máquinas secuenciales y autómatas finitos
- Máquinas de Turing
- Funciones recursivas
- Gramáticas y lenguajes formales
- Redes neuronales

En la Ingeniería Técnica en Informática de Gestión, sin embargo, la materia no aparece como troncal.

Siguiendo necesariamente estas directrices propias, en la Universidad de Alicante, se han distribuido los contenidos de “Teoría de Autómatas y Lenguajes Formales” en dos asignaturas: “Lenguajes, Gramáticas y Autómatas” y “Computabilidad”. Ambas asignaturas son troncales en las titulaciones de Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas, con 4,5 créditos cada una, y optativas en la Ingeniería Técnica en Informática de Gestión. La división de estos descriptores en las dos asignaturas es la siguiente:

- Lenguajes, Gramáticas y Autómatas
- Máquinas secuenciales y autómatas finitos
- Gramáticas y lenguajes formales

- Redes neuronales
- Computabilidad
- Máquinas de Turing
- Funciones recursivas

De esta manera, el ámbito científico de la asignatura Computabilidad queda fijado en los aspectos relacionados con la teoría de la misma, mientras que el de la asignatura Lenguajes, Gramáticas y Autómatas es el de modelos formales de la teoría de lenguajes propiamente dicha.

La necesidad de proporcionar una definición estricta de qué es la computación aparece a principios de siglo, cuando el matemático alemán David Hilbert propuso su famoso Décimo problema, en el Congreso Internacional de Matemáticas de París (1900). El problema planteado consistía en encontrar un procedimiento algorítmico para determinar si un sistema de polinomios con coeficientes enteros (ecuaciones Diofantinas) tiene soluciones enteras. Implícitamente, el problema suscita una cuestión fundamental, ya que cualquier problema matemático puede codificarse como un sistema de ecuaciones diofantinas. El Décimo Problema de Hilbert planteaba, por tanto, la pregunta de si existía un procedimiento algorítmico general que fuera capaz de responder cualquier problema matemático. Parte de la dificultad en responder a esta pregunta se encontraba en decidir qué era un “procedimiento algorítmico”. Fue para responder a esta necesidad, para lo que surgieron los trabajos de Turing, Church, Kleene o Post, que establecieron las bases, en 1936, de lo que más adelante constituiría la teoría de la computación.

Por resumirlo en una frase, la pregunta fundamental que responde la teoría de la computación es: ¿Qué es susceptible de ser (eficientemente) computado?

Los hitos fundamentales relacionados con los aspectos de modelos de computación, indecidibilidad y teoría de la complejidad, que constituyen las materias centrales de la disciplina, se basan en los artículos fundacionales de Turing y Church. La última es la aportación práctica más importante de la teoría de la computación a la informática en general, y surge también a partir del artículo de Turing, aunque algunas décadas después.

Desde la publicación de estos trabajos, la informática teórica ha evolucionado de manera muy notable. Hoy en día se considera que esta disciplina es, de una manera bastante amplia, el conjunto de modelos formales, métodos y técnicas aplicables al análisis de los algoritmos, lenguajes de programación y arquitecturas de computadores. Dentro de esta definición se pueden encuadrar un gran número de disciplinas:

- Modelos de computación secuencial
- Teoría de la complejidad
- Teoría de la indecidibilidad
- Sistemas de reescritura

- Programación funcional y cálculo Lambda
- Modelos de computación concurrente
- Autómatas finitos
- Lenguajes incontexturales
- Teoría de la complejidad de Kolmogorov
- Criptografía
- Lógica temporal y modal
- Programación Lógica
- Semántica de los lenguajes de programación
- Especificaciones algebraicas y validación formal de programas
- Redes neuronales

La siguiente tabla relaciona los tópicos de la asignatura Computabilidad, con los distintos perfiles profesionales:

Desarrollo de software y aplicaciones

- Determinar qué problemas son computables.

Arquitectura y diseño de software

- Identificación del ordenador como una máquina de estados.
- Determinar qué problemas son computables.

Diseño multimedia

- Determinar qué problemas son computables.

Ingeniería de comunicación de datos.

- Identificación del ordenador como una máquina de estados que puede comunicarse.

Diseño de redes de comunicación

- Identificación de una red como un conjunto de máquinas de estados que colaboran.

Asistencia técnica

- Aplicación de los principios de la computabilidad en el control de sistemas informáticos.

Ingeniería de integración y pruebas e implantación y pruebas

- Aplicación de los principios de la computabilidad en el diseño de test de prueba.

Consultoría en empresas de TI

- Aplicación de los principios de la computabilidad en la planificación.

Especialista en sistemas

- Identificación del ordenador como una máquina de estados.
- Determinar qué problemas son computables.
- Aplicación de los principios de la computabilidad en el control de sistemas informáticos.

Desarrollo de investigación y tecnología

- Determinar qué problemas son computables.

Dirección de TIC

- Identificación del ordenador como una máquina de estados.
- Determinar qué problemas son computables.
- Aplicación de los principios de la computabilidad en el control de sistemas informáticos.
- Aplicación de los principios de la computabilidad en la planificación.

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Computabilidad forma parte de los planes de estudio de Ingeniería Informática, Ingeniería Técnica en Informática de Sistemas e Ingeniería Técnica en Informática de Gestión, en los dos primeros como asignatura troncal y en el último como asignatura optativa, y se imparte en el segundo cuatrimestre. Los descriptores de dicha asignatura son: Máquinas de Turing y Funciones recursivas. Dicha asignatura tiene como incompatibilidad una única asignatura:

- **Álgebra:** Troncal de primer curso. Descriptores: Álgebra, Teoría de Conjuntos, Estructuras algebraicas y Teoría de matrices.

Aunque Álgebra es la única asignatura prerequisite para Computabilidad, existe una relación clara y estrecha con varias asignaturas de primero donde se imparten conceptos básicos, y cuyo entendimiento es necesario para abordar con éxito la materia que nos ocupa. Concretamente, dichas asignaturas son:

- **Matemática Discreta:** Troncal de primer curso. Descriptores: Matemática discreta: Aritmética modular, combinatoria, grafos.
- **Informática Básica:** Troncal de primer curso. Descriptores: Unidades funcionales: Memoria, procesador, periferia, lenguajes máquina y ensamblador, esquema de funcionamiento. Arquitectura Von Neumann. Electrónica. Sistemas digitales. Familias lógicas integradas. Periféricos.
- **Fundamentos de Programación I:** Troncal de primer curso. Descriptores: Introducción a la programación. Diseño de algoritmos. Análisis de algoritmos.
- **Fundamentos de Programación II:** Troncal de primer curso. Descriptores: Análisis y diseño de programas. Lenguajes de programación. Diseño descendente.

- **Lógica Computacional:** Obligatoria de primer curso. Descriptores: Lógica de primer orden (sintaxis y semántica). Sistemas de deducción. Demostración automática. Programación lógica.
- **Lenguajes, Gramáticas y Autómatas:** Troncal de segundo curso. Descriptores: Máquinas secuenciales y autómatas finitos. Gramáticas y lenguajes formales. Redes neuronales.

Por otro lado, Computabilidad, por tratar una materia básica en la informática, sirve como fundamento teórico de toda la Teoría de la Computación y, por lo tanto, es fundamental para el desarrollo de la mayoría de los contenidos de las asignaturas de cursos posteriores.

2. OBJETIVOS:

2.1 OBJETIVOS GENERALES:

2.1.1 Objetivos instrumentales generales:

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes:

- OI1: Reflexionar sobre el contexto de la Informática Teórica en general, y de la Computabilidad en particular a partir de los conceptos básicos en que se fundamenta toda la Informática.
- OI2: Tomar contacto con literatura general y específica de la Computabilidad.
- OI3: Dominar y emplear la terminología utilizada en la Computabilidad.
- OI4: Comprender, conocer, analizar y aplicar los conceptos fundamentales de la Computabilidad.
- OI5: Discernir entre lo que es un problema computable y otro que no lo es, independientemente de la eficiencia en la computación.
- OI6: Identificar qué problemas son computables.
- OI7: Distinguir entre conjuntos recursivamente enumerables, recursivos y no recursivos.
- OI8: Conocer las principales teorías para formalizar el concepto de algoritmo y de computación.
- OI9: Adquirir aquellos conceptos básicos, resultados y métodos asociados con la Computabilidad que son correquisitos o prerrequisitos de otras asignaturas.

2.1.2 Objetivos interpersonales generales:

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales:

Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS:

2.2.1 Competencias instrumentales:

Las competencias instrumentales se clasifican en habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas. Se han agrupado por bloques temáticos de la asignatura:

Habilidades cognitivas:

Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2 desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque 1: Teoría de Conjuntos y de Lenguajes

- CIC1: Dominar los aspectos fundamentales de la Teoría de Conjuntos.
- CIC2: Comprender las definiciones de tupla, función y predicado.
- CIC3: Dominar los aspectos fundamentales de la Teoría de Lenguajes Formales.
- CIC4: Comprender las definiciones de alfabeto, cadena y lenguaje.
- CIC5: Dominar los principales métodos de demostración formal y distinguir cuándo debe utilizarse cada uno de ellos.
- CIC6: Comprender el concepto de codificación de los elementos de un conjunto.
- CIC7: Comprender el concepto de numerabilidad y dominar los métodos de demostración asociados.
- CIC8: Discernir claramente entre conjuntos numerables y no numerables.

Bloque 2: Modelos de Computación

- CIC9: Comprender los conceptos de Algoritmo y de Computación.
- CIC10: Conocer los trabajos de Turing y Church, fundamentos de la Computabilidad: Tesis de Church-Turing.
- CIC11: Conocer el modelo conceptual básico de Máquina de Turing.
- CIC12: Conocer los distintos modelos de Máquinas de Turing.
- CIC13: Comprender y saber demostrar la equivalencia entre los diferentes modelos de Máquinas de Turing.
- CIC14: Comprender la relación entre lenguajes computables, funciones y Máquinas de Turing.
- CIC15: Caracterizar lenguajes con Máquinas de Turing.

- CIC16: Comprender el uso de lenguajes de programación como modelos de computación.
- CIC17: Dominar la sintaxis y la semántica del lenguaje L.
- CIC18: Programar en lenguaje L.
- CIC19: Caracterizar a las funciones L-Computables.
- CIC20: Aplicar la tesis de Church-Turing al lenguaje L, conjeturando la equivalencia entre Máquinas de Turing y Lenguaje L.
- CIC21: Comprender el formalismo de las Funciones Recursivas Primitivas (FRP) como modelo de computación.
- CIC22: Caracterizar a las FRP, utilizando el lenguaje R.
- CIC23: Utilizar constructores básicos y avanzados como forma alternativa de caracterizar FRP.
- CIC24: Comprender las limitaciones del Modelo de FRP, y la no equivalencia con los modelos de Máquinas de Turing y de Lenguaje L.

Bloque 3: Decidibilidad e Indecidibilidad

- CIC25: Comprender los conceptos de codificación de programas y de ejecuciones de programas.
- CIC26: Comprender el concepto de descripción instantánea.
- CIC27: Conocer el concepto de Programa Universal.
- CIC28: Asimilar algunos de los problemas clásicos de la computabilidad: El problema de la parada, y el problema del castor afanoso.
- CIC29: Comprender la diferencia entre conjuntos recursivos, recursivamente enumerables y no recursivamente enumerables.
- CIC30: Conocer las relaciones entre conjuntos recursivos, conjuntos recursivamente enumerables y funciones.
- CIC31: Conocer algunos conjuntos no recursivos (conjunto K) y no recursivamente enumerables (conjunto $\neg K$, conjunto TOT).
- CIC32: Comprender el proceso de reducción de conjuntos como mecanismo para demostrar que un conjunto es recursivo o recursivamente enumerable.
- CIC33: Comprender el teorema de Rice.

Capacidades metodológicas:

Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3, incluidas a nivel general en el capítulo introductorio de este ensayo.

Bloque 1: Teoría de Conjuntos y de Lenguajes

- CIM1: Aplicar los métodos de demostración formal, identificando cuál utilizar en cada caso.

- CIM2: Codificar y decodificar adecuadamente pares de números.
- CIM3: Aplicar los métodos de demostración de la numerabilidad o la no numerabilidad de un conjunto adecuadamente.

Bloque 2: Modelos de Computación

- CIM4: Diseñar Máquinas de Turing reconocedoras de lenguajes.
- CIM5: Diseñar Máquinas de Turing calculadoras de funciones.
- CIM6: Demostrar que un lenguaje es recursivo o recursivamente enumerable, utilizando Máquinas de Turing.
- CIM7: Programar en Lenguaje L.
- CIM8: Demostrar la computabilidad o no de una función, utilizando Lenguaje L.
- CIM9: Demostrar si una función es Recursiva Primitiva utilizando los constructores básicos y avanzados.

Bloque 3: Decidibilidad e Indecidibilidad

- CIM10: Codificar y decodificar programas en lenguaje L.
- CIM11: Codificar y decodificar ejecuciones de programas.
- CIM12: Aplicar el concepto de Programa Universal para demostrar si una función es L-Computable o Recursiva Primitiva.
- CIM13: Demostrar que un conjunto es recursivo, construyendo un procedimiento de decisión.
- CIM14: Demostrar que un conjunto es recursivamente enumerable, construyendo un procedimiento de semidecisión.
- CIM15: Demostrar que un conjunto es no recursivo, aplicando reducción.
- CIM16: Demostrar que un conjunto es no recursivamente enumerable aplicando reducción.

Destrezas tecnológicas:

- CIT1: Manejar simuladores de Máquinas de Turing.
- CIT2: Manejar intérpretes de los lenguajes típicos de la Computabilidad: Lenguajes L y R.

Destrezas lingüísticas:

- CIL1: Conocer y saber utilizar la jerga relativa a la computabilidad
- CIL2: Utilizar adecuadamente los términos conjunto, función, tupla, alfabeto, lenguaje.
- CIL3: Utilizar adecuadamente los términos conjunto numerable, conjunto no numerable, conjunto recursivo, conjunto no recursivo, conjunto recursivamente enumerable, conjunto no recursivamente enumerable.
- CIL4: Utilizar correctamente las distintas notaciones asociadas a cada modelo de computación.

2.2.2 Competencias interpersonales

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas a compromiso con el trabajo.

Competencias para tareas colaborativas

Estas competencias se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo

Estas competencias se refieren a las competencias comunes dadas de el artículo introductorio de este libro. Más concretamente, son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Estas competencias hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Están etiquetadas como cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

La asignatura de Computabilidad es una asignatura Troncal de segundo curso, tiene como prerequisite la asignatura de Álgebra de primer curso. Por otro lado, al ser una asignatura que se imparte a alumnos de Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas, el modelo de computación que se utiliza para explicar los conceptos más abstractos y complejos de la asignatura (que se estudian en el bloque de decidibilidad e indecidibilidad) es el Lenguaje L, junto con las Funciones Recursivas Primitivas. Por ello, es conveniente que los alumnos tengan unas nociones básicas de Fundamentos de Programación, a pesar de no ser esta última asignatura prerequisite para poder cursar Computabilidad.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerequisites necesarios para el estudio y entendimiento de esta asignatura se cubren en las asignaturas de Álgebra y Fundamentos de Programación de primer curso. Aunque Fundamentos de Programación no es prerequisite, las nociones de programación que se necesitan son tan básicas que normalmente cualquier alumno de una Ingeniería Informática los ha comprendido. No obstante, en el primer tema de la asignatura se estudian una serie de conceptos básicos de teoría de conjuntos, lenguajes, métodos de demostración,..., necesarios para la comprensión de la materia.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS Y DE APRENDIZAJE.

TEORÍA DE CONJUNTOS Y DE LENGUAJES.

- Tema 1. Preliminares.

MODELOS DE COMPUTACIÓN.

- Tema 2. Máquinas de Turing.
- Tema 3. Funciones L-computables.
- Tema 4. Funciones recursivas primitivas.

DECIDIBILIDAD E INDECIDIBILIDAD.

- Tema 5. Un programa Universal.
- Tema 6. Decidibilidad e indecidibilidad.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO.

TEORÍA DE CONJUNTOS Y DE LENGUAJES.

- Tema 1. Preliminares.
 - 1.1. Conjuntos.
 - 1.2. Tuplas
 - 1.3. Funciones
 - 1.4. Predicados
 - 1.5. Alfabetos, cadenas y lenguajes
 - 1.6. Métodos de demostración
 - Reducción al absurdo
 - Demostración por inducción
 - 1.7. Codificación.
- Codificación de pares de números
 - 1.8. Numerabilidad.
 - 1.9. Diagonalización.
 - 1.10. Problemas.

MODELOS DE COMPUTACIÓN.

- Tema 2. Máquinas de Turing.
 - 2.1. Introducción.
 - 2.2. El problema de definir un lenguaje.
 - Expresiones regulares.
 - 2.3. El modelo conceptual de Máquina de Turing.
 - Funcionamiento de una MT.
 - Representación de una MT.
 - Definición formal de una MT
 - 2.4. Equivalencia de modelos de MT.
 - Máquina de Turing multipista.

- Máquina de Turing con registro acotado de memoria.
- Máquina de Turing multicinta.
- Máquina de Turing no determinista
- 2.5. Lenguajes computables y funciones.
 - Lenguajes recursivamente enumerables y recursivos.
 - Las MT como computadores de funciones de naturales.
- 2.6. Caracterización de lenguajes con MT generadoras.
 - Caracterización de los lenguajes recursivos.
 - Caracterización de los lenguajes recursivamente enumerables.
- 2.7. Problemas.

- Tema 3. Funciones L-computables.
 - 3.1. Introducción de un lenguaje de programación.
 - El lenguaje L.
 - Algunos ejemplos de programas.
 - 3.2. Definiciones formales.
 - Notación BNF.
 - Definición de L usando la notación BNF.
 - Semántica de L.
 - 3.3. Definición de macros.
 - 3.4. Ejemplos de programas.
 - 3.5. Funciones L-computables.
 - 3.6. Macros genéricas.
 - Asignación.
 - Comparación.
 - 3.7. Tesis de Church aplicada al lenguaje L.
 - 3.8. Problemas.

- Tema 4. Funciones recursivas primitivas
 - 4.1. Definición de las funciones recursivas primitivas mediante notación matemática.
 - 4.2. Introducción al lenguaje R.
 - 4.3. Definición formal de R.
 - Sintaxis del lenguaje R.
 - Semántica del lenguaje R.
 - Funciones recursivas primitivas.
 - 4.4. Constructores avanzados.
 - Predicados.
 - Iteradores acotados.
 - Cuantificadores acotados.
 - Minimización acotada.

- 4.5. Una función L-computable total no recursiva primitiva.
 Minimización no acotada.
 Funciones recursivas μ .
- 4.6. Problemas.

DECIDIBILIDAD E INDECIDIBILIDAD.

- Tema 5. Un programa universal.
 - 5.1. Numeración de Gödel.
 Función de codificación.
 Función de descodificación.
 - 5.2. Codificación de programas L.
 Codificación de instrucciones.
 Codificación y descodificación de programas.
 - 5.3. Codificación de ejecuciones de programas.
 Codificación de instantáneas.
 La función Inst.
 - 5.4. El programa Universal.
 El predicado PASOS.
 El programa Universal.
 - 5.5. Problemas.
- Tema 6. Decidibilidad e indecidibilidad.
 - 6.1. El problema de la parada.
 - 6.2. El castor afanoso.
 - 6.3. Conjuntos decidibles e indecidibles.
 Conjuntos recursivos.
 Conjuntos recursivamente enumerables.
 Conjuntos recursivos y recursivamente enumerables.
 Conjuntos recursivamente enumerables y funciones.
 - 6.4. Conjuntos no recursivos.
 El conjunto K.
 - 6.5. Conjuntos no recursivamente enumerables.
 El conjunto $\neg K$.
 El conjunto TOT.
 - 6.6. Reducción.
 El conjunto K_0 no es recursivo.
 El conjunto NO_VACIO no es recursivo.
 El conjunto VACIO no es recursivamente enumerable.
 El conjunto FINITO no es recursivamente enumerable.
 - 6.7. El teorema de Rice.
- 7. Problemas.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

En los últimos años, como respuesta a la baja efectividad del método tradicional basado únicamente en la sucesión de lecciones magistrales, han surgido nuevos enfoques que propician el pensamiento creativo, enseñando a aprender por encima de enseñar conocimientos. Planteamos utilizar un modelo de enseñanza en el que la clase magistral tiene un papel primordial en la transmisión de conocimientos pero se complementa con otras actividades de no menor importancia. Se proponen las siguientes actividades:

Clases de teoría: esta actividad ha sido ampliamente criticada e incluso rechazada. Sin embargo, según las investigaciones al respecto, la lección magistral, en comparación con otros métodos, es la técnica más eficaz y económica de transmitir y sintetizar información de diversas fuentes. Existen opiniones favorables que abogan por introducir técnicas auxiliares para resolver la pasividad inducida en el alumno por el método de exposición oral. Para ello se debe seguir un secuenciamiento adecuado:

- Despertar la motivación
- Informar sobre los objetivos
- Estimular la actualización de aprendizajes previos
- Presentar los conocimientos y mostrar las habilidades objeto de aprendizaje
- Facilitar orientaciones para el aprendizaje
- Intensificar la retención
- Fomentar la transferencia
- Proporcionar retroalimentación

Para favorecer la participación, las clases magistrales se complementan con materiales de apoyo, tanto impresos: apuntes de la asignatura, programación de la asignatura, como audiovisuales: presentaciones con ordenador, transparencias, páginas Web, etc. Además el empleo de la pizarra es fundamental para los desarrollos detallados y ejemplos aclaratorios. Se comentarán los errores más comunes que se cometen al ir adentrándose en la asignatura. Al final de algunas sesiones, se propondrán ejercicios que favorezcan la aplicación de los conceptos introducidos en la clase.

Prácticas: la importancia de la práctica en los estudios de informática es crucial. El trabajo personal en los laboratorios de computación permite fijar los conocimientos que se han adquirido en las clases expositivas y mediante el material de apoyo. Las prácticas se realizan de forma individual. Con las prácticas de laboratorio se intenta impulsar el aprendizaje, experimentación, asimilación y ampliación de algunos de los contenidos de la asignatura de Computabilidad con el uso del ordenador. Para comprender las metodologías estudiadas, se utilizarán simuladores de Máquinas de Turing, Lenguaje L y funciones recursivas primitivas.

Trabajos complementarios: como extensión de los conocimientos teóricos y prácticos adquiridos, se propondrán trabajos complementarios de realización voluntaria que incidirán en la nota final de la asignatura. Dichos trabajos pueden ser de índole teórico-práctica y se realizan en grupos para fomentar el trabajo colaborativo.

Tutorías: el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como dudas concretas sobre el contenido de la asignatura. Además, resulta de gran utilidad el correo electrónico y el Campus Virtual, ya que hay muchas dudas que son susceptibles de ser resueltas por este medio. No obstante, hay ocasiones en que cuando la duda es de compleja resolución, es preferible la asistencia a tutorías, ya que es la forma de asegurarse de que se ha entendido la explicación.

6. Estrategias de aprendizaje

Junto a los medios tradicionales como las transparencias, apuntes y presentaciones por ordenador, las páginas Web y el Campus Virtual ofrecen innumerables posibilidades que no hay que dejar pasar. La información que el alumno necesita se ha dejado disponible en dos sitios: Campus virtual y página Web de la asignatura. Todo el material necesario para las clases de teoría se deja disponible en el Campus virtual con varios días de antelación. Entre otras cosas, en dicha página es posible encontrar:

- Novedades: esto es un tablón de anuncios, mediante el cual el alumno puede estar perfectamente informado de cualquier tema relacionado con la asignatura, además de recordar los plazos de entrega de los trabajos.
- Clases teóricas: todo lo relacionado con el temario, los guiones de teoría, los objetivos, la bibliografía y forma de evaluación.
- Enlaces de interés: aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en algunos contenidos de la materia.
- Clases prácticas: material necesario para la realización de las prácticas: enunciados de las mismas y software necesario para poder realizarlas.

En la página Web de la asignatura, se deja disponible todo el material que por sus características no es posible dejar en el Campus Virtual.

Las distintas estrategias seguidas para llevar a cabo los distintos planteamientos del apartado anterior son:

- Clases teóricas: el alumno debe leer con anterioridad a la sesión los apuntes del tema a tratar. Ello permite que puedan seguir con más facilidad las explicaciones del profesor así como preguntar dudas sobre detalles que no se hayan asimilado correctamente. Es obvio que para tener dudas acerca de una materia, hay que haber preparado algo sobre la misma.

- Practicas de laboratorio: durante la sesión de prácticas, el alumno debe seguir las directrices fijadas por el profesor así como preguntar acerca de cualquier detalle que no haya quedado suficientemente establecido mediante el enunciado de la práctica y la explicación del profesor. Es fundamental motivar al alumno para la realización de las prácticas cumpliendo el calendario de entrega.
- Trabajos complementarios: a lo largo del curso se dejarán en el Campus Virtual y se comentarán la clase propuestas de trabajos relacionados con los distintos bloques de la asignatura. Es de destacar que dadas las características de la materia, muchos estudiantes encuentran motivador la realización de trabajos con una cierta componente de investigación. La realización de estos trabajos incidirá de manera positiva en la calificación de la asignatura siempre y cuando se haya aprobado la misma.

7. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se resume cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en el aula, donde el papel del profesorado es esencial, y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura de forma autónoma por los estudiantes.

HORAS DE ACTIVIDADES PRESENCIALES				
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos / Tutorías docentes
Bloque 1: Teoría de Conjuntos y Lenguajes	Presentación asignatura	0,5	0	0
	Tema 1	2,5	3	0
	Ejercicio de autoevaluación	0	0	1
Bloque 2: Modelos de computación	Tema 2	3	3	0
	Tema 3	3	3	0
	Tema 4	3	3	0
	Ejercicio de autoevaluación	0	0	1
Bloque 3: Decidibilidad e Indecidibilidad	Tema 5	4,5	4,5	0
	Tema 6	4,5	4,5	0
	Ejercicio de autoevaluación			1
TOTAL: 45		21	21	3

Cuadro 1. Plan de trabajo de la asignatura, número de horas presenciales

HORAS DE ACTIVIDADES NO PRESENCIALES				
ACTIVIDAD		Estudio de la asignatura	Realización de prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos/Tutorías docentes
Bloque 1: Teoría de Conjuntos y Lenguajes	Presentación asignatura	1	0	0
	Tema 1	4	6	0
	Ejercicio de autoevaluación	0	0	0,25
Bloque 2: Modelos de computación	Tema 2	5	6	0
	Tema 3	5	6	0
	Tema 4	5	6	0
	Ejercicio de autoevaluación	0	0	0,25
Bloque 3: Decidibilidad e Indecidibilidad	Tema 5	8	9	0
	Tema 6	8	9	0
	Ejercicio de autoevaluación			0,25
TOTAL: 78,75		36	42	0,75

Cuadro 2. Plan de trabajo de la asignatura, número de horas no presenciales

8. BIBLIOGRAFÍA

8.1 BIBLIOGRAFÍA BÁSICA

Davis, M.; Sigal, R. and Weyuker, E. (1994) *Computability, Complexity and Lenguajes*. Academic Press.

Gallardo, G.; Arques, P. y Lesta, I. (2003) *Introducción a la teoría de la Computabilidad..* Publicaciones Universidad de Alicante.

8.2 BIBLIOGRAFÍA COMPLEMENTARIA

Hopcroft, J.; Motwani, R. y Ullman, J. (2002) *Introducción a la Teoría de Automatas, Lenguajes y Computación*. Addison Wesley.

Michael Sipser (1997) *Introduction to the Theory of Computation*. PWS Publishing Company.

8.3 Otros recursos

<http://ilc.sourceforge.net/>. Intérprete de lenguaje L.

9. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

9.1 PROCEDIMIENTOS DE EVALUACIÓN

El proceso de evaluación debe permitir obtener información útil acerca de los conocimientos que ha adquirido el alumno al cursar la materia para poder tomar una decisión al respecto. La evaluación de esta asignatura abarca, de manera conjunta, la parte teórica y la parte práctica. Para ello se realizan las siguientes pruebas: un test de conocimientos básicos de la asignatura prerrequisito (Álgebra), dos hojas de problemas (a mitad y a final de curso) y un examen final de carácter teórico-práctico.

El examen final puntúa sobre 10 puntos, las dos hojas de problemas 1 punto total y el test previo, 0,25 puntos, de manera que la nota máxima a alcanzar es el 12,25 puntos. A pesar de ello el aprobado está en 5 puntos, es decir, tanto las hojas de problemas como el test previo se añaden a la nota del examen siempre que ésta supere 4 puntos. La realización de las hojas de problemas y del test previo es totalmente voluntaria, aunque para su entrega se les exige una asistencia regular a clases prácticas. Tanto en las clases prácticas como en la realización de las hojas de problemas trabajan en grupos de hasta 4 personas, es decir, entregan una sola hoja de problemas por grupo de prácticas.

9.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • Los problemas y cuestiones relacionados con la asignatura son resueltos con eficiencia y corrección; los procedimientos algorítmicos y de resolución de problemas son ajustados a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • La comprensión conceptual es notable. Los problemas y algoritmos relacionados con la asignatura son resueltos con eficiencia y corrección; los procedimientos algorítmicos y de resolución de problemas son generalmente ajustados a la naturaleza del problema. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de los resultados aceptables. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.

Calificación	Rango	Pautas
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento de los contenidos de la asignatura es básico. • La comprensión conceptual es aceptable. • Los problemas propuestos se resuelven adecuadamente en la mayor parte de los casos. • La capacidad para programar en lenguaje ensamblador es básica pero suficiente. • La actitud y participación en clase es aceptable.
Suspenso	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento de la asignatura no es aceptable. • No se han resuelto la mayor parte de los problemas propuestos. • La capacidad para programar en ensamblador no llega al mínimo deseable. • La actitud y participación en las clases es escasa o nula.

Cuadro 3. *Procedimientos de evaluación*

10. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

El análisis de coherencia permite condensar en un todo las relaciones que existen entre los objetivos y las competencias a desarrollar con el plan de trabajo propuesto para el alumnado y el sistema de evaluación empleado. El análisis se ha estructurado en tres tablas separadas en función del tipo de objetivo y competencia analizados en cada caso: instrumental, interpersonal y sistémico.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales	Teoría de conjuntos y de Lenguajes	Modelos de Computación	Decidibilidad e indecidibilidad		
OI1, OI2, OI3, OI4,	De CIC1 a CIC8 De CIM1 a CIM3 CIL1 y CIL2	1			Enseñanza presencial (lección magistral/ Trabajo en aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios). Tutorías de atención al alumno	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión de los conceptos relativos a teoría de conjuntos y lenguajes.
OI3 OI4 OI8 OI9	De CIC9 a CIC24 De CIM4 a CM9 CIT1 y CIT2, CIL1 y CIL4		2 3 4		Enseñanza presencial (lección magistral/ Trabajo en aula en grupos (Prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios). Tutorías de atención al alumno	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión y aplicación de los conceptos relativos a los distintos métodos de computación explicados.
OI4, OI5 OI6 OI7	De CIC25 a CIC33 De CIM10 a CIM16 CIL1 y CIL3			5 6	Enseñanza presencial (lección magistral/ Trabajo en aula en grupos (Prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios). Tutorías de atención al alumno.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión y aplicación de los conceptos relativos a la decidibilidad e indecidibilidad.

Cuadro 4. *Análisis de coherencia: competencias instrumentales*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales	Teoría de conjuntos y de Lenguajes	Modelos de Computación	Decidibilidad e indecidibilidad		
cOIP1 cOIP2 cOIP3	CIPTC1, CIPTC2, CIPTR1	1			Enseñanza presencial (trabajo en aula en grupos). Enseñanza no presencial (realización de ejercicios, revisión de libros).	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.
cOIP1 cOIP2 cOIP3	CIPTC1, CIPTC2, CIPTR1, CIPTR2, CIPTR3, CIPTR4		2 3 4		Enseñanza presencial (trabajo en aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías de atención al alumno individualizadas y organizadas	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.
cOIP1 cOIP2 cOIP3	CIPTC1, CIPTC2, CIPTR1, CIPTR2, CIPTR3, CIPTR4			5 6	Enseñanza presencial (trabajo en aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías de atención al alumno individualizadas y organizadas	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.

Cuadro 5. Análisis de coherencia: competencias interpersonales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Teoría de conjuntos y de Lenguajes	Modelos de Computación	Decidibilidad e indecidibilidad		
OS4	CS1, CS2	1			<p>Enseñanza presencial (lección magistral/ Trabajo de aula en grupos). Enseñanza no presencial (revisión de libros/ realización de ejercicios). Tutorías de atención al alumno.</p>	<p><u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de asimilación del contexto de los conceptos básicos de la teoría de conjuntos y lenguajes.</p>
OS2, OS3, OS4	CS1, CS2, CS3		2 3 4		<p>Enseñanza presencial (lección magistral/ Trabajo de aula en grupos/ prácticas en laboratorio). Enseñanza no presencial (revisión de libros/ realización de ejercicios). Tutorías de atención al alumno</p>	<p><u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión de distintos modelos de computación y su equivalencia.</p>
OS1, OS2, OS3	CS1, CS2, CS3, CS4, CS5			5 6	<p>Enseñanza presencial (lección magistral/ Trabajo de aula en grupos/ prácticas en laboratorio). Enseñanza no presencial (revisión de libros/ realización de ejercicios). Tutorías de atención al alumno.</p>	<p><u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión de las limitaciones de los modelos de computación.</p>

Cuadro 6. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE ESTRUCTURAS DE COMPUTADORES

Ángel Grediaga Olivo; Antonio Soriano Payá; Lliüsa Rico Soliveres

*Departamento de Tecnología Informática y Computación
Universidad de Alicante*

angel.grediaga@ua.es, (soriano; rico)@dtic.ua.es

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

A la hora de elaborar la propuesta docente que nos ocupa, se ha de tener en cuenta cuál ha sido la evolución de las distintas recomendaciones curriculares de informática publicadas por instituciones de prestigio internacional, desde las propuestas iniciales de la ACM (Association for Computing Machinery) [ACM68] y el IEEE (Institute for Electrical and Electronic Engineers) [EC77], de 1968 y 1977, respectivamente –donde por primera vez se intenta dar un carácter autónomo a la informática– hasta las tendencias actuales recogidas en el Computing Curricula 2001 [ACM01b], realizado conjuntamente por el IEEE y la ACM. Analizando esta evolución se observa que los avances técnicos han hecho que muchas de las materias de la Ingeniería Informática de la década pasada hayan ganado importancia dando lugar a cuatro grandes perfiles genéricos, tal y como se refleja en el Computing Curricula 2001: Computer Science, Computer Engineering, Software Engineering e Information Systems.

La propuesta curricular conjunta de ACM e IEEE [ACM-IEEE, 2001] plantea a nivel introductorio el curso Introduction to the Computer (CS111) y a nivel intermedio el curso Computer Architecture (CS220). Asimismo, dedica el área de conocimiento Architecture and Organization (AR) a la estructura y arquitectura de computadores distribuyendo las materias en un conjunto de contenidos fundamentales y optativos. Las materias de este área son:

- (AR1) Lógica digital y sistemas digitales: introducción e historia de la arquitectura de computadores, bloques fundamentales de construcción (puertas lógicas, flip-flops, contadores, registros, PLA), expresiones lógicas, minimización, suma de productos, notación para transferencia de registros y consideraciones físicas (retraso de puertas, fan-in, fan-out). Esta materia es de tipo fundamental y se recomienda 6 horas de dedicación.
- (AR2) Representación de datos a nivel máquina: bits, bytes y palabras, representación numérica de datos y bases numéricas, sistemas de coma fija y coma flotante, representaciones con signo y en complemento a dos, representación de datos no numéricos (códigos de caracteres, datos gráficos) y representación de registros y matrices. Esta materia es de tipo fundamental y se recomienda 3 horas de dedicación.
- (AR3) Organización de la máquina a nivel ensamblador: organización básica de una máquina von Neumann, unidad de control (búsqueda, decodificación y ejecución de instrucciones), conjuntos y tipos de instrucciones (manipulación de datos, control, entrada/salida), lenguajes máquina y ensamblador, formatos de instrucciones, modos de direccionamiento, llamadas a subrutinas y mecanismos de retorno e interrupciones y entrada/salida. Materia de tipo fundamental, se recomienda 9 horas de dedicación.
- (AR4) Organización y arquitectura del sistema de memoria: sistemas de almacenamiento y tecnologías, codificación, compactación de los datos, integridad, jerarquía de memoria, organización y operaciones de la memoria principal, latencia, tiempo de ciclo, ancho de banda y entrelazado, memoria caché (mapa de direccionamiento, tamaño de bloque, políticas de reemplazo y almacenamiento), memoria virtual (tabla de páginas, TLB) y manejo de fallos y fiabilidad. Esta materia es de tipo fundamental y se recomienda 5 horas de dedicación.
- (AR5) Interfaces y comunicación: fundamentos de entrada/salida: protocolos, buffering, entrada/salida por programa, entrada/salida guiada por interrupciones, estructura de interrupciones (prioridad, vectorizadas), reconocimiento de interrupciones, almacenamiento externo, organización física y manejadores, buses (protocolos de bus, arbitraje, acceso directo a memoria), introducción a las redes de computadores, soporte multimedia y arquitecturas RAID. Es de tipo fundamental y se recomienda 3 horas de dedicación.
- (AR6) Organización funcional: implementación de rutas de datos sencillas, unidad de control (implementaciones cableadas y microprogramadas), pipeline de instrucciones e introducción al paralelismo a nivel de instrucción (ILP). Materia de tipo fundamental. Se recomienda 7 horas de dedicación.

- (AR7) Multiprocesamiento y arquitecturas alternativas: introducción a SIMD, MIMD, VLIW, EPIC, arquitectura sistólica, interconexión de redes (hipercubo, barajadas, malla, crossbar), sistemas de memoria compartida, coherencia de caché y modelos y consistencia de memoria. Es de tipo fundamental y se recomienda 3 horas de dedicación.
- (AR8) Mejoras del rendimiento: arquitectura superescalar, predicción de ramificaciones, prebúsqueda, ejecución especulativa, multihilo y escalabilidad. Materia de tipo optativa.
- (AR9) Arquitectura de redes y sistemas distribuidos: introducción a LANs y WANs, diseño de protocolos de capas, ISO/OSI, IEEE 802, impacto de la arquitectura en los algoritmos distribuidos, computación en red y multimedia distribuida. Materia de tipo optativa.

De todo lo expuesto se deduce que, aunque los tópicos relacionados con esta asignatura van a ser básicos en la formación de cualquier ingeniero informático, sea cual sea su perfil académico y su futuro perfil profesional, toman especial relevancia en los siguientes perfiles, englobados a su vez en tres perfiles profesionales generales.

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: los lenguajes de descripción hardware, proporcionan la flexibilidad de la que adolece el hardware y permiten el acercamiento al diseño hardware.
- Arquitectura y diseño de software: debajo del software existe un hardware sobre el cual debe correr el software, resulta fundamental conocer en profundidad el hardware para obtener el mayor rendimiento del software.
- Diseño multimedia: los dispositivos que conforman los desarrollos multimedia cada vez son más reconfigurables y necesitan de la computación reconfigurable para dotarlos de flexibilidad y competitividad.

Perfil profesional de sistemas:

- Desarrollo de productos: diseño de circuitos analógicos y digitales y procesamiento de la señal.
- Diseño de redes de comunicación: conocimiento de LANs y WANs, diseño de protocolos de capas, ISO/OSI, IEEE 802, así como de las arquitecturas RAID.
- Asistencia técnica: es fundamental conocer las partes que componen el sistema y cómo intervienen en las anomalías y errores de funcionamiento.
- Ingeniería de integración y pruebas e implantación y pruebas: las herramientas y metodologías de la integración de sistemas forman parte fundamental del desarrollo hardware de las estructuras y arquitecturas del computador.

Perfil profesional de gestión y explotación de tecnologías de la información:

- Consultoría de empresas de TI: la estrategia empresarial requiere de técnicas de planificación.
- Especialista en sistemas: En dicho perfil es necesaria una base fuerte de lógica digital y sistemas digitales, además de los lenguajes de descripción hardware y de las nuevas tecnologías reconfigurables.
- Desarrollo de investigación y tecnología: la participación y gestión de proyectos ya sean de pequeña envergadura o proyectos de investigación internacionales, requieren de estrategia, planificación y control.
- Dirección de TIC (marketing, proyectos, dirección general): este perfil requiere un conocimiento amplio del resto de áreas tecnológicas de las Tecnologías de la Información y las Comunicaciones (TIC).

Esta adecuación de los tópicos relacionados con la asignatura Ampliación de Estadística a los distintos perfiles profesionales queda resumida en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimiento y habilidad para utilizar la descripción hardware de dispositivos. • Conocimiento del lenguaje ensamblador.
Arquitectura y diseño de software	<ul style="list-style-type: none"> • Conocimiento de las arquitecturas que mejoran el rendimiento.
Diseño multimedia	<ul style="list-style-type: none"> • Conocimiento y habilidad para utilizar la computación reconfigurable y sus arquitecturas.
Ingeniería de comunicación de datos.	<ul style="list-style-type: none"> • Conocimiento y habilidad en el manejo de la entrada/salida de los computadores y los métodos de transferencia de datos.
Diseño de redes de comunicación	<ul style="list-style-type: none"> • Conocimiento del impacto de la arquitectura en los algoritmos distribuidos, computación en red y multimedia distribuida.
Asistencia técnica	<ul style="list-style-type: none"> • Conocimiento de las partes que componen el sistema y su implicación en las anomalías de funcionamiento. • Conocimiento y habilidad en la configuración de equipos informáticos.
Ingeniería de integración y pruebas e implantación y pruebas	<ul style="list-style-type: none"> • Conocimiento de la metodología y herramientas de la ingeniería de sistemas. • Conocimiento y habilidad de las Herramientas y métodos de prueba.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Consultoría en empresas de TI	<ul style="list-style-type: none"> • Conocimiento y habilidad en técnicas de planificación.
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimiento de lógica digital y sistemas digitales. • Conocimiento de los lenguajes de descripción hardware.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento y habilidad para el diseño de sistemas de alto rendimiento. • Conocimiento y habilidad en técnicas de de co-diseño.
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento y habilidad en el trabajo en grupo Conocimiento y habilidad en técnicas de planificación.

Cuadro 1. Adecuación de los tópicos relacionados con la asignatura Estructura de computadores

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

Las asignaturas del plan de estudio de la Ingeniería Informática (II) y de la Ingeniería Técnica Informática Especialidad Sistemas (ITIS), de perfil cercano, relacionadas con Estructuras de Computadores (EC) se pueden observar en la siguiente figura:

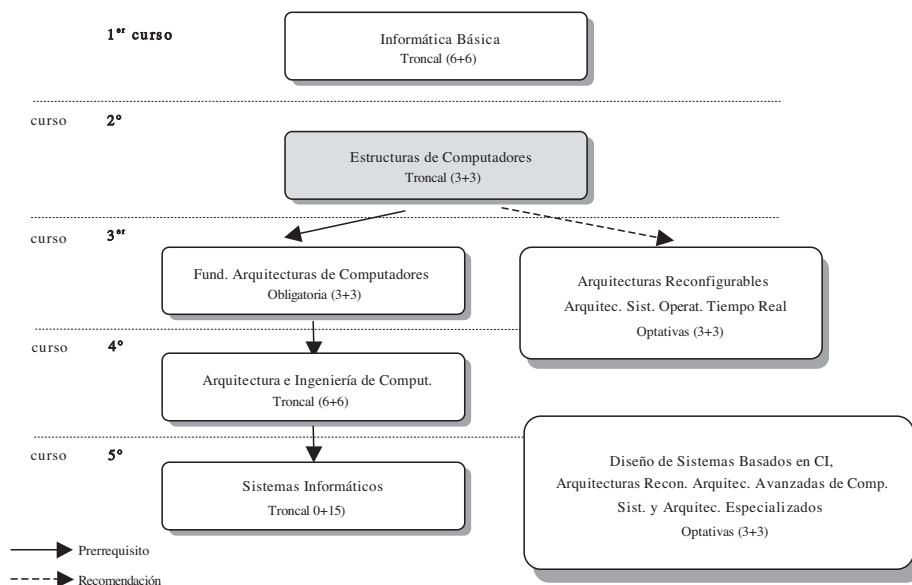


Ilustración 6. Relaciones de Estructuras de computadores en el plan de estudios

Analizando la figura, se aprecia que EC tiene como prerequisites Informática Básica. En esta asignatura se adquiere la base de diseño lógico y electrónica digital que sirve para emplearla tanto en el diseño de las estructuras, como para comprender la naturaleza de cada uno de los elementos que conforman el computador.

En cuanto a relaciones con asignaturas posteriores, en EC se adquiere una base sobre la organización del computador que será utilizada en las asignaturas Fundamentos de Arquitecturas de Computadores, Arquitectura e Ingeniería de Computadores y Sistemas Informáticos para profundizar en temas más complejos, centrándose en la mejora de los aspectos funcionales y en el rendimiento de los computadores. Finalmente, EC también proporciona una base de conocimientos importante para asignaturas optativas afines a arquitectura y estructuras de computadores.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos:

- OI1: Comprender e interpretar la organización interna de un computador, estructurándolo en sus unidades funcionales.
- OI2: Comprender e interpretar y las diferentes unidades funcionales del computador.
- OI3: Comprender el funcionamiento de la unidad de control así como las ventajas e inconvenientes de sus posibles métodos de diseño.
- OI4: Comprender el funcionamiento global de la ALU y las diferentes formas de implementar operadores y/o algoritmos.
- OI5: Conocer las características de los diferentes sistemas de almacenamiento que es posible encontrar en un computador y analizar la necesidad de establecer una organización jerárquica.
- OI6: Conocer los lenguajes de descripción hardware y su aplicación al diseño de procesadores.

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

A continuación se enumeran las competencias que, a partir de los objetivos desarrollados en el punto anterior, se considera que se deben adquirir en la asignatura Estructuras de Computadores.

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber): además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2 desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque 1: Lenguajes de descripción hardware

- CIC1: Conocer los diferentes tipos de datos en los lenguajes de descripción hardware.
- CIC2: Comprender las diferentes arquitecturas de descripción hardware que permiten los lenguajes HDL's.
- CIC3: Entender una descripción en lenguaje VHDL y comprender lo que está describiendo.
- CIC4: Conocer cómo se describen sistemas combinacionales con VHDL.
- CIC5: Conocer cómo se describen sistemas secuenciales con VHDL.
- CIC6: Comprender la metodología básica de diseño hardware, utilizando VHDL.

Bloque 2: Estructuras del Computador

- CIC7: Comprender las diferentes tecnologías en las que se basan las memorias.
- CIC8: Entender las diferentes formas de organizar las memorias internamente.
- CIC9: Comprender los cronogramas de los ciclos de lectura y escritura de las memorias.
- CIC10: Entender los mapas de memoria y su importancia en el diseño hardware de sistemas.
- CIC11: Comprender la jerarquía de las memorias.
- CIC12: Conocer cómo se describe en VHDL diferentes tipos de memorias.
- CIC13: Comprender la ruta de datos de un procesador.
- CIC14: Entender el diseño de rutas de datos.
- CIC15: Comprender los diferentes elementos que intervienen en una ruta de datos.
- CIC16: Comprender el diseño de rutas de datos de modo estructural utilizando VHDL.

- CIC17: Comprender el diseño de rutas de datos de modo algorítmico utilizando VHDL.
- CIC18: Entender las necesidades hardware de una ruta de datos.
- CIC19: Comprender la necesidad de la unidad de control en las rutas de datos.
- CIC20: Entender las unidades de control cableadas.
- CIC21: Entender las unidades de control microprogramadas.
- CIC22: Entender los cronogramas de las instrucciones.
- CIC23: Comprender la metodología de diseño de la unidad de control.
- CIC24: Comprender la metodología de diseño de un procesador.
- CIC25: Entender las necesidades y recursos hardware, que son necesarios para el diseño de procesadores de propósito general.
- CIC26: Entender las necesidades y recursos hardware, que son necesarios para el diseño de procesadores dedicados.

Destrezas tecnológicas (saber hacer): además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con fluidez los paquetes de software de diseño (ISE, Xilinx) y simulación (MaNoTaS) que servirán de guía para la realización de actividades relacionadas con la asignatura.

Destrezas lingüísticas (saber hacer): Además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje arquitectural, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso ya sea de descripción hardware como de diseño.

2.2.2 Competencias interpersonales (ser y estar): se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas: se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo: se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Conocer los diferentes sistemas de numeración y la conversión entre bases.
- Conocer las funciones lógicas y sus equivalentes mediante puertas.
- Saber operar en aritmética binaria sobre números enteros y en coma flotante.
- Conocer las tecnologías de los circuitos integrados.
- Conocer el diseño lógico combinacional a nivel de pequeña escala de integración.
- Conocer el diseño lógico secuencial.
- Comprender la arquitectura von Neumann y la relación entre las diferentes unidades funcionales de un computador.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerrequisitos necesarios para el estudio y entendimiento de esta asignatura se cubren en la asignatura Informática Básica. Además existen conceptos Físicos que se deben tener adquiridos de 2º de Bachillerato; sin embargo, no todos los alumnos han cursado la asignatura de Física y por lo tanto conviene realizar un repaso a las unidades de medida de tiempos y frecuencias.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Lenguajes de descripción hardware

- Tema 1: Introducción al VHDL.
- Tema 2: Fundamentos del lenguaje.
- Tema 3: Modelado Combinacional.
- Tema 4: Modelado Secuencial. Máquinas de estado finitos.

Bloque 2: Estructuras del Computador

- Tema 5: Unidad Aritmético-Lógica.
- Tema 6: Unidad de Control.
- Tema 7: Rutas de Datos.
- Tema 8: Unidad de Memoria.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1: Lenguaje de descripción hardware

- Tema 1: Introducción al VHDL.
 - 1.1 Historia.
 - 1.2 Características generales.
 - 1.3 Ventajas del VHDL para el diseño de sistemas.
- Tema 2: Fundamentos del lenguaje.
 - 2.1 Estructura de una descripción.
 - 2.2 Librerías.
 - 2.3 Entidades.
 - 2.4 Arquitecturas.
 - 2.5 Identificadores. Delimitadores.
 - 2.6 Operadores: Aritméticos, Relacionales, Lógicos y de Desplazamiento.
 - 2.7 Objetos: Constantes, Variables, Señales y Ficheros.
 - 2.8 Tipos de Datos y Subtipos.
 - 2.9 Procesos.
 - 2.10 Bloques.
 - 2.11 Sentencia when-else.
 - 2.12 Sentencia whit-select.
 - 2.13 Sentencias secuenciales.
- Tema 3: Modelado Combinacional.
 - 3.1 Multiplexores y Demultiplexores.
 - 3.2 Codificadores y Decodificadores.
 - 3.3 Comparadores.
- Tema 4: Modelado Secuencial. Máquinas de estado finitos.
 - 4.1 Contadores.
 - 4.2 Registros.
 - 4.3 Autómata de Moore y Mealy.

Bloque 2: Estructuras del Computador

- Tema 5: Unidad Aritmético-Lógica.
 - 5.1 Estructura general de la ALU.
 - 5.2 Operadores lógicos.
 - 5.3 Sumadores y restadores.
 - 5.4 Multiplicadores.
 - 5.5 Divisores.
 - 5.6 Otros operadores.
 - 5.7 Operaciones en coma flotante.

- Tema 6: Unidad de Control.
 - 6.1 Fases de ejecución de una instrucción.
 - 6.2 Unidad de Control cableada. Descripción en VHDL.
 - 6.3 Unidad de Control microprogramada.
- Tema 7: Rutas de Datos.
 - 7.1 MaNoTas. Ejemplo de ruta de datos.
 - 7.2 Elementos que intervienen en una ruta de datos.
 - 7.3 Diseño de rutas de datos
- Tema 8: Unidad de Memoria.
 - 8.1 Tecnologías de las memorias.
 - 8.2 Jerarquía de memoria.
 - 8.3 Memoria principal.
 - 8.4 Análisis de mapas de memoria.
 - 8.5 Diseño de mapas de memoria.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente se ha desarrollado en el artículo introductorio de este libro desde el punto de vista general para el segundo curso de la titulación de informática. Dicho desarrollo, aunque genérico, se considera válido para el caso particular de la asignatura Estructuras de Computadores. Resumiendo, podemos decir que aunque los nuevos modelos educativos propician el pensamiento creativo, enseñando a aprender por encima de enseñar conocimientos, no hay que menospreciar un modelo en el que la clase magistral tiene un papel importante pero no exclusivo en la transmisión de conocimientos, de forma que este tipo de enseñanza se va a complementar con otros procesos entre los que cabe destacar las prácticas de laboratorio y las actividades en grupos pequeños, que jugarán un papel fundamental. Las actividades que se proponen son:

- **Clases de teoría con apoyo de material audiovisual:** en lo que se refiere a las clases de teoría, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía sobre los contenidos más importantes de la asignatura. Además, algunos profesores de la asignatura hemos editado un libro que incluye parte de los contenidos de la misma. Dichas explicaciones teóricas se intercalarán con la realización de problemas, ejemplos prácticos y aplicaciones siempre que el contenido lo requiera. Se le recomienda al alumno que tome apuntes de las explicaciones del profesor, para de esta manera contrastar con otras fuentes, bibliografía recomendada y apuntes de otros compañeros.
- **Actividades en grupos pequeños / tutorías docentes:** estas actividades estarán relacionadas con la resolución de problemas y cuestiones teórico-

prácticas de la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos a situaciones reales concretas y fomentar la capacidad de análisis, síntesis y auto-evaluación del alumnado.

- **Prácticas de laboratorio:** en cuanto a las prácticas de laboratorio, cabe mencionar que éstas se apoyan en tutoriales disponibles para el alumnado que le pueden servir de guía para aprender a utilizar el correspondiente software de diseño y simulación, que les permite comprobar en todo momento sus diseños y, además, interpretar los resultados obtenidos.
- **Trabajos complementarios:** en cuanto a los trabajos complementarios, comentar que éstos incidirán en la nota final de la asignatura y pueden ser de índole teórica, de índole práctica o de índole teórico-práctica, y deberán realizarse de forma individual.
- **Tutorías de atención al alumnado:** el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como dudas concretas sobre el contenido de la asignatura. Además de dichas tutorías individualizadas, se programarán varias tutorías en grupo, al menos una para cada bloque de la asignatura correspondiente.

5.2 ESTRATEGIAS DE APRENDIZAJE

Las estrategias de aprendizaje se han establecido a nivel común en el artículo introductorio de este libro. Tal y como se indicó, los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador, no son los únicos medios sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web y el campus virtual ofrecen innumerables posibilidades que no hay que dejar pasar. El uso de la misma ha sido mayoritario en las experiencias llevadas a cabo hasta el momento.

Entre otras cosas, en dicha página podemos encontrar:

- **Tablón de anuncios:** desde aquí, el alumno puede estar perfectamente informado de cualquier novedad relacionada con la asignatura. Además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, etc.
- **Tutorías:** indica el horario de atención al alumnado.
- **Objetivos:** resume los objetivos que se pretenden alcanzar en esta asignatura.
- **Temario:** especifica el temario de esta asignatura.
- **Material docente de la asignatura:** aquí se puede encontrar todo el material que como mínimo va a ser necesario para el seguimiento de las clases teóricas y prácticas.
- **Bibliografía complementaria:** aparece un listado de bibliografía complementaria que pueden consultar para profundizar en la asignatura o preparar los trabajos complementarios.

- Plan de prácticas: se especifica semana a semana qué trabajo debe realizar el alumnado tanto en las horas presenciales como en las no presenciales de forma autónoma.
- Horarios de teoría y prácticas: contiene información sobre los grupos de teoría y prácticas, así como el profesorado que los imparte.
- Enlaces de interés: aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en algunos contenidos de la materia.
- Planificación de los tests de autoevaluación: en el campus virtual se publicará un ejercicio de autoevaluación por cada tema mediante el cual se podrá medir el grado de asimilación obtenido. En este apartado se recuerda el periodo en el que estará activo cada uno de los tests.

En base a todo esto, la estrategia de aprendizaje que se propone se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura.

Planificación de las clases teóricas:

- Lectura previa del guión correspondiente a la sesión de teoría que se trate.
- Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida en la propia aula entre todos o por el profesor fuera del aula.
- Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario se pedirá ayuda al profesor correspondiente.

Planificación de las clases prácticas:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.

- Parte de las prácticas se realizarán en los laboratorios y parte en horas no presenciales. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá con bastante celeridad dichas prácticas, indicando una vez corregidas los fallos más comunes. Cada estudiante, de forma individual, debe analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Auto-evaluación: una vez realizadas todas las actividades previas relacionadas con un tema concreto, el estudiante debe discernir si cree que dicho tema ha sido totalmente entendido. En caso de no ser así, debe incidir en el estudio de los contenidos que crea tener más flojos, utilizando si lo cree conveniente las tutorías y realizando algunos problemas de ampliación, bien de los propuestos en las hojas de problemas o bien haciendo uso de la bibliografía. Cuando crea estar preparado puede realizar el ejercicio de auto-evaluación del tema correspondiente, publicado en el campus virtual.

De forma opcional se podrá hacer trabajos individuales complementarios, para subir la nota, siempre y cuando el trabajo realizado a lo largo del curso se considere satisfactorio.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES				
ACTIVIDAD		Clases de teoría	Clases de prácticas	Actividades en grupos pequeños / tutorías docentes
Presentación:		1	0	1
BLOQUE 1	Tema 1	1	1	0
	Tema 2	2	2	1
	Tema 3	2	2	2
	Tema 4	2	2	2
BLOQUE 2	Tema 5	4	4	2
	Tema 6	4	4	2
	Tema 7	4	4	2
	Tema 8	4	4	2
Preparación del examen final:		3	2	4
Examen final:		3	2	0
TOTAL: 75		30	27	18

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES				
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños/ tutorías docentes
Presentación:		1,75	0	0,25
BLOQUE 1	Tema 1	1,75	2	0
	Tema 2	3,5	4	0,25
	Tema 3	3,5	4	0,5
	Tema 4	3,5	4	0,5
BLOQUE 2	Tema 5	7	8	0,5
	Tema 6	7	8	0,5
	Tema 7	7	8	0,5
	Tema 8	7	8	0,5
Preparación del examen final		5,25	4	1
Tutorías				
TOTAL: 101,75		47,25	50	4,5

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

Las horas no presenciales de la sesión de presentación estarán dedicadas a la recopilación de la documentación de la asignatura y, en su caso, al repaso de aquellos prerrequisitos que no se hayan alcanzado. La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y entendimiento de los enunciados de los problemas, y revisión de los problemas ya corregidos por el profesorado. De las horas no presenciales dedicadas a las clases de teoría se utiliza una por cada tema para la realización del ejercicio de auto-evaluación.

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

- Grediaga, M.L; Rico, A.; Soriano y Párraga, A. (2004) *Estructuras de computadores. Un computador ejemplo: MANOTAS*. Servicio de publicaciones de la Universidad de Alicante.
- Hennesy, J.L. y Patterson, D.A. (2002) *Arquitectura de computadores: un enfoque cuantitativo*. McGraw-Hill.
- Patterson, J.L.; Hennesy; Reverté (2000) *Estructura y diseño de computadores. Interficie, circuitería/programación*. D.A.
- Pellerin, D. and Taylor, D. (1997) *VHDL Made Easy*. Prentice Hall.
- Stallings, W. (2000) *Organización y arquitectura de computadores*. Prentice Hall.
- Villar, E. (1998) *VHDL, lenguaje estándar de diseño electrónico*. McGrawHill.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Angulo, J.M.; García, J. y Angulo, I. (2003) *Fundamentos y estructura de computadores*. Thomson.

Tanenbaum, A.S. (2000) *Organización de computadoras: Un enfoque estructurado*. Prentice Hall.

7.3 OTROS RECURSOS

Como se ha mencionado ya, en esta asignatura, el alumnado dispone de un guión de la asignatura y de un libro de texto, del que se ayudan en las clases de teoría. Además, el alumnado dispone en la red de todas las actividades que deben realizar. Pero además de estos recursos básicos, incluimos aquí otros recursos de especial interés, tanto para el profesorado como el alumnado.

Publicaciones periódicas:

- Integration, the VLSI journal
- IEEE Transactions on Computers
- IEEE Transactions on VLSI Systems
- IEE Proceedings - Computers and Digital Techniques
- IEE Proceedings - Circuits, Devices and Systems
- Journal of Circuits, Systems and Computers
- Circuits, Systems, and Signal Processing
- Journal of the ACM
- Journal of Systems Architecture
- The Journal of Instruction-Level Parallelism
- IPL (Information Processing Letters)
- Parallel Computing
- IEEE Concurrency
- Journal of Parallel and Distributed Computing
- International Journal of High Speed Computing
- The International Journal of Supercomputer Applications and High Performance Computing
- Parallel Algorithms and Applications

Asociaciones y grupos de interés:

- ANSI: American National Standards Institute. <http://www.ansi.org/>
- EAPLS: European Association for Programming Languages and Systems. <http://danae.unimuenster.de/eapls/>
- EUROMICRO: <http://www.euromicro.org/>
- IASTED: International Association of Science and Technology for Development. <http://www.iasted.com/>
- ISTECS: Iberoamerican Science and Technology Education Consortium: <http://www.istec.org/>

- ACM: Association for Computing Machinery. <http://www.acm.org>
- IEEE: Institute of electrical and Electronics Engineers. <http://www.ieee.org>
- IEEE Computer Society. <http://www.computer.org>
- ACM SIGCSE: Special Interest Group on Computer & Science Education. <http://www.acm.org/sigcse>

Páginas web de interés:

Algunas páginas Web muy interesantes, que proporcionan enlaces o recopilaciones de enlaces a recursos estadísticos (organizaciones, revistas electrónicas, conferencias, bibliografía, tutoriales, etc.) son las siguientes:

- <http://atc.ugr.es/~acanas/arquitectura.html><http://www.analyse-it.com/download/dl.asp> (enlaces de interés)
- <http://www.cs.wisc.edu/~arch/www/> (Web de AC de la Universidad de Wisconsin)
- <http://csjava.occ.cccd.edu/~pharao/CS116Links.html> (Diccionarios, software, etc sobre AC)
- <http://www.ee.iastate.edu/~acar/archlink.html> (lista de grupos de investigación sobre AC)
- <http://www.vhdl-online.de/tutorial/> (Tutorial de VHDL)
- http://www.ehu.es/Electronica_EUITI/vhdl/pagina/inicio.htm (Tutoria de VHDL)

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces es fácil la resolución de dudas de esta forma. No obstante, cuando ésta es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

8.1 PROCEDIMIENTOS DE EVALUACIÓN

El sistema de evaluación utilizado intenta determinar las competencias que posee el estudiante tras cursar la asignatura. En este tipo de evaluación se recoge información mediante un instrumento, prueba, procedimiento o actividad para poder describir e identificar las competencias adquiridas por los estudiantes acerca de un dominio de referencia, descrito en la planificación docente inicial. Así, los criterios de evaluación deben correlacionarse con los objetivos y especificar principalmente un dominio conceptual claro, la adquisición de procedimientos, técnicas habilidades y destrezas de ejecución profesional y académica, que concretan el nivel de capacidades personales y profesionales necesarias para el ejercicio profesional. Atendiendo a esto, la nota de la asignatura se desglosa de la siguiente forma:

NOTA DE ESTRUCTURA DE COMPUTADORES	
Examen de teoría	40%
Realización de las prácticas propuestas	25%
Examen de prácticas	10%
Test de auto-evaluación de Bloque 1	5%
Test de auto-evaluación de tema 5	5%
Test de auto-evaluación de tema 6	5%
Test de auto-evaluación de tema 7	5%
Test de auto-evaluación de tema 8	5%

Cuadro 4.: Desglose de la nota de la asignatura Estructura de Computadores

La asistencia a las clases de prácticas y la resolución de los ejercicios propuestos en grupos pequeños será obligatoria, valorándose la participación de los alumnos.

8.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas y procedimientos de diseño relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos de diseño y de resolución de problemas se ajustan a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre las estructuras del computador cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. Los problemas y procedimientos de diseño relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos de diseño y de resolución de problemas son generalmente ajustados a la naturaleza del problema. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de los resultados aceptables. • La actuación en las destrezas transferibles es generalmente buena. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.

Calificación	Rango	Pautas
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas y procedimientos de diseño relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio estándares son usualmente desarrolladas con éxito razonable, aunque el análisis de los resultados puede no ser entendido completamente. • Las destrezas transferibles están a un nivel básico. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspense	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas y procedimientos de diseño relacionados con la asignatura no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente y el significado y análisis de los resultados no son entendidos generalmente. • Las destrezas transferibles están a un nivel deficiente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 5. Criterios de evaluación

Para obtener la obtención de matrícula de honor, es necesario conseguir un sobresaliente y hacer un trabajo complementario.

8.3 EVALUACIÓN DEL PROCESO DOCENTE

La evaluación del proceso docente se llevará a cabo tanto por procesos internos como externos:

- Internamente, los profesores de la asignatura realizarán encuestas periódicas sobre los conocimientos adquiridos por el alumnado, así como sobre el funcionamiento general de la asignatura. En ellas se pondrán de manifiesto las dificultades de aprendizaje y de transmisión de conocimientos, experimentadas tanto por los alumnos como por los profesores.
- Externamente, se solicitará al Secretariado de Calidad de la Universidad de Alicante que realice su proceso de evaluación de docencia, cuyos resultados complementarán a los anteriores.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Estructuras de Computadores. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo de propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI1	De CIC4 a CIC6 cCIC1 cCIC2	1-4	7	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de diseño). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI2	De CIC13 a CIC19 cCIC1 cCIC2	1-4	5-8	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación y diseño). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI3	CIC20 a CIC23 cCIC1 cCIC2	1-4	6-7	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de simulación y diseño). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.

Cuadro 6. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI4	CIC5 y CIC6 cCIC1	3	5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de diseño y simulación). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI5	De CIC7 a CIC12 cCIC1		8	Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de diseño). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.
OI6	De CIC23 a CIC26 cCIC1	1-4	7	Enseñanza presencial (Lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestas y prácticas con el entorno de diseño). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Resolución de ejercicios. Examen final. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al objetivo.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)		
cOIP1	CIPTC1 cCIPTC1 cCIPTC2	1,4	7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3	1,4	5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4		6-8	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 8. Análisis de coherencia: competencias interpersonales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)		
cOS1	cCS1 cCS4 cCS5	1,4	7	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3	1,4	6	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Trabajo Complementario <u>Criterios:</u> Grado de análisis y evaluación de los conocimientos adquiridos
cOS3	cCS1 cCS2 cCS4 cCS5	1,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2	1,4	6-8	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos estadísticos aplicados.

Cuadro 9. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE HERRAMIENTAS DE PROGRAMACIÓN

*Antonio-M. Corbí Bellot; Alicia Garrido Alenda; Ramón Rubio Serna;
José Oncina Carratalá*

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
(acorbi; alicia; rrubio; oncina@dlsi.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

La Association for Computing Machinery (ACM) es una de las dos asociaciones profesionales internacionales más importantes en el campo de la informática. La otra es la sección *Computer Society* del Institute for Electric and Electronic Engineers, la IEEE-CS.

En 1989, una comisión de la ACM publicó en la revista *Communications of the ACM* un resumen de un informe elaborado en el que se proponía la definición de la computación como una disciplina, definiendo y describiéndolo en subáreas con la intención de servir de base para el diseño de currícula. El informe define nueve subáreas:

- algoritmos y estructuras de datos
- lenguajes de programación
- arquitectura
- computación numérica y simbólica
- sistemas operativos
- metodología e ingeniería de la programación
- sistemas de bases de datos y recuperación de datos
- inteligencia artificial y robótica
- comunicación ser humano-ordenador

Además, diferencia en cada una de ellas tres paradigmas: teoría, abstracción y diseño.

Los conceptos que abarca la asignatura de Herramientas de Programación se sitúan en las subáreas de lenguajes de programación, sistemas operativos y metodología e ingeniería de la programación.

Posteriormente en 1990 un comité conjunto de la ACM-IEEE publicó el informe Computing Curricula 1991, en el que se utiliza la misma división anterior en subáreas, pero subdivide las áreas en unidades de conocimiento más concretas que se utilizan en la composición de las asignaturas comunes de los currícula que se proponen al final del informe.

En el nuevo documento Computing Curricula 2001 de la IEEE-CS y de la ACM sobre los estudios de informática, aparecen 14 subáreas de las que la asignatura Herramientas de Programación tiene elementos en común con las siguientes:

Fundamentos de programación

- PF1: fundamental programming constructs
- PF2: fundamental data structures

Arquitectura y organización

- AR2: machine level representation of data
- AR4: memory system organization and architecture
- AR5: interfacing and communication
- AR8: performance enhancements

Sistemas operativos

- OS5: memory management
- OS11: system performance evaluation
- OS12: scripting

Aspectos sociales y profesionales

- SP1: history of computing
- SP3: methods and tools of analysis
- SP4: professional and ethical responsibilities

Ingeniería del software

- SE1: software design
- SE2: using APIs
- SE3: software tools and environments
- SE5: software requirements and specifications
- SE6: software validation
- SE7: software evolution
- SE8: software project management
- SE11: software reliability

Por otra parte, si nos situamos en el contexto español, debemos observar primero cuáles son las directrices generales propias de las titulaciones de Informática (Reales decretos 1459/1990, 1460/1990, 1461/1990 de 26 de octubre, BOE 1990).

Siguiendo las directrices generales propias, publicadas en el B.O.E. de 20-11-90, se aprobó el nuevo plan de estudios para las tres titulaciones de Informática de la Universidad de Alicante B.O.E. del 25-9-2001, puesto en marcha en el curso 2001-2002.

En dicho plan de estudios, y en las titulaciones de Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas, aparece como materia optativa mientras que lo hace como obligatoria en la titulación de Ingeniería Informática. En las tres titulaciones tiene un total de 6 créditos, 3 teóricos y 3 prácticos.

No quedaría completa la contextualización de la asignatura Herramientas de Programación sin observar su relación con los perfiles profesionales generales propuestos en el *Career Space*:

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: para el diseño de software es esencial saber emplear las distintas herramientas que tiene disponible el programador, así como saber detectar y corregir los errores que surgen al ejecutar una aplicación.
- Arquitectura y diseño de software: en el diseño de software intervienen los distintos componentes del equipo de desarrollo, los cuales empleando las herramientas oportunas podrán comunicarse más fácilmente.
- Diseño multimedia: las aplicaciones multimedia requieren de un uso adecuado de la memoria así como de un código optimizado para obtener el mejor rendimiento.

Perfil profesional de sistemas:

- Ingeniería de comunicación de datos: los conocimientos adquiridos en la asignatura capacitan al futuro titulado para llevar a cabo la implantación y gestión de la información dentro del equipo de desarrolladores.
- Diseño de redes de comunicación: por motivos similares a los anteriores, el estudiante de esta asignatura es capaz de crear e implantar las redes de comunicación necesarias para el desarrollo, distribución y mantenimiento de sus productos.
- Asistencia técnica: en este perfil se debe tener conocimientos sobre detección de fallos de una aplicación así como de las posibles soluciones que se les pueden dar.
- Ingeniería de integración y pruebas e implantación y pruebas: en la realización de pruebas, es conveniente conocer un marco de desarrollo y paso de tests, de manera que se valide el producto desarrollado.

Perfil profesional de gestión y explotación de tecnologías de la información:

- Consultoría de empresas de TI: las empresas de *Tecnologías de la Información* requieren de profesionales que sepan cómo funciona una aplicación por dentro y por tanto cuáles son los errores que se pueden producir para poder subsanarlos.
- Especialista en sistemas: en dicho perfil se requiere conocer el funcionamiento de estos sistemas para emplear las herramientas necesarias para el desarrollo y/o adaptación de las aplicaciones a los mismos.
- Desarrollo de investigación y tecnología: la participación y gestión de proyectos ya sean de pequeña envergadura o proyectos de investigación internacionales se puede ver beneficiada por el uso de herramientas que faciliten la comunicación entre los integrantes del equipo, por ejemplo para realizar control de versiones, para generar documentación, para formatear el código fuente, etc.
- Dirección de TIC (marketing, proyectos, dirección general): en este perfil es importante un conocimiento amplio del resto de áreas tecnológicas de las Tecnologías de la Información y la Comunicaciones (TIC) y por tanto, es un buen punto de apoyo el conocer aquellas herramientas que nos pueden ayudar en ese sentido.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	Conocimiento y habilidad para emplear las diversas herramientas que intervienen en el desarrollo de software y en la detección y resolución de problemas que pueden aparecer. Conocimiento y habilidad para comunicarse con los compañeros de trabajo.
Arquitectura y diseño de software	Conocimiento en el diseño de aplicaciones para comprender su funcionamiento. Conocimiento y habilidad para desarrollar una aplicación fácilmente internacionalizable.
Diseño multimedia	Conocimiento en el desarrollo de aplicaciones optimizadas y con una gestión de la memoria que emplean.
Ingeniería de comunicación de datos	Conocimiento y habilidad para implantar y usar mecanismos de compartición de información entre el grupo de desarrolladores
Diseño de redes de comunicación	Conocimiento y habilidad desarrollar productos por parte de grupos de desarrolladores distribuidos por todo el mundo.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Asistencia técnica	Conocimiento y habilidad en la detección y corrección de errores producidos por la ejecución de una aplicación. Conocimiento y habilidad para optimizar la ejecución de aplicaciones
Ingeniería de integración y pruebas e implantación y pruebas	Conocimiento y habilidad para el desarrollo de tests. Conocimiento y habilidad para configurar una aplicación antes de compilarla. Conocimiento y habilidad para escribir y generar documentación.
Consultoría en empresas de TI	Conocimiento y habilidad en herramientas que permitan generar de forma más rápida y eficiente aplicaciones optimizadas.
Especialista en sistemas	Conocimiento y habilidad en los sistemas donde se va a desarrollar y/o instalar una aplicación.
Desarrollo de investigación y tecnología	Conocimiento y habilidad en el uso de herramientas que permitan la comunicación entre los investigadores de un proyecto.
Dirección de TIC	Conocimiento y habilidad de las herramientas necesarias para el desarrollo, detección de errores, generación de documentación, internacionalización, etc., que hacen que un entorno de desarrollo sea altamente productivo y competitivo.

Cuadro 1. *Relación de la asignatura con los perfiles profesionales*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Herramientas de Programación se imparte en el segundo curso de Ingeniería Informática como asignatura obligatoria y también en el segundo curso de la Ingeniería Técnica en Informática de Sistemas y de la Ingeniería Técnica en Informática de Gestión como asignatura optativa.

La finalidad de esta asignatura es dotar a los Ingenieros en Informática de los conocimientos necesarios para desarrollar de manera correcta aplicaciones a gran escala, bien porque son ellos los programadores, bien porque realizan labores de coordinación de un grupo de programadores.

En entornos de programación profesionales existe toda una ‘cultura de la programación’ debida, entre otras cosas, al uso de una serie de herramientas y a una metodología de trabajo que fuera de ellos no se suele conocer y, por tanto, no se emplea en la práctica.

Nos encontramos así con empresas de desarrollo de software cuyo funcionamiento es totalmente anárquico en este sentido –nula o mínima coordinación

entre sus programadores, deficiente gestión de las versiones de sus productos, desconocimiento de las capacidades de las herramientas empleadas, etc...

Con esta asignatura pretendemos difundir toda la información necesaria para que nuestros alumnos puedan realizar las labores relacionadas con la programación de la manera más apropiada y eficiente posible.

Algunas de las aplicaciones inmediatas que tienen los conceptos estudiados en la asignatura son:

- Detección de errores de ejecución debidos al mal uso de memoria dinámica.
- Mejor aprovechamiento de herramientas básicas para el programador como pueden ser los compiladores o editores.
- Detección de ‘cuellos de botella’ en las aplicaciones.
- Conocer mecanismos y herramientas que permiten el trabajo en grupo de una forma más sencilla.
- Conocer aspectos básicos de los sistemas de control de versiones.
- Aprender a depurar una aplicación a nivel de código fuente.
- Aprender a internacionalizar, de manera sencilla, los mensajes que una aplicación presenta al usuario.
- Conocer los distintos tipos de licencias que se pueden aplicar al software desarrollado.
- Aprender a construir aplicaciones extensibles. Dar a conocer los lenguajes de extensión.
- Conocer toda una serie de herramientas para la gestión y tratamiento de ficheros tanto de texto como binarios.
- Mostrar una serie de pautas/normas a tener en cuenta para el desarrollo de aplicaciones a gran escala y para el trabajo en grupo (estándares de nomenclatura de identificadores, estándares de uso de opciones de llamada a la aplicación, indentado y comentarios del código fuente, así como, para cada proyecto, la distribución de su código en un árbol de directorios).
- Aprender a crear y pasar ‘tests’ para comprobar el funcionamiento correcto de una aplicación de manera automática.

Dicha asignatura tiene una relación directa -como prerrequisitos de ella- con varias asignaturas de primer curso donde se imparten conceptos relacionados con la *programación*, si bien en el Plan de Estudios actual no es *prerrequisito* ni *recomendación* de ninguna asignatura, creemos que debería serlo de cualquiera que emplee la *programación* tanto en su parte teórica como práctica. De hecho nos encontramos con alumnos no matriculados en ella que requieren conocimientos de ésta para desarrollar las prácticas de otras asignaturas:

Ya en segundo y posteriores, cursos existen otras asignaturas que bien para comprender mejor sus contenidos teóricos, bien para llevar a cabo sus prácticas requieren del conocimiento de los temas impartidos en Herramientas de Programación:

- **Aplicaciones Industriales del Reconocimiento Automático:** asignatura optativa, cuyos descriptores son: técnicas de reconocimiento de formas y sus aplicaciones.
- **Lenguajes, Gramáticas y Autómatas:** es una asignatura troncal de segundo curso de la titulación de Ingeniería Informática. Sus descriptores son: máquinas secuenciales y autómatas finitos, gramáticas y lenguajes formales, y redes neuronales.
- **Algoritmia Avanzada:** asignatura obligatoria de cuarto curso para Ingeniería Informática. Sus descriptores son: búsqueda exhaustiva y estocástica, programación dinámica y algoritmos de codificación y compresión.
- **Programación Orientada a Objetos:** asignatura obligatoria de segundo curso para Ingeniería Informática. Sus descriptores son: Características de la POO. Clases y objetos. diseño orientado a objetos. Lenguajes de programación orientados a objetos. Objetos distribuidos. Herencia y genericidad. Persistencia en un entorno orientado a objetos.
- **Sistemas Operativos I:** asignatura troncal de segundo curso para Ingeniería Informática. Sus descriptores son: Organización, estructura y servicio de los sistemas operativos. Gestión y administración de memoria y de procesos. Gestión de Entrada/Salida. Sistemas de ficheros. Modelos.
- **Informática Musical:** asignatura optativa para Ingeniería Informática. Sus descriptores son: Sonido analógico y sonido digital. Procesamiento de señales musicales. Síntesis de sonido. Secuenciación y control. Composición asistida y análisis musical.
- **Programación en Entornos Interactivos:** asignatura optativa para Ingeniería Informática. Sus descriptores son: Programación visual. Programación dirigida por eventos. Interfaces gráficos de usuario.
- **Programación Concurrente:** asignatura optativa para Ingeniería Informática. Sus descriptores son: Procesos. Sincronización, competencia y cooperación. Exclusión mutua. Memoria compartida. Memoria distribuida. Csp.
- **Programación y Estructuras de Datos:** asignatura troncal de segundo curso para Ingeniería Informática. Sus descriptores son: Estructuras de datos y algoritmos de manipulación. Tipos abstractos de datos. diseño recursivo.
- **Gráficos por Computador:** asignatura obligatoria de tercer curso para Ingeniería Informática. Sus descriptores son: Transformaciones 2D y 3D. Proyecciones y vistas. Visualización.
- **Diseño y Programación Avanzada de Aplicaciones:** asignatura obligatoria de tercer curso para Ingeniería Informática. Sus descriptores son: Aplicaciones distribuidas. Aplicaciones internet. Sistemas abiertos. Objetos distribuidos. Cliente/Servidor.
- **Diseño y Análisis de Algoritmos:** asignatura troncal de tercer curso para Ingeniería Informática. Sus descriptores son: diseño de programas: des-

composición modular y documentación. Técnicas de verificación y pruebas de programas. La eficiencia de los algoritmos. Divide y vencerás. Algoritmos voraces. Algoritmos con retroceso.

- **Sistemas Operativos II:** asignatura troncal de tercer curso para Ingeniería Informática. Sus descriptores son: Módulos. Gestión y administración de memoria y de procesos. Gestión de entrada/salida. Sistemas de ficheros. Seguridad. diseño.
- **Procesadores de Lenguaje:** asignatura troncal de cuarto curso para Ingeniería Informática. Sus descriptores son: Compiladores. Traductores e intérpretes. Fases de compilación. Optimización de código. Macroprocesadores.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES.

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos:

- **OI1:** Comprender y discernir las actividades de preproceso, compilación y enlace.
- **OI2:** Comprender lo que significa la depuración de una aplicación a nivel de código fuente y ser capaz de emplear un depurador para realizar esta tarea.
- **OI3:** Comprender los problemas y errores que pueden aparecer por un uso erróneo de la memoria dinámica y conocer mediante qué herramientas se pueden detectar más fácilmente.
- **OI4:** Saber representar dependencias temporales entre archivos para optimizar los tiempos de compilación.
- **OI5:** Conocer la gran variedad de herramientas existentes para procesar de manera 'off-line' la información contenida en archivos de texto y binarios.
- **OI6:** Conocer cómo funciona un intérprete de órdenes y saber realizar pequeños guiones para automatizar tareas.
- **OI7:** Conocer las características necesarias que debe tener un editor de textos para ser apto para la tarea de programar.
- **OI8:** Conocer toda una serie de estándares relacionados con la programación, tales como formato de los comentarios, distribución del código fuente, indentado del código fuente, etc...
- **OI9:** Conocer los conceptos básicos del control de versiones. Saber utilizar al menos una aplicación de control de versiones centralizada.
- **OI10:** Conocer los conceptos básicos de la *programación por contrato*.
- **OI11:** Saber detectar los '*cuellos de botella*' de una aplicación. Saber detectar la completitud del conjunto de datos de entrada de una aplicación.

OI12: Conocer los conceptos de *internacionalización* y *localización*. Saber preparar el código fuente de una aplicación para internacionalizarla.

2.1.2 Objetivos interpersonales generales (ser y estar).

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, éstos son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales.

Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, éstos son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro. Además, en la asignatura Herramientas de Programación, planteamos los siguientes:

- **OS1:** Aprender a desarrollar trabajo en grupo, usando para ello las herramientas estudiadas.
- **OS2:** Comprender el funcionamiento de las aplicaciones para entender más fácilmente por qué fallan en ocasiones.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer).

Dentro de las competencias instrumentales, distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Vamos a desarrollar estas competencias para cada uno de los temas de la asignatura.

Habilidades cognitivas (saber).

Además de las habilidades cognitivas cCIC1 y cCIC2, vistas en el artículo introductorio de este libro, consideramos las siguientes:

Bloque 1: Interfaz con el sistema operativo.

- Tema 1: El intérprete de órdenes: bash.
 - CIC1: Conocer la herramienta fundamental para comunicarnos con el sistema operativo: el intérprete de órdenes.
 - CIC2: Conocer como automatizar tareas con el intérprete de órdenes.
- Tema 2: Utilidades para el proceso de ficheros: de texto, binarios y en general.
 - CIC3: Conocer las herramientas básicas para el proceso de archivos de texto.

CIC4: Conocer las herramientas básicas para el proceso de archivos binarios.

CIC5: Conocer un conjunto de herramientas para el proceso de archivos, independientemente de que estos sean de texto o binarios.

- Tema 3: Editores.

CIC6: Conocer diversos editores orientados a la tarea de programar.

CIC7: Conocer los criterios básicos para decidir si un editor es apto para la tarea de programar.

Bloque 2: Compilación.

- Tema 4: Estándares.

CIC8: Conocer una práctica estándar de distribución del código fuente de un proyecto.

CIC9: Conocer una práctica estándar para incluir comentarios en el código fuente.

CIC10: Conocer una práctica estándar para escribir el código fuente.

- Tema 5: Compilación y enlace.

CIC11: Conocer las distintas fases de lo que llamamos compilación.

CIC12: Conocer las distintas opciones que admiten las herramientas que forman parte del proceso de compilación y cómo afectan estas al resultado obtenido.

- Tema 6: Make.

CIC13: Saber representar dependencias temporales entre archivos.

CIC14: Saber optimizar el proceso de compilación del código fuente de un proyecto.

- Tema 7: Optimización de código.

CIC15: Saber detectar ‘cuellos de botella’ en una aplicación.

CIC16: Conocer lo que es un perfil de ejecución de una aplicación y un test de cobertura.

Bloque 3: Detección y resolución de problemas.

- Tema 8: Resolución de problemas con memoria dinámica.

CIC17: Saber encontrar y eliminar errores debidos al mal uso de memoria dinámica.

- Tema 9: Programación por contrato.

CIC18: Saber cómo utilizar los mecanismos de verificación formal en la práctica.

- Tema 10: Depuración a nivel de código fuente.

CIC19: Conocer las herramientas de depuración a nivel de código fuente.

CIC20: Conocer los requisitos para poder depurar una aplicación a nivel de código fuente.

CIC21: Conocer las operaciones básicas que permite realizar un depurador.

Bloque 4: Herramientas adicionales.

- Tema 11: Control de versiones.
 - CIC22: Saber en qué consiste el ‘control de versiones’ a nivel de desarrollo de un proyecto software.
 - CIC23: Conocer la terminología básica empleada en el control de versiones.
 - CIC24: Conocer las operaciones básicas empleadas en el control de versiones.
- Tema 12: Internacionalización de aplicaciones.
 - CIC25: Conocer el concepto de ‘internacionalización’ –i18n–.
 - CIC26: Conocer cómo separar código de mensajes emitidos por una aplicación.
 - CIC27: Conocer la idea de lo que es un catálogo de mensajes.

Capacidades metodológicas (saber hacer).

Las capacidades metodológicas se han agrupado por los temas de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3, incluidas a nivel general en el artículo introductorio de este libro.

Bloque 1: Interfaz con el sistema operativo.

- Tema 1: El intérprete de órdenes: bash.
 - CIM1: Saber ejecutar un intérprete de órdenes en modo ‘login’ y en modo normal. Saber configurar los archivos necesarios para automatizar su puesta en marcha/finalización.
 - CIM2: Saber construir guiones o programas para el intérprete de órdenes.
- Tema 2: Utilidades para el proceso de ficheros: de texto, binarios y en general.
 - CIM3: Saber realizar el proceso de archivos de texto con las herramientas oportunas.
 - CIM4: Saber realizar el proceso de archivos binarios con las herramientas oportunas.
 - CIM5: Saber realizar el proceso de archivos con las herramientas oportunas, independientemente de que estos sean de texto o binarios.
- Tema 3: Editores.
 - CIM6: Ser capaz de emplear diversos editores orientados a la tarea de programar, conociendo sus puntos fuertes y débiles. Ser capaz de adaptarlos al uso de cada uno.
 - CIM7: Ser capaz de evaluar si un editor es apto para la tarea de programar en función de sus opciones de configuración.

Bloque 2: Compilación.

- Tema 4: Estándares.
 - CIM8: Adquirir destreza en la distribución del código fuente de un proyecto.
 - CIM9: Adquirir destreza en el uso de comentarios en el código fuente.
 - CIM10: Adquirir destreza en la escritura del código fuente respetando un estándar (nombres de archivos, formato de identificadores, etc...).
- Tema 5: Compilación y enlace.
 - CIM11: Aprender a usar y distinguir las distintas fases de lo que llamamos compilación.
 - CIM12: Aprender a usar las distintas opciones que admiten las herramientas que forman parte del proceso de compilación y comprobar de qué manera afectan estas al resultado obtenido.
- Tema 6: Make.
 - CIM13: Ser capaz de representar dependencias temporales entre archivos. Ser capaz de escribir archivos 'Makefile' correctos.
 - CIM14: Ser capaz de escribir archivos 'Makefile', que optimicen el proceso de compilación del código fuente.
- Tema 7: Optimización de código.
 - CIM15: Familiarizarse con las operaciones básicas que permiten detectar 'cuellos de botella' en una aplicación.
 - CIM16: Ser capaz de obtener un perfil de ejecución de y un test de cobertura de una aplicación.

Bloque 3: Detección y resolución de problemas.

- Tema 8: Resolución de problemas con memoria dinámica.
 - CIM17: Adquirir destreza a la hora de encontrar y eliminar errores debidos al mal uso de memoria dinámica. Ser capaz de usar las distintas herramientas vistas en teoría y de decidir cuando usar una u otra dependiendo del caso.
- Tema 9: Programación por contrato.
 - CIM18: Ser capaz de utilizar los mecanismos de verificación formal en la práctica. Ser capaz de detectar qué parte del 'contrato' es la incumplida.
- Tema 10: Depuración a nivel de código fuente.
 - CIM19: Ser capaz de emplear las herramientas de depuración a nivel de código fuente.
 - CIM20: Aprender a cumplir los requisitos para poder depurar una aplicación a nivel de código fuente.
 - CIM21: Conocer y ser capaz de emplear las característica más comunes de un depurador para depurar una aplicación a nivel de código fuente.

Bloque 4: Herramientas adicionales.

- Tema 11: Control de versiones.
 CIM22:Familiarizarse con el llamado ‘control de versiones’ a nivel de desarrollo de un proyecto software.
 CIM23:Adquirir destreza en el uso de la terminología básica empleada en el control de versiones.
 CIM24:Ser capaz de utilizar las operaciones básicas del ‘control de versiones’.
- Tema 12: Internacionalización de aplicaciones.
 CIM25:Familiarizarse con el concepto de ‘internacionalización’ –i18n–.
 CIM26:Aprender cómo se puede separar código de mensajes emitidos por una aplicación.
 CIM27:Ser capaz de generar y procesar catálogos de mensajes.

Destrezas tecnológicas (saber hacer).

Además de la destreza tecnológica cCIT1, introducida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con soltura las herramientas que permiten resolver los distintos tipos de problemas que se estudian en la asignatura.
- CIT2: Utilizar con soltura herramientas específicas de búsqueda de información de las herramientas empleadas.
- CIT2: Saber combinar diversas herramientas para solucionar a un problema.

Destrezas lingüísticas.

Además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir un buen conocimiento del lenguaje y los términos empleados en determinados temas de la asignatura.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas:

Las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2

Además añadimos la siguiente competencia particular:

- CIPTC1: Ser capaz de combinar los conocimientos y las destrezas obtenidas en la asignatura para, tras el análisis de un problema, saber plantear su

solución en términos de lo estudiado. En determinados casos este proceso requiere la comunicación y coordinación con el resto de compañeros del grupo de trabajo.

Compromiso con el trabajo:

Las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Conocer el uso de un intérprete de comandos.
- Conocer y saber usar herramientas básicas para el tratamiento de ficheros.
- Conocer las características básicas de un editores de texto para programadores
- Conocer el uso del compilador y enlazador.
- Saber usar un depurador a nivel de código fuente así como saber depurar de distintos tipos de procesos.
- Conocer cómo llevar a cabo la gestión de proyectos con “make”.
- Conocer y saber aplicar los principios de la programación por contrato.
- Saber detectar y corregir los problemas derivados del mal uso de la memoria dinámica.
- Saber detectar cuellos de botella y optimizar la ejecución de aplicaciones.
- Saber utilizar un sistema de control de versiones.
- Saber internacionalizar una aplicación.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

El alumnado de esta asignatura es heterogéneo ya que Herramientas de Programación es una asignatura obligatoria en Ingeniería Informática y optativa en Ingeniería Técnica en Informática de Gestión y en Ingeniería Técnica en Informática de Sistemas.

Con la idea de paliar esta diferencia en la procedencia de nuestros alumnos y, a la vez, intentando hacer más agradable la asignatura, tratamos de dar un

enfoque totalmente práctico a las clases, así como a las evaluaciones que hacemos de la misma. Para ello realizamos dos evaluaciones al alumnado, una no presencial y una presencial; en ambas el equivalente al examen a realizar se lleva a cabo sobre un computador y disponiendo de toda la bibliografía que el alumno considere que puede necesitar.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Interfaz con el sistema operativo.

- Tema 1: El intérprete de órdenes: bash.
- Tema 2: Utilidades para el proceso de ficheros: de texto, binarios y en general.
- Tema 3: Editores.

Bloque 2: Compilación.

- Tema 4: Estándares.
- Tema 5: Compilación y enlace.
- Tema 6: Make.
- Tema 7: Optimización de código.

Bloque 3: Detección y resolución de problemas.

- Tema 8: Resolución de problemas con memoria dinámica.
- Tema 9: Programación por contrato.
- Tema 10: Depuración a nivel de código fuente.

Bloque 4: Herramientas adicionales.

- Tema 11: Control de versiones.
- Tema 12: Internacionalización de aplicaciones.

Temas o unidades de contenido. Desarrollo

Bloque 1: Interfaz con el sistema operativo.

- Tema 1: El intérprete de órdenes: bash.
 1. Tipos de *shells*.
 2. Palabras reservadas.
 3. Redirección de entrada salida.
 4. Control de procesos y recursos.
 5. Guiones de *shell*.
- Tema 2: Utilidades para el proceso de ficheros: de texto, binarios y en general.
 1. Ficheros de Texto:

- Utilidades generales (búsqueda de patrones, concatenación de archivos, ordenación, unión, etc...).
 - Editor orientado a “streams”: Sed.
 - Preprocesadores: M4.
 - Indentado de código fuente: indent.
2. Ficheros de Binarios:
 - Creación y manipulación de bibliotecas: ar, ranlib.
 - Información de archivos objeto: nm.
 - Manipulación de archivos objeto: strip.
 - Estudio de la ejecución de un programa: strace, ltrace.
 3. Ficheros de Texto y Binarios:
 - Creación de paquetes: tar.
 - Compresión de paquetes: compress, gzip, bzip2.
 - Búsqueda de archivos: find.
- Tema 3: Editores.
 1. Características deseables de un editor de texto para programadores.
 2. Un ejemplo: vim.
 - Configuración.
 - Uso.
 3. Un ejemplo: emacs.
 - Configuración.
 - Uso.

Bloque 2: Compilación.

- Tema 4: Estándares.
 1. Estructura de un proyecto software.
 2. Formato de ficheros de cabecera y código fuente.
 3. Declaración de macros, variables globales y funciones.
 4. Uso correcto de comentarios.
 5. Formato del código fuente, de ficheros “Makefile”, de números de versión y de opciones de llamada a una aplicación.
- Tema 5: Compilación y enlace.
 1. Familia de compiladores con licencia GNU: gcc.
 2. Opciones para el control de la salida.
 3. Opciones para el control de la generación de advertencias, la generación de información extra para depuración y para optimización.
 4. Opciones relativas a la selección y optimización para una arquitectura concreta.
 5. Opciones relativas al preprocesador.
 6. Enlazador y opciones relativas al enlazador.
 7. Compilación condicional.

- Tema 6: Make.
 1. Uso de “make”.
 2. Formato de los ficheros “Makefile”.
 3. Uso de variables y de reglas implícitas.
 4. Uso de “comodines”.
 5. Uso de “include”.
 6. Objetivos “estándar”.
 7. Herramientas similares a “make”.
- Tema 7: Optimización de código.
 1. Generadores de perfiles de ejecución: gprof.
 - Tipos de perfiles: plano y grafo de llamadas.
 - Uso de gprof. ¿Cuándo usarlo?, ¿cómo usarlo?
 2. Generadores de tests de cobertura: gcov.
 - ¿Qué es un test de cobertura? ¿Qué finalidad tiene?
 - Uso de gcov. ¿Cuándo usarlo?, ¿cómo usarlo?

Bloque 3: Detección y resolución de problemas.

- Tema 8: Resolución de problemas con memoria dinámica.
 1. Tipos de problemas que surgen.
 2. Soluciones propuestas: electric-fence, dmalloc, ccmalloc y valgrind.
¿Cuándo y cómo emplear cada una de ellas?
- Tema 9: Programación por contrato.
 1. Base teórica de “programación por contrato”: la Verificación Formal de Programas.
 2. Uso de asertos. Limitaciones.
 3. Uso de Nana.
- Tema 10: Depuración a nivel de código fuente.
 1. ¿Qué es un depurador a nivel código fuente?
 2. Consejos útiles para no tener que usar un depurador a nivel código fuente.
 3. Cuidado especial con los errores no reproducibles.
 4. El depurador gdb. Características especiales.
 5. Uso de gdb. “Breakpoints” y “Watchpoints”.
 6. Trabajando con la *pila de llamadas*.
 7. Depuración de procesos en ejecución y de procesos ya “muertos”/terminados anómalamente.
 8. Frontales sobre gdb.

Bloque 4: Herramientas adicionales.

- Tema 11: Control de versiones.
 1. ¿Qué es el control de versiones?
 2. Herramientas de control de versiones distribuidas y centralizadas.

3. Conceptos generales del control de versiones en cvs. Cvs en un día.
 4. Otras órdenes de cvs menos empleadas.
- Tema 12: Internacionalización de aplicaciones.
 1. Internacionalización (I18N) vs. Localización (L10N).
 2. Una solución práctica: gettext.
 3. Uso de gettext. Catálogos de mensajes.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente empleada está fuertemente basada en la clase magistral, pero teniendo en cuenta que estamos ante una asignatura eminentemente práctica. Es por eso que intentaremos que el alumno aprenda a usar las herramientas empleadas de forma combinada en lugar de *solo* enumerarle las herramientas existentes y cuál es su función aislada respecto a otras.

Para ello complementaremos las clases magistrales de la siguiente manera:

- Clases de teoría con apoyo de material audiovisual: las clases de teoría se apoyan del material audiovisual disponible para el alumno (tanto en castellano como en valenciano) en: <<http://www.dlsi.ua.es/asignaturas/hp/material.html>>. Además se podrá ir probando en clase todos los conocimientos transmitidos debido al uso de un computador portátil por parte del profesor.
- Prácticas en laboratorios: constan de una serie de ejercicios resueltos y de otros sin resolver, de manera que el alumno pueda probar y comprobar los primeros de forma que le permitan adquirir y asentar los conocimientos necesarios para encontrar la solución a los segundos.
- Trabajos complementarios: se pueden considerar como ejercicios prácticos de un nivel superior y que engloban materias de diferentes temas que se han de poner en común para resolverlos.
- Tutorías de atención al alumnado: el alumno tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización, planificación y contenidos de los temas de la asignatura. No olvidemos en este punto la posibilidad de hacer uso de las tutorías a través del Campus Virtual e, incluso, del correo electrónico. Estas tutorías individualizadas, se complementarán con varias tutorías en grupo, al menos una para cada bloque de la asignatura correspondiente.
- Actividades en grupos pequeños / tutorías docentes: Estas actividades tienen que ver con la realización de problemas y cuestiones teórico-prácticas relacionadas con la asignatura surgidas bien en las clases de teoría, bien en las de prácticas. De esta forma pretendemos reforzar y aplicar los conceptos básicos a situaciones reales concretas, así como fomentar la capacidad de análisis, síntesis y auto-evaluación del alumnado frente a problemas nuevos.

5.2 ESTRATEGIAS DE APRENDIZAJE

Como ya se ha indicado, las clases de teoría de la asignatura serán impartidas con ayuda de un computador personal, que permitirá al profesor ir exponiendo los contenidos de la asignatura de forma gradual y al mismo tiempo ir comprobando en la máquina aquello de lo que se está hablando.

Pero hemos de destacar que utilizaremos otros medios de apoyo en la docencia, como pueden ser el Campus Virtual y una página web propia de la asignatura situada en: <http://www.dlsi.ua.es/asignaturas/hp/>. El uso de esta página ha sido mayoritario a lo largo de los cursos que lleva en funcionamiento por diversas razones que a continuación comentaremos. El alumno puede encontrar en ella lo siguiente:

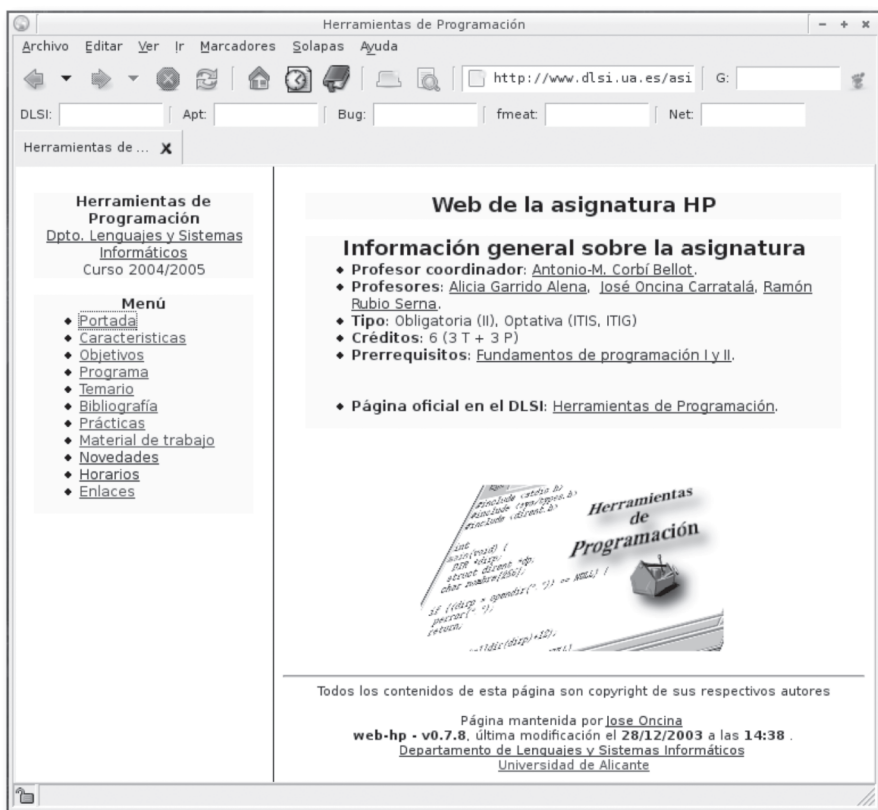


Ilustración 7. página web de la asignatura Herramientas de Programación

- Características: indica al alumno las características de la asignatura (carácter, créditos, cuatrimestre, etc.).
- Objetivos: presenta de forma clara los objetivos que se pretenden conseguir al cursar la asignatura.

- Programa: muestra de forma resumida y agrupada por bloques los contenidos a cubrir.
- Temario: enumera los temas de que consta la asignatura.
- Bibliografía: presenta de forma detallada la bibliografía de la asignatura y, en aquellos casos donde es factible, proporciona un enlace en internet al “libro” referido.
- Prácticas: proporciona acceso a los cuadernos de ejercicios para prácticas.
- Material de trabajo: proporciona acceso tanto a los ejercicios de prácticas como a los ficheros empleados en las exposiciones de clase de teoría.
- Novedades: sirve como portal de comunicación rápida de los profesores con los alumnos; por ejemplo, aquí se pueden colocar avisos, notas, días de reunión, etc.
- Horarios: muestra al alumno los horarios de teoría y prácticas, así como el aula o laboratorio donde se imparte cada clase de teoría y/o prácticas y el profesor asignado a ese turno.
- Enlaces: sirve para proporcionar al alumno enlaces a materias complementarias y útiles para el contenido de la asignatura.

Teniendo en cuenta toda la información disponible para el alumno, proponemos la estrategia de aprendizaje siguiente:

- Al inicio del curso el alumno debe obtener toda la documentación disponible de la asignatura, tanto de teoría como de prácticas.
- Para las clases de teoría es conveniente que haga una lectura previa del texto de apoyo que el profesor va a emplear en clase. Una vez realizada la clase de teoría, procederá a su estudio y si no entendiese algo puede consultarlo primero con otros compañeros, además de usar la bibliografía recomendada o comprobar su aplicación en los ejercicios resueltos. Si esto no es suficiente, se puede acudir a tutorías para intentar solucionar el problema.
- Planificación de las actividades en grupos pequeños: una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida en la propia aula, entre todos, o por el profesor, fuera del aula.
- Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Planificación de las clases de prácticas:

- Una vez terminado el tema de teoría, el alumno debería leer el enunciado de la práctica correspondiente para preguntar las dudas al profesor antes de comenzarla.

- A continuación, el alumno debe planificar la realización de la práctica, parte en los laboratorios y parte en horas no presenciales.

Auto-evaluación: aproximadamente un mes antes de terminar la asignatura, se entregará al alumno un enunciado de examen no-presencial que el alumno podrá hacer en un plazo de unas dos semanas y decidir si lo entrega a o no. Por un lado, le servirá para saber si ha ido adquiriendo los conocimientos necesarios para superar la asignatura y por otro, si entrega resuelto este examen no-presencial -no importa la calificación- tiene derecho a presentarse al examen presencial de la asignatura completa.

Evaluación final: será un promedio de las notas obtenidas en los dos exámenes (presencial y no-presencial), además de tener en cuenta la evolución y trabajo en el desarrollo de las prácticas a lo largo de todo el curso.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las dos tablas siguientes se detalla cuál va a ser el plan de trabajo de la asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES				
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos pequeños / tutorías docentes
Presentación		1	0	0
BLOQUE 1	Tema 1	2	1	0
	Tema 2	2	1	0
	Tema 3	2	1	1
BLOQUE 2	Tema 4	2	1	1
	Tema 5	2	1	1
	Tema 6	2	1	1
	Tema 7	2	2	2
BLOQUE 3	Tema 8	2	1	1
	Tema 9	2	1	1
	Tema 10	2	1	1
BLOQUE 4	Tema 11	2	2	2
	Tema 12	2	2	2
Preparación del examen final		2	0	2
Examen final:	3	0	0	
TOTAL: 60		30	15	15

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES				
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños / tutorías docentes
Presentación		0	0	0
BLOQUE 1	Tema 1	3,5	2	1
	Tema 2	3,5	2	0
	Tema 3	2	2	0,25
BLOQUE 2	Tema 4	2	2	0,25
	Tema 5	4	2	0
	Tema 6	3,5	2	0
	Tema 7	3,5	2	0
BLOQUE 3	Tema 8	4	2	0,25
	Tema 9	3,5	2	0
	Tema 10	3,5	2	0
BLOQUE 4	Tema 11	4	2	1
	Tema 12	4	2	1
Preparación del examen final:		4	2	0
Tutorías:	7,5	4	0	
TOTAL: 86,25		52,5	30	3,75

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

Las horas no presenciales de la sesión de presentación estarán dedicadas a la recopilación de la documentación de la asignatura y, en su caso, al repaso de aquellos prerrequisitos que no se hayan alcanzado. La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y entendimiento de los enunciados de los problemas y su revisión.

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

Diomidis Spinellis (2003) *Code Reading. The Open Source Perspective*. Ed. Addison Wesley.

Gough, Brian J. (2004) *An introduction to gcc*. Ed. Network Theory Ltd. Año 2004.

Kernighan, B.W.; Pike, R. (1987) *El entorno de programación Unix*. Ed. Prentice Hall.

Kernighan, B.W.; Pike, R. (2000) *La práctica de la programación*. Ed. Prentice Hall.

Loukide, M.; Oram, A. (1996) *Programming with GNU Software*. Ed. O'Reilly Media.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Brian Gough ; foreword by Richard M. Stallman (2004). *An introduction to GCC : for the GNU compilers gcc and g++* . Ed. Network Theory, Bristol.

Eric Steven Raymond (2004). *The art of Unix programming*. Ed. Addison-Wesley.

Rafeeq Ur Rehman, Christopher Paul (2003). *The Linux development platform : configuring, using, and maintaining a complete programming environment*. Ed. Prentice Hall PTR, Upper Saddle River.

Dada las características de que esta asignatura se basa por un lado en una aplicación práctica de los conocimientos y por otro en el uso de software libre, la práctica totalidad de la documentación está disponible de forma “libre” en formato electrónico. En este sentido se puede visitar el enlace a la página de la asignatura dedicada a la bibliografía donde se recogen de forma dinámica (añadiendo, quitando y corrigiendo) los enlaces a los documentos pertinentes: <http://www.dlsi.ua.es/asignaturas/hp/bibliografia.html>.

7.3 OTROS RECURSOS

- Página web de la asignatura: en ella mantenemos una sección de “enlaces” donde se recoge este tipo de información, se puede consultar en: <http://www.dlsi.ua.es/asignaturas/hp/enlaces.html>.
- Campus Virtual: servicio Internet de complemento a la docencia y a la gestión académica y administrativa. El alumno puede encontrar material de trabajo, calificaciones de exámenes y realizar consultas a los profesores.
- Correo electrónico: son cada vez más los alumnos que eligen esta vía para realizar tutorías que por su carácter así lo permiten, por tanto, podemos considerarlo como un recurso más pero siempre teniendo en cuenta que cuando la duda es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.
- Transparencias y presentaciones “on-line” de ejercicios: son elaboradas por los profesores de la asignatura; se emplean como guiones en las clases, y al disponer de un ordenador portátil, se puede comprobar in-situ aquello de lo que se está hablando.
- Problemas y ejercicios: son un conjunto de problemas y ejercicios resueltos que el alumno tiene disponibles en los cuadernos de prácticas.
- Evaluación de los procesos y resultados de aprendizaje. Sistema de evaluación.

7.4 PROCEDIMIENTOS DE EVALUACIÓN

Para evaluar esta asignatura, intentamos acercarnos lo más posible a la evaluación continua a partir de las distintas actividades propuestas a lo largo del curso. Además de tener en cuenta la evolución del alumno en las distintas prác-

ticas planteadas, se realizarán dos exámenes, uno “no presencial” y otro presencial. Atendiendo a esto, la nota de la asignatura se desglosa de la siguiente forma:

NOTA DE HERRAMIENTAS DE PROGRAMACIÓN	
Prácticas de cada uno de los temas de la asignatura	10%
Examen no presencial	20%
Examen presencial	70%

Cuadro 4. *Desglose de la nota de la asignatura*

Todos los alumnos que quieran pueden hacer el examen “no presencial” y entregar todas las prácticas y actividades realizadas a lo largo del curso. Esto les da derecho a realizar el examen presencial. La nota del examen presencial podrá tener hasta un peso del 70 % si no se resuelven correctamente los ejercicios propuestos a lo largo del curso. Sólo se superará la asignatura si en cada una de las partes se obtiene una nota mayor o igual que 5 sobre 10.

7.5 CRITERIOS DE EVALUACIÓN

Realizaremos la calificación del alumno según las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas relacionados con la asignatura son resueltos con eficacia y precisión. • La participación en las clases y distintas actividades ha sido correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre las diversas herramientas de programación estudiadas cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio son usualmente desarrolladas con éxito razonable. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.

Calificación	Rango	Pautas
Suspenso	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 5. Criterios de evaluación

8. Análisis de coherencia de la guía docente

En las siguientes tablas presentamos el análisis de coherencia de la guía docente de “*Herramientas de programación*”. En ellas se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo propuesto para el alumnado así como el sistema y criterios de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
OI1	De CIC8 a CIC10 CIM11, CIM12 cCIC1, cCIC2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1		5			Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al muestreo y la estadística descriptiva.
OI2	De CIC19 a CIC21 De CIM19 a CIM21 cCIC1, cCIC2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1			10		Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a la inferencia estadística.

Cuadro 6. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
OI3	CIC17 CIM17 cCIC1, cCIC2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1			8		Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
OI4	CIC13, CIC14 CIM13, CIM14 cCIC1, cCIC2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1		6			Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
OI5	CIC13, CIC14 CIM13, CIM14 cCIC1, cCIC2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1	1,2				Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
OI6	De CIC1 a CIC15 De CIM1 a CIM5 CIM1, CIM2 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1	1,2				Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
OI7	CIC6, CIC7 CIM6, CIM7 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1	3				Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
OI8	De CIC8 a CIC10 De CIM8 a CIM10 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1	4				Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
OI9	De CIC22 a CIC24 De CIM22 a CIM24 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1				11	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 8. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
OI10	CIC18 CIM18 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1			9		Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
OI11	CIC15, CIC16 CIM15, CIM16 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1		7			Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
OI12	De CIC25 a CIC27 De CIM25 a CIM27 cCIT1 De CIT1 a CIT3 cCIL1, cCIL2 CIL1				12	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 9. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (ser y estar)	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
cOIP1 cOIP2 cOIP3	cCIPTC1, cCIPTC2 De cCIPTR1 a cCIPTR4	1,2, 3	4, 5, 6, 7	8,9, 10	11, 12	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 10. Análisis de coherencia: competencias instrumentales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Interfaz con el S.O	Compilación	Detección y resolución de problemas	Herramientas adicionales		
cOS1 cOS2 cOS3 cOS4	De cCS1 a cCS5	1,2, 3	4, 5, 6, 7	8,9, 10	11, 12	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 11. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE LENGUAJES, GRAMÁTICAS Y AUTÓMATAS

*María Luisa Micó Andrés; Jorge Calera Rubio; Mikel L. Forcada Zubizarreta;
José Luis Verdú Más*

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
(mico; calera; mlf; verdu)@dlsi.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

La Association for Computing Machinery (ACM) es una de las dos asociaciones profesionales internacionales más importantes en el campo de la informática. La otra es una sección del Institute for Electric and Electronic Engineers: la IEEE Computer Society (IEEE-CS). En 1989, una comisión de la ACM publicó en la revista *Communications of the ACM* un resumen de un informe elaborado en el que se proponía la definición de la computación como una disciplina, definiendo y describiéndola en subáreas con la intención de servir de base para el diseño de currícula. El informe define nueve subáreas (algoritmos y estructuras de datos, lenguajes de programación, arquitectura, computación numérica y simbólica, sistemas operativos, metodología e ingeniería de la programación, sistemas de bases de datos y recuperación de datos, inteligencia artificial y robótica y comunicación ser humano-ordenador), diferenciando en cada una de ellas tres paradigmas: teoría, abstracción y diseño. Los conceptos que componen la asignatura de Lenguajes, Gramáticas y Autómatas se recogen principalmente en la subárea de lenguajes de programación, entre los que destacan:

- lenguajes formales y autómatas
- teoría del análisis sintáctico y de la traducción
- implementación de modelos abstractos

En 1990 un comité conjunto de la ACM-IEEE publicó el informe *Computing Curricula 1991*, en el que se utiliza la misma división anterior en subáreas, pero

subdivide las áreas en unidades de conocimiento más concretas que se utilizan en la composición de las asignaturas comunes de los currícula que se proponen al final del informe. Las unidades de conocimiento básicas que mejor se corresponden con Lenguajes, Gramáticas y Autómatas, son las siguientes:

- PL7: autómatas de estados finitos y expresiones regulares. Autómatas de estados finitos como modelos restringidos de computación y aceptores de expresiones regulares. Aplicación de las expresiones regulares al análisis de los lenguajes de programación.
- PL8: gramáticas independientes del contexto (GIC) y autómatas con pila. Uso de GIC como dispositivos de descripción formal para análisis sintáctico de lenguajes de programación. Equivalencia de una GIC y un autómata con pila. Uso de los autómatas con pila en el análisis sintáctico de los lenguajes de programación.

El informe finaliza con varios tipos de currícula en los que las unidades de conocimiento PL7 y PL8 aparecen casi siempre en diversas asignaturas de tercer curso, poniendo de relieve la importancia que tiene el estudio de teoría de autómatas y lenguajes formales en todas las especialidades de la Informática.

En el nuevo documento Computing Curricula 2001 de la IEEE-CS y de la ACM sobre los estudios de informática, aparecen 14 subáreas, entre las cuales la materia de Lenguajes, Gramáticas y Autómatas quedaría encuadrada en la unidad llamada Algoritmos y Complejidad. En este caso, la unidad de conocimiento básica que mejor se corresponde con Lenguajes, Gramáticas y Autómatas es la AL7: Teoría de autómatas.

Por otro lado, siguiendo las directrices generales propias publicadas en el BOE. de 20-11-90, y la modificación introducida por el decreto 779/98 en el año 2001, se aprobó el nuevo plan de estudios para las tres titulaciones de Informática de la Universidad de Alicante (B.O.E. del 25-9-2001), puesto en marcha en el curso 2001—2002. En concreto, la materia de Teoría de Autómatas y Lenguajes Formales aparece distribuida en dos asignaturas troncales de 4,5 créditos cada una. Estas dos asignaturas son troncales en Ingeniería Informática y en Ingeniería Técnica en Informática de Sistemas, y optativas en Ingeniería Informática de Gestión:

- Lenguajes, Gramáticas y Autómatas. Descripción: Máquinas secuenciales y autómatas finitos. Gramáticas y lenguajes formales. Redes neuronales.
- Computabilidad. Descripción: máquinas de Turing. Funciones recursivas.

En el libro blanco del título de grado de ingeniería informática se presenta una propuesta de contenidos dentro del marco de la declaración de Bolonia, en la que aparece la materia Procesadores de Lenguajes con los contenidos: teoría de autómatas, gramáticas y lenguajes formales, computabilidad, compiladores, traductores e intérpretes, teoría de los lenguajes de programación. Esta propuesta ha sido realizada considerando los currícula nacionales e internacionales en

informática, la preparación para los perfiles profesionales de actividad actual de los ingenieros en informática, y la preparación para los nuevos perfiles profesionales de actividad relacionados con informática.

Algunas de las aplicaciones más importantes en las que se utilizan conceptos relacionados con esta asignatura son:

- Software para explorar grandes corpus de texto (por ejemplo, páginas web), o para encontrar las apariciones de ciertos patrones.
- Analizadores léxicos en compiladores e intérpretes.
- Tareas de procesamiento léxico (por ejemplo, análisis morfológico) en sistemas de tratamiento del lenguaje humano (p.e., traductores automáticos).
- Correctores ortográficos para OCR y procesadores de textos.
- Software para comprobar los sistemas de protocolos de comunicación o protocolos para el intercambio seguro de información.
- Software para el diseño y verificación del comportamiento de circuitos digitales.

De acuerdo con las tendencias actuales, esta asignatura tiene gran importancia en la mayoría de los perfiles profesionales, que están englobados en los siguientes perfiles profesionales generales:

Perfil profesional de desarrollo de software

- Es fundamental el estudio de las teorías de lenguajes formales, autómatas, expresiones regulares y gramáticas libres del contexto, en el diseño e implementación de aplicaciones reales de software.

Perfil profesional de sistemas:

- Es fundamental el estudio de las gramáticas formales y autómatas para definir y verificar protocolos de comunicación (modelización de estados finitos).

Perfil profesional de gestión y explotación de tecnologías de la información:

- Las bases teóricas de la computación son fundamentales para el desarrollo y gestión de aplicaciones prácticas. Tecnologías del lenguaje humano.

La adecuación de los temas relacionados con la asignatura Lenguajes, Gramáticas y Autómatas a los distintos perfiles profesionales queda resumida en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	• Conocimiento y habilidad en la aplicación de las teorías de lenguajes formales en el desarrollo del software. Diseño de lenguajes.
Arquitectura y diseño de software	• Conocimiento y habilidad en la aplicación de la teoría de autómatas en el diseño del software.
Diseño multimedia	• Conocimiento y habilidad en la aplicación de las teorías de lenguajes formales en el diseño multimedia.
Ingeniería de comunicación de datos	• Conocimiento y habilidad para el estudio y aplicación de la teoría de autómatas para establecer y caracterizar protocolos de comunicación.
Diseño de redes de comunicación	• Conocimiento y habilidad para el estudio y aplicación de la teoría de autómatas para establecer diagnósticos y protocolos de comunicación.
Asistencia técnica	• Conocimiento y habilidad en la aplicación de las teorías de lenguajes formales.
Ingeniería de integración y pruebas e implantación y pruebas	• Conocimiento y habilidad en la aplicación de las teorías de lenguajes formales.
Consultoría en empresas de TI	• Conocimiento y habilidad en técnicas de planificación.
Especialista en sistemas	• Conocimiento y habilidad en la aplicación de la teoría de lenguajes al control de sistemas informáticos.
Desarrollo de investigación y tecnología	• Conocimiento y habilidad en la aplicación de la teoría de lenguajes formales a diversas tecnologías (buscadores, sistemas de tratamiento de textos).
Dirección de TIC	• Conocimiento y habilidad en la aplicación de la teoría de lenguajes formales.

Cuadro 1. *Adecuación de los perfiles profesionales a los perfiles de la asignatura*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Lenguajes, Gramáticas y Autómatas está ubicada en el segundo curso de Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas, como asignatura troncal, y de la Ingeniería Técnica en Informática de Gestión, como asignatura optativa, en la Universidad de Alicante. Aunque esta asignatura tiene relación directa con varias materias de primer curso, solamente tiene un prerrequisito en el plan de estudios, la asignatura de Fundamentos de Programación II.

Estas asignaturas son:

- **Álgebra:** forma parte del primer cuatrimestre del primer curso de las tres titulaciones de Informática como asignatura troncal. Dentro de sus descriptores aparecen la teoría de conjuntos, básica para la comprensión de la asignatura.

- **Matemática Discreta:** forma parte del primer curso de las titulaciones de informática como asignatura troncal que se imparte en el segundo cuatrimestre. Los descriptores son: “aritmética entera y modular, combinatoria y grafos”. La teoría de grafos (y sobre todo, los algoritmos asociados, como clausuras y recorridos) es básica para la asignatura.
- **Fundamentos de Programación II:** forma parte del segundo cuatrimestre del primer curso de las tres titulaciones de Informática como asignatura troncal. Los descriptores de dicha asignatura son: “análisis y diseño de programas. Lenguajes de programación. Diseño descendente”. Esta asignatura es importante ya que su conocimiento permite que el alumnado pueda realizar un trabajo práctico de programación relacionado con los contenidos de la asignatura.
- **Computabilidad:** forma parte del segundo cuatrimestre del segundo curso de las mismas titulaciones en las que se imparte la asignatura. Los descriptores de esta asignatura son: “máquinas de Turing. Funciones recursivas”. Esta materia es claramente una continuación en los conceptos de teoría de lenguajes formales, que se inician en el primer cuatrimestre con Lenguajes, Gramáticas y Autómatas.

Lenguajes, Gramáticas y Autómatas es un prerrequisito de:

- **Procesadores de Lenguajes:** asignatura troncal para la Ingeniería en Informática en cuarto curso. Los descriptores de dicha asignatura son: “compiladores. Traductores e intérpretes. Fases de compilación. Optimización de código. Macroprocesadores”.

Los conceptos de ambigüedad, análisis léxico y sintáctico a partir de GIC, autómatas finitos, o máquinas de estados finitos como las máquinas de Mealy (importantes en operaciones de procesamiento léxico) son importantes en las siguientes asignaturas optativas:

- **Aprendizaje computacional y extracción de información:** se ofrece como asignatura optativa para la Ingeniería en Informática. Los descriptores de dicha asignatura son: “técnicas de aprendizaje computacional. Extracción y recuperación de información de documentos electrónicos”.
- **Historia de la Informática y Metodología Científica:** se ofrece como asignatura optativa para la Ingeniería en Informática. Sus descriptores son: “ciencia y tecnología. Historia de la informática: el hardware y el software. Caracterización de ciencia y método científico. Ciencia y técnica como ideología”.
- **Ingeniería del Lenguaje Natural:** asignatura optativa de la Ingeniería Informática cuyos descriptores son: “procesamiento del lenguaje natural. Ambigüedad en el lenguaje. Fases de análisis. Aplicaciones”.

- **Sistemas de Información Semiestructurada:** es una asignatura optativa de la Ingeniería Informática. Los descriptors de dicha asignatura son: “marcado de texto, XML. Sistemas de publicación e indexación. Tecnologías de soporte. Aplicaciones”.

La siguiente figura resume las relaciones anteriormente descritas:

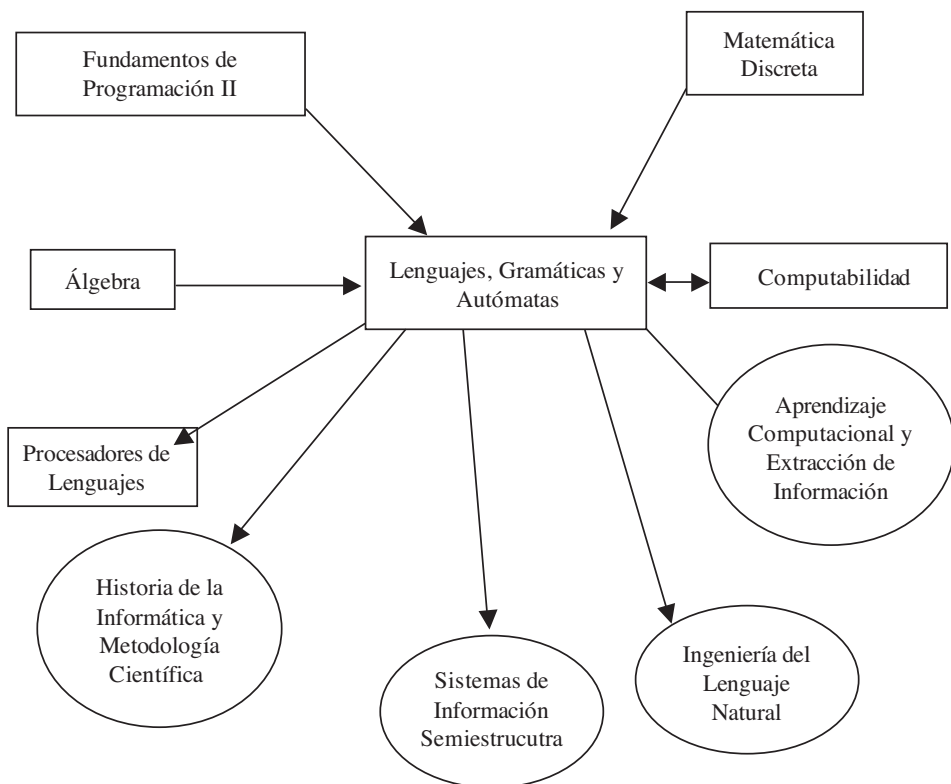


Ilustración 8. relaciones de Lenguajes, Gramáticas y Autómatas en el plan de estudios

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes:

- OI1: Conocer y comprender las teorías formales para la descripción de lenguajes naturales y artificiales.
- OI2: Conocer, interpretar y analizar algunos problemas en los que dichas teorías tienen aplicación o que han motivado su construcción.

- OI3: Conocer las relaciones existentes entre los lenguajes, las gramáticas y los autómatas, tanto en la clase de los lenguajes regulares como en la de los lenguajes incontextuales, y saber pasar con destreza de una forma de representación a otra.
- OI4: Conocer y analizar las propiedades principales que caracterizan a las clases de lenguajes estudiados y aplicarlas cuando corresponda.
- OI5: Adquirir aquellas herramientas básicas necesarias para otras asignaturas que utilizan elementos de la teoría de lenguajes.
- OI6: Desarrollar su capacidad de abstracción y análisis teórico en relación con la teoría de lenguajes, en particular, reforzar el pensamiento recursivo y la soltura en el manejo de conjuntos y variables.

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOIP1, cOIP2 y cOIP3 detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber):

Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2, desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes.

Bloque 1: Autómatas finitos y lenguajes regulares.

- CIC1: Conocer los lenguajes regulares y los distintos formalismos mediante los cuales se pueden representar autómatas finitos, gramáticas y expresiones regulares.
- CIC2: Entender la diferencia entre los tipos de autómatas finitos definidos y saber cómo se transforma un autómata finito indeterminista a determinista.

- CIC3: Conocer el funcionamiento de un traductor secuencial y saber cómo se transforma una máquina de Mealy en una máquina de Moore y viceversa.
- CIC4: Conocer las expresiones regulares como un mecanismo para representar lenguajes regulares.
- CIC5: Conocer los diferentes tipos de operaciones que se pueden realizar con lenguajes regulares y saber utilizarlos para obtener otros lenguajes regulares.
- CIC6: Conocer y comprender al menos un algoritmo para decidir la equivalencia de autómatas y otro de minimización.
- CIC7: Conocer algunas de las aplicaciones reales: almacenamiento eficiente de diccionarios (“tries” con AFD), sistemas de escritura predictiva con teclados reducidos (AFI), buscadores, etc.

Bloque 2: Gramáticas y lenguajes independientes del contexto.

- CIC8: Conocer el concepto de gramática y comprender su relación con los lenguajes naturales y artificiales.
- CIC9: Comprender la diferencia entre los distintos tipos de gramáticas en función de las reglas de derivación que se utilizan para definir las.
- CIC10: Conocer los lenguajes independientes del contexto.
- CIC11: Saber identificar lenguajes sencillos, regulares e independientes del contexto a partir de la definición de una gramática.
- CIC12: Conocer algún algoritmo (por ejemplo, el de Cocke, Younger y Kasami) para discriminar frases admisibles de un lenguaje que ha sido definido mediante una gramática independiente del contexto.
- CIC13: Conocer algunas técnicas elementales de transformación de gramáticas: cómo eliminar la recursión por la izquierda o cómo transformarla en forma normal de Chomsky o Greibach.
- CIC14: Conocer los autómatas con pila como mecanismo de análisis de un lenguaje independiente del contexto.

Capacidades metodológicas (saber hacer):

Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3, incluidas a nivel general en el artículo introductorio de este libro.

Bloque 1: Autómatas finitos y lenguajes regulares.

- CIM1: Ser capaz de representar lenguajes regulares con los mecanismos estudiados en clase: autómatas finitos y expresiones regulares.
- CIM2: Ser capaz de aplicar las operaciones definidas sobre lenguajes regulares para obtener otros lenguajes regulares más complejos.
- CIM3: Ser capaz de construir traductores secuenciales tales como máquinas de Mealy o Moore.

- CIM4: Ser capaz de aplicar al menos un algoritmo de equivalencia y de minimización de autómatas.

Bloque 2: Gramáticas y lenguajes independientes del contexto.

- CIM5: Ser capaz de aplicar un algoritmo de análisis sintáctico para comprobar la pertenencia de una cadena a un lenguaje independiente del contexto.
- CIM6: Ser capaz de definir gramáticas independientes del contexto, sencillas (ambiguas o no), que describan lenguajes independientes del contexto.
- CIM7: Ser capaz de simplificar y normalizar gramáticas independientes del contexto.
- CIM8: Ser capaz de construir un autómata a pila a partir de una GIC.

Destrezas tecnológicas (saber hacer):

Además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con fluidez las herramientas de linux/unix, que permiten realizar análisis léxico mediante el uso de expresiones regulares (lex).
- CIT2: Manejar con fluidez las herramientas de edición, implementación y compilación de programas en C/C++ que servirá de guía para la resolución de problemas relacionados con la asignatura.

Destrezas lingüísticas (saber hacer):

Además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el capítulo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje matemático, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso ya sea algorítmico o teórico.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas: las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el capítulo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo: las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el capítulo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el capítulo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Tener nociones básicas sobre teoría de conjuntos.
- Entender los conceptos relacionados con la teoría de grafos.
- Entender los conceptos relacionados con la aritmética entera y modular.
- Conocer el proceso de diseño de un algoritmo para construir autómatas finitos.
- Comprender, analizar y aplicar los elementos básicos de un lenguaje de programación de alto nivel para construir autómatas finitos para el reconocimiento y análisis de lenguajes regulares.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Aunque según el vigente plan de estudios, el único prerrequisito de Lenguajes, Gramáticas y Autómatas es la asignatura Fundamentos de Programación II, asignatura troncal de primer curso en las tres titulaciones, los prerrequisitos necesarios para el estudio y entendimiento de esta materia requieren además el conocimiento de conceptos que se desarrollan en las asignaturas de Álgebra y Matemática Discreta, ambas también impartidas en el primer curso de las tres titulaciones de informática. Para paliar el problema que conllevaría el estudio de esta materia sin el conocimiento de las materias de estas últimas asignaturas, se intenta, en la medida de lo posible, dar a los temas un carácter bastante autocontenido, recordando, al principio de cada uno, aquellos conceptos que van a necesitar para entenderlo. Además, se incorpora al principio del temario un breve tema recordatorio sobre teoría de conjuntos.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Autómatas finitos y lenguajes regulares.

- Tema 1: Nociones básicas de teoría de conjuntos.
- Tema 2: Lenguajes y computadores.
- Tema 3: Autómatas finitos.
- Tema 4: Expresiones y lenguajes regulares.
- Tema 5: Propiedades de los lenguajes regulares.

Bloque 2: Gramáticas y lenguajes independientes del contexto.

- Tema 6: Gramáticas y lenguajes independientes del contexto.
- Tema 7: Autómatas con pila.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1: Autómatas finitos y lenguajes regulares.

- Tema 1: Nociones básicas de teoría de conjuntos.
 - 1.1. Conjunto. Definiciones y propiedades.
 - 1.2. Correspondencias y relaciones.
 - 1.3. Cardinal. Conjuntos infinitos.
 - 1.4. Conjuntos numerables. Propiedades y ejemplos.
 - 1.5. Conjuntos no numerables. Ejemplos.
- Tema 2: Lenguajes y computadores.
 - 2.1. Alfabetos y lenguajes.
 - 2.2. Lenguajes y computadores.
 - 2.3. Problemas y lenguajes.
- Tema 3: Autómatas finitos.
 - 3.1. Definición y representaciones de un autómata finito determinista (AFD).
 - 3.2. El AFD como clasificador. Ejemplos.
 - 3.3. Lenguaje aceptado por un AFD.
 - 3.4. El AFD como traductor. Máquinas de Moore y de Mealy.
 - 3.5. Autómatas finitos indeterministas (AFI). Ejemplos.
 - 3.6. Lenguaje aceptado por un AFI.
 - 3.7. Autómatas finitos estocásticos. Ejemplos.
- Tema 4: Expresiones y lenguajes regulares.
 - 4.1. Definición de expresión regular (ER). Ejemplos
 - 4.2. Propiedades de las ER.
 - 4.3. Equivalencia entre ER y autómatas finitos.
- Tema 5: Propiedades de los lenguajes regulares
 - 5.1. Propiedades de clausura de los lenguajes regulares.
 - 5.2. Algoritmos y decidibilidad.
 - 5.3. Equivalencia y minimización de autómatas.
 - 5.4. Lema del bombeo para lenguajes regulares.

Bloque 2: Gramáticas y lenguajes independientes del contexto.

- Tema 6: Gramáticas y lenguajes independientes del contexto.
 - 6.1. Definición y notación de las gramáticas.
 - 6.3. Lenguaje generado por una gramática y tipos de gramáticas.
 - 6.4. Gramáticas regulares. Ejemplos.

- 6.5. Gramáticas independientes del contexto (GIC). Ejemplos.
- 6.6. Análisis sintáctico (algoritmo de Cocke, Younger y Kasami).
- 6.7. Simplificación de GIC.
- 6.8. Formas normalizadas de Chomsky y Greibach.
- Tema 7: Autómatas con pila.
 - 7.1. Definiciones y notación de los autómatas con pila (AP).
 - 7.2. Autómata con pila para una GIC.
 - 7.3. Gramática correspondiente a un AP.
 - 7.4. Análisis sintáctico.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

En los últimos años, y sobre todo a partir de la reforma de las enseñanzas medias, se ha ido cuestionando cada vez más la metodología tradicional en la que el profesor estructura el conocimiento como algo coherente y armonioso y se lo presenta a sus alumnos. Esta crítica viene dada porque no induce en el receptor el aprendizaje de habilidades y establece, en general, una barrera entre el conocimiento y la aplicación del mismo. En el lado opuesto se han venido desarrollando numerosas experiencias, tanto en las enseñanzas medias como en algunos sectores universitarios, basadas en la metodología constructivista. Este modelo de enseñanza por investigación en el que el profesor plantea problemas abiertos, generales, que deben generar interés (motivar al alumno) y dar una visión preliminar adecuada de la materia, para después proponer al alumnado el estudio detallado de cada problema, con el fin de prepararlo para su solución en un proceso creativo. Sin embargo, a pesar de la coherencia de esta metodología, este método está siendo rebatido por sectores amplios del profesorado, básicamente por los inconvenientes que conlleva su aplicación, algunos de los cuales son: el alto número de alumnos por aula, la dificultad para elegir problemas suficientemente representativos y concretos, el tiempo requerido para la metodología (superior al tradicional), etc.

Particularizando esta discusión a la asignatura que nos ocupa, con un contenido eminentemente abstracto, desarrollado históricamente a partir de los problemas concretos a los que se ha ido enfrentando la comunidad científica en su intento de utilizar lenguajes de alto nivel en la programación, de traducir automáticamente, así como de matematizar el lenguaje humano para procesarlo automáticamente, nos conduce a adoptar un camino intermedio entre ambos métodos de aprendizaje que expondremos a continuación:

- **Clases de teoría con apoyo de material audiovisual:** en lo que se refiere a las clases de teoría, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumno antes de comenzar el curso y que le puede servir de guía sobre los contenidos más importantes de la asignatu-

ra. Además, existe un libro editado por algunos profesores de la materia que incluye todos los contenidos de la misma. Al comenzar una clase, siempre es necesario hacer un breve resumen de lo que se vio en la clase anterior y, si se comienza un nuevo tema, establecer bien su conexión con conceptos ya estudiados, si es necesario, a través de un breve diálogo o cuestiones al alumnado. En esta asignatura hay una gran cantidad de temas en los que es posible conducir a los alumnos hacia conceptos que se pueden definir a través del seguimiento de un ejemplo o problema práctico inicial. Esto pone a prueba su capacidad de razonar, y el hecho de que estos conceptos surjan de una forma más o menos natural facilita enormemente el proceso de asimilación de los mismos. En las clases, además de transmitir los conceptos, se proponen actividades, bien seleccionadas, para que los alumnos propongan soluciones que serán discutidas por todos.

- **Actividades en grupos pequeños / tutorías docentes:** estas actividades estarán vinculadas con la realización de problemas y cuestiones teórico-prácticas relacionadas con la asignatura, que supongan una comprensión más profunda de la misma. Para facilitar la realización de algunos ejercicios, se ha construido una aplicación (applet de Java) que proporciona la solución correcta a un buen número de operaciones sobre autómatas, y que el alumno puede utilizar, para comprobar su solución, desde cualquier ordenador conectado a Internet.
- **Prácticas de laboratorio:** se pretende que en las prácticas los alumnos mejoren el dominio en la programación, adquirido el curso anterior en la asignatura de Fundamentos de Programación II, y aprendan a utilizar en aplicaciones concretas las herramientas que proporciona la teoría de lenguajes. El dominio de estas herramientas será imprescindible en futuros cursos. Las clases en el laboratorio se centran en la explicación de los diversos pasos que el alumno debe seguir para la construcción de sus algoritmos.
- **Atención al alumnado:** el alumnado tiene a su disposición unas horas de tutoría o atención al alumnado en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como dudas concretas sobre el contenido de la misma. Además, en el departamento de DLSI, se ha puesto en marcha un sistema automático de reserva de tutorías que está teniendo bastante aceptación entre el alumnado, y que permite flexibilizar y optimizar su uso. Gracias a las nuevas formas de comunicación aparecidas, como son el correo electrónico y la Universidad Virtual, las tutorías están aumentando su popularidad entre los alumnos. Su uso se incrementa notablemente en épocas cercanas a los exámenes (como ocurre con las tutorías presenciales), pero a su vez, permite una mayor flexibilidad horaria al momento de realizar una consulta, no solamente al

alumno para realizarla, sino al profesor para contestarla. Lógicamente el uso de foros electrónicos permitiría una menor sobrecarga de mensajes en épocas muy concretas, ya que en muchos casos las dudas que surgen son las mismas, y por otro lado, los estudiantes pueden colaborar en la resolución de problemas que plantean otros compañeros, fomentando de esta manera el aprendizaje colaborativo y el trabajo en equipo.

5.2 ESTRATEGIAS DE APRENDIZAJE

Además de los medios de apoyo como las transparencias, apuntes o presentaciones por ordenador, las páginas web de los departamentos y el campus virtual ofrecen innumerables posibilidades de servicios a la docencia tanto al profesorado como al alumnado. Así, podemos encontrar una página de la asignatura, tanto a través de las páginas del departamento responsable de su docencia, como del campus virtual, donde se incluye toda la información que el alumno necesita. Así como actualmente los materiales aparecen publicados en la página web de la asignatura, ya que de esta forma son públicos y accesibles para cualquier persona interesada en los mismos, estos materiales se incorporarán también al campus virtual (hoy en día la mitad de los materiales de cualquier materia se pueden encontrar en ambos lugares).

Entre otras cosas, la información disponible será la siguiente:

- Información general: profesores, características de la asignatura (créditos, prerrequisitos, recomendaciones, etc).
- Objetivos que se pretende alcanzar en la asignatura.
- Programa de la asignatura.
- Programa de prácticas.
- Criterios de evaluación.
- Bibliografía básica y complementaria (libros recomendados de la asignatura).
- Tablón de anuncios (o últimas noticias): desde aquí, el alumno puede estar perfectamente informado de cualquier novedad relacionada con la asignatura. Además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, publicación de notas, etc.
- Material docente de la asignatura: Aquí se puede encontrar todo el material que como mínimo va a ser necesario para el seguimiento de las clases teóricas y prácticas.
- Applet de Java para el desarrollo de numerosas operaciones con autómatas y expresiones regulares.

En base a todo esto, la estrategia de aprendizaje que se propone se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura. Los alumnos tendrán disponible todo el material necesario para el desarrollo de la teoría y las

prácticas de la asignatura en la página web de la misma y en el campus virtual antes del comienzo de la impartición del módulo correspondiente. Cada vez que haya disponible un nuevo material, o se realice la actualización de alguno ya publicado, se enviará un mensaje a los alumnos notificando el cambio en los materiales.

Planificación de las clases teóricas:

- Las clases teóricas se dedicarán tanto a la presentación de contenidos nuevos como a la discusión y resolución de problemas.
- Una vez realizada la clase de teoría, el alumno debe estudiar de forma autónoma su contenido y, en caso de no entender algo, intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- En las actividades en grupos pequeños, cada subgrupo tendrá que realizar la actividad propuesta.

Planificación de las clases prácticas:

- Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.
- Las prácticas se realizarán en los laboratorios. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá dichas prácticas, indicando a continuación los fallos más comunes. Cada estudiante debe analizar de forma individual cuáles han sido sus errores para intentar no volver a cometerlos. Si es necesario, se pedirá ayuda al profesor correspondiente.

Evaluación final: Si los alumnos han sido capaces de realizar satisfactoriamente los ejercicios propuestos y los ejercicios desarrollados disponibles en los materiales, los alumnos podrán realizar con garantías el examen final de la asignatura. Para la evaluación final, además se tendrá en cuenta el resultado de las prácticas propuestas y desarrolladas a lo largo del cuatrimestre.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se esquematiza el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

NÚMERO DE HORAS PRESENCIALES					
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos Pequeños / tutorías docentes	
Presentación		0.5			
BLOQUE 1	Tema 1	1.5			
	Tema 2	1			
	Tema 3	2	2	1	
	Tema 4	2	2	1,5	
	Tema 5	2	1,5	2	
BLOQUE 3	Tema 6	3	1	1	
	Tema 7	1	1		
Preparación del examen final		2	0	2	
Examen final:		1,5	0	0	
TOTAL: 46,5		30	7,5	7,5	

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES					
ACTIVIDAD		Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños / tutorías docentes	
Presentación:		1	1	0,5	
BLOQUE 1	Tema 1	4	2		
	Tema 2	2	2		
	Tema 3	6	2		
	Tema 4	6	4	1	
	Tema 5	6	4	1	
BLOQUE 2	Tema 6	8	4		
	Tema 7	3	3		
Preparación del examen final:		7			
Tutorías:		3			
TOTAL: 60		46	21	2,5	

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

Las horas no presenciales de las sesiones de presentación estarán dedicadas a la recopilación de la documentación de la asignatura tanto para teoría como para prácticas y, en su caso, al repaso de aquellos prerrequisitos que no se hayan alcanzado. La columna correspondiente a horas no presenciales de las actividades en grupos pequeños corresponderá con la lectura y comprensión de los enunciados de los problemas, y revisión de los problemas ya corregidos por el profesorado.

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

- Carrasco, R.C.; Calera, J.; Forcada, M.L. (2000) *Teoría de lenguajes, gramáticas y autómatas para informáticos*. Publicaciones de la Universidad de Alicante.
- Carrasco, R.C.; Forcada, M.L. *Introducció a la Teoria de Llenguatges Formals* (material en línea).
- Ferri, F. (2004) *Teoria d'automats I llenguatges formals*. PUV.
- Hopcroft, J.E.; Motwani, R.; Ullman, J. (2002) *Introducción a la teoría de autómatas, lenguajes y computación*. Addison Wesley.
- Hopcroft, J.E.; Ullman, J. (1979) *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.
- Isasi, P.; Martínez, P.; Borrajo, D. (1999) *Lenguajes, gramáticas y autómatas: un enfoque práctico*. Addison Wesley.
- Kelley, D. (1999) *Teoria de autómatas y lenguajes formales*. Addison Wesley.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

- Aho, A.V.; Ullman, J. (1972) *The theory of parsing, translation and compiling*. Prentice Hall.
- Alfonseca, M.; Sancho, J.; Martínez, M. (1997) *Teoria de lenguajes, gramáticas y autómatas*. R.A.E.C.
- Brookshear, J.G. (1999) *Teoría de la computación: lenguajes formales, autómatas y complejidad*. Pearson Education.
- De Sande, F. (2001) *Prácticas de teoría de autómatas y lenguajes formales*. Gobierno de Canarias.
- Martín, J.C. (2003) *Introduction to languages and the theory of computation*. McGraw-Hill.
- Rocha, J.; Rosselló, F. (2003) *Autòmats i llenguatges: verificació, implementació i concurrència*. Universitat de les Illes Balears.

7.3 OTROS RECURSOS

Además de la bibliografía referida previamente, el alumnado dispone de una serie de recursos adicionales que facilitan el aprendizaje de la asignatura:

- **Página web de la asignatura:** información completa y actualizada de la asignatura a la que el alumno puede acudir en cualquier momento, durante el curso académico.

Contiene información acerca de :

4. Administración: profesores, aulas y horarios.
 5. Evaluación: fechas y criterios de evaluación.
 6. Clases teóricas: materiales y enlaces relacionados.
 7. Clases prácticas: enunciados, software y enlaces de interés
- **Campus Virtual:** servicio Internet de complemento a la docencia y a la gestión académica y administrativa. El alumno puede encontrar material de trabajo, calificaciones de exámenes y realizar consultas a los profesores.
 - **Applet** implementado en java para aplicar muchos de los métodos tratados en esta asignatura, así como su interpretación. Esta herramienta les permite familiarizarse con los lenguajes regulares y las operaciones que se pueden realizar con ellos, sirve también de banco de pruebas y para corregir muchos de los ejercicios propuestos en clase.

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que frecuentemente se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces es fácil la resolución de dudas de esta forma. No obstante, cuando la duda es de resolución compleja, es preferible la asistencia presencial a horas de tutoría ya que es la forma de asegurarnos que se ha entendido la explicación.

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

8.1 PROCEDIMIENTOS DE EVALUACIÓN

Para evaluar esta asignatura intentamos acercarnos lo más posible a la evaluación continua a partir de las distintas actividades propuestas a lo largo del curso. Concretamente, a aquellos alumnos que asistan a clase se les realizará un seguimiento de los conocimientos adquiridos y de la realización de las distintas actividades. También se tendrá en cuenta la actitud mostrada por el alumno hacia la asignatura. Atendiendo a esto, la nota final de la asignatura se desglosa de la siguiente forma:

NOTA DE LENGUAJES, GRAMÁTICAS Y AUTÓMATAS	
Prácticas de uso de expresiones regulares con la herramienta lex	10%
Prácticas de implementación de un programa	10%
Resolución de ejercicios propuestos	20%
Examen	60%

Cuadro 4. Desglose de la nota de la asignatura

Todos los alumnos deberán hacer un examen final y entregar todas las prácticas y actividades realizadas a lo largo del curso. La nota del examen podrá tener hasta un peso del 80% si no se resuelven correctamente los ejercicios propuestos a lo largo del curso. Sólo se superará la asignatura si en cada una de las partes (teoría y prácticas) se obtiene una nota mayor o igual que 5, sobre 10. La tabla anterior detalla los porcentajes en caso de que los alumnos hayan resuelto correctamente durante el curso los ejercicios propuestos.

8.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es excelente. • Los problemas relacionados con la asignatura son resueltos con eficiencia y precisión.
Notable	7,0-8,9	<ul style="list-style-type: none"> • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria. • El conocimiento sobre teoría de lenguajes formales cubre de manera satisfactoria el programa. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio son usualmente desarrolladas con éxito razonable. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspense	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 5. *Criterios de evaluación*

Queremos hacer notar que para la obtención de matrícula de honor es necesario obtener un sobresaliente alto.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Lenguajes, Gramáticas y Automatas. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo de propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI1	De CIC1 a CIC3 cCIC1 cCIC2	2,3,4,5		Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a lenguajes y autómatas finitos.

Cuadro 6. *Análisis de coherencia: competencias instrumentales (saber)*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI2	CIC3 CIC4 CIC7 cCIC1 cCIC2	2,3		Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a lenguajes y autómatas finitos.
OI3	De CIC1 a CIC11 cCIC1 cCIC2	2,3	6	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión conceptual.
OI4	De CIC5 a CIC11 cCIC1 cCIC2	1,5	6	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión conceptual.
OI5	De CIC4 a CIC14 cCIC1 cCIC2	4,5	6,7	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión conceptual.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)		
OI6	De CIC1 a CIC14	1,5	6	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental. y la resolución del problema.

Cuadro 8. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)		
cOI1	De CIM1 a CIM8 CIT1 CIT2 cCIT1	3, 4,5	6,7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental. y la resolución del problema.

Cuadro 9. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)		
cOI2	CIT1 CIT2 cCIT1	3,4,5	6,7	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ realización de prácticas propuestos). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI3	CIL1 cCIL1	1,2,3,4,5	6,7	Enseñanza presencial (lección magistral / trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de rigurosidad en las explicaciones de los procedimientos aplicados.
cOI4	cCIL2	1,2,3,4,5	6,7	Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de conocimiento de los distintos términos estudiados, en castellano y/o valenciano y en inglés.
cOI5	De CIM1 a CIM12 cCIM3	1,2,3,4,5	6,7	Enseñanza no presencial (aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, análisis y aplicación de los conceptos.
cOI6	cCIC1 cCIC2	1,2,3,4,5	6,7		

Cuadro 10. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS		PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)		
cOIP1	cCIPTC1 cCIPTC2	3,4,5	6,7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTC2 cCIPTR1 cCIPTR2 cCIPTR3	3,4,5	6,7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4	3,4,5	6,7	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 11. Análisis de coherencia: competencias interpersonales (ser y estar)

GUÍA DOCENTE DE LENGUAJES, Y PARADIGMAS DE PROGRAMACIÓN

Domingo Gallardo López; Cristina Pomares Puig

*Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
{domingo; cpomares}@dccia.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN.

En este apartado especificamos el lugar que ocupa la asignatura Lenguajes y Paradigmas de Programación en el Computing Curricula 2001. Esta asignatura se encuadra dentro del área de Programming Languages (PL), concretamente las subáreas PL5 Abstraction mechanisms, PL7 Functional programming, PL1 Overview of programming languages, PL6 Object-Oriented programming incluye gran parte del temario, aunque se pueden identificar algunas subáreas de las áreas de Programming Fundamentals (PF1 y PF4) que completan el contenido de la misma. Se puede constatar que todas estas subáreas se encuentran clasificadas como troncales (core). A continuación se describen las subáreas relacionadas con la asignatura.

1.1.1 Área de Lenguajes de Programación

Un lenguaje de programación es la interface principal con el ordenador. En lugar de saber programar en un determinado lenguaje, los programadores necesitan comprender los diferentes estilos de programación propuestos por distintos lenguajes. En la vida profesional tendrán que trabajar con diferentes lenguajes y estilos a la vez, y se encontrarán también con distintos lenguajes en sus distintos puestos de trabajo. Si se comprende la variedad de lenguajes de programación desde el punto de vista de los compromisos de diseño que imponen cada uno de ellos, se hace mucho más fácil dominar nuevos lenguajes con rapidez.

El área consta de once unidades de conocimiento:

- PL1 Overview of programming languages (2)
- PL2 Virtual machines (1)
- PL3 Introduction to language translation (2)
- PL4 Declarations and types (3)
- PL5 Abstraction mechanisms (3)
- PL6 Object-oriented programming (10)
- PL7 Functional programming
- PL8 Language translation systems
- PL9 Type systems
- PL10 Programming language semantics
- PL11 Programming language design

1.1.2 Área de Fundamentos de Programación

El conocimiento y manejo de un lenguaje de programación es un prerrequisito para el estudio de prácticamente cualquier disciplina dentro de las ciencias de la computación. En el informe de 1991 ya se identificó el conocimiento de un lenguaje como esencial, aunque no se le dio suficiente énfasis. El área *Introduction to a Programming Language* del informe de 1991 tenía asignadas solo 12 horas de clase y se lo consideraba opcional bajo el supuesto (optimista) de que un creciente número de estudiantes obtiene experiencia en la escuela secundaria. En el presente informe se sostiene que el programa de Ciencias de la Computación debe proporcionar el conocimiento fluido de al menos un lenguaje de programación. Más aun, se recomienda que los programas deberían formar a los estudiantes en lenguajes que hagan uso de al menos dos paradigmas de programación. Para conseguir este último objetivo, se requieren bastante más que 12 horas. Esta área de conocimiento comprende aquellas habilidades y conceptos que son esenciales en la práctica de la programación, independientemente del paradigma concreto. Como resultado, incluye unidades sobre conceptos de fundamentos de programación, mecanismos de abstracción, estructuras de datos básicas y procesos algorítmicos. Estas unidades no cubren en ningún caso el rango completo de conocimientos de programación que debe poseer un ingeniero. Muchas de las otras áreas, principalmente las de *Programming Languages (PL)* y *Software Engineering (SE)* también poseen unidades relativas a programación que son parte troncal (*core*) de la carrera.

El área consta de cinco unidades de conocimiento:

- PF1 Fundamental programming constructs (9)
- PF2 Algorithm and problem solving (6)
- PF3 Fundamental data structures (14)
- PF4 Recursion (5)
- PF5 Event-driven programming (4)

La asignatura Lenguajes y Paradigmas de Programación que actualmente imparto está descrita, de acuerdo con los contenidos, en las siguientes unidades de conocimiento:

- PF1 Fundamental programming constructs (9)
 - PF3 Fundamental data structures (14)
 - PF4 Recursion (5)
 - PL4 Declarations and types (3)
 - PL5 Abstraction mechanisms (3)
 - PL6 Object-oriented programming (10)
 - PL7 Functional programming
 - PL11 Programming language design
- **PF1. Fundamental programming constructs** [troncal] Tiempo mínimo asignado: 9 horas

TEMAS DE CLASE
<ul style="list-style-type: none"> • Sintaxis y semántica de un lenguaje de alto nivel. • Variables, tipos, expresiones y asignación. • Entrada/Salida. • Estructuras de control condicionales e iterativas. • Funciones y paso de mensajes. • Descomposición estructurada.

Cuadro 1. *Temas de la unidad PF1*

OBJETIVOS DOCENTES
<ul style="list-style-type: none"> • Analizar y explicar el comportamiento de programas sencillos en base a los fundamentos de la programación. • Modificar y expandir pequeños programas que usen condicionales, estructuras iterativas y funciones. • Diseñar, implementar, probar y depurar un programa que utilice las siguientes construcciones fundamentales de programación: computación básica, entrada/salida, condicionales, iteración y definición de funciones. • Elegir el condicional y la iteración apropiados para una tarea dada. • Aplicar las técnicas de descomposición estructurada para “partir” un programa en piezas más pequeñas. • Describir los mecanismos de paso de mensajes.

Cuadro 2. *Objetivos docentes de la unidad PF1*

- **PF4. Fundamental data structures** [troncal] Tiempo mínimo asignado: 14 horas

TEMAS DE CLASE
<ul style="list-style-type: none">• Tipos primitivos.• Registros.• Cadenas y procesamiento de cadenas.• Representación de datos en memoria.• Static, pilas y heap allocation.• Manejo de almacenamiento en memoria.• Punteros y referencias.• Estructuras enlazadas.• Implementación de estrategias para pilas, colas y tablas hash.• Implementación de estrategias para árboles y grafos.• Estrategias de selección de la mejor estructura de datos.

Cuadro 3. *Temas de la unidad PF4*

OBJETIVOS DOCENTES
<ul style="list-style-type: none">• Representación y uso de tipos de datos primitivos y construcción de estructuras de datos.• Describir la reserva y uso de memoria de las estructuras de datos.• Describir aplicaciones básicas de manejo de estructuras de datos.• Implementar estructuras de datos en un lenguaje de alto nivel.• Comparar implementaciones alternativas de estructuras de datos.• Escribir programas que utilicen las siguientes estructuras: arrays, registros, cadenas, listas enlazadas, pilas, colas y tablas hash.• Comparar y contrastar el coste y los beneficios de implementaciones de estructuras de datos dinámicas y estáticas.• Seleccionar la mejor estructura de datos para el modelado de un problema.

Cuadro 4. *Objetivos docentes de la unidad PF4*

- **PL4. Declarations and types** [troncal] Tiempo mínimo asignado: 3 horas

TEMAS DE CLASE
<ul style="list-style-type: none">• La concepción de tipos como un conjunto de valores junto con otro conjunto de operaciones.• Modelos de declaración (binding, visibility, scope, and lifetime).• Introducción a la comprobación de tipos.• Garbage collection.

Cuadro 5. *Temas de la unidad PL4*

OBJETIVOS DOCENTES
<ul style="list-style-type: none"> • Explicar el valor de los modelos de declaración, especialmente con respecto a la programación a gran escala. • Identificar y describir las propiedades de una variable tales como su dirección asociada, valor, alcance (scope), persistencia y tamaño. • Comprensión de la incompatibilidad de tipos. • Demostrar las diferentes formas de gestión de binding, visibility, scoping, and lifetime. • Defensa de la importancia de los tipos y su comprobación como aporte a la abstracción y corrección. • Evaluación de compromisos en lifetime management (reference counting vs. garbage collection).

Cuadro 6. *Objetivos docentes de la unidad PL4*

- **PL5. Abstraction mechanisms** [troncal] Tiempo mínimo asignado: 3 horas

TEMAS DE CLASE
<ul style="list-style-type: none"> • Procedimientos, funciones, e iteradores como mecanismos de abstracción. • Mecanismos de parametrización (referencia vs. valor). • Registros de activación y gestión de almacenamiento. • Tipos de parámetros y tipos parametrizados. • Módulos en los lenguajes de programación.

Cuadro 7. *Temas de la unidad PL5*

OBJETIVOS DOCENTES
<ul style="list-style-type: none"> • Explicar los mecanismos de abstracción como soporte a la creación de componentes de software reutilizables. • Demostrar las diferencias entre el paso de parámetros por valor y por referencia. • Defensa de la importancia de las abstracciones, especialmente con respecto a la programación a gran escala. • Describir el uso de los registros de activación en la gestión de los módulos de programas y los datos.

Cuadro 8. *Objetivos docentes de la unidad PL5*

- **PL6. Object-oriented programming** [troncal] Tiempo mínimo asignado: 10 horas

TEMAS DE CLASE
<ul style="list-style-type: none"> • Diseño orientado a objeto. • Encapsulado y ocultamiento de información. • Separación de comportamiento e implementación. • Clases y subclases. • Herencia (overriding, dynamic dispatch). • Polimorfismo (polimorfismo de subtipo vs. herencia). • Jerarquías de clases. • Clases colección y protocolos de iteración. • Representación interna de objetos y tablas de métodos.

Cuadro 9. *Temas de la unidad PL6*

OBJETIVOS DOCENTES
<ul style="list-style-type: none">• Justificar la filosofía del diseño orientado a objetos y los conceptos de encapsulado, abstracción, herencia y polimorfismo.• Diseñar, implementar, probar, y depurar programas sencillos con un lenguaje de programación orientado a objetos.• Describir cómo el mecanismo de clase da soporte al encapsulado y al ocultamiento de información.• Diseñar, implementar, y probar la implementación de una relación "is-a" entre objetos usando una jerarquía de clases y herencia.• Comparar y contrastar las nociones de sobrecarga y sobreescritura (overriding) de métodos en un lenguaje de programación orientado a objetos.• Explicar la relación entre la estructura estática de la clase y la estructura dinámica de las instancias de la clase.• Describir cómo acceden los iteradores a los elementos de un contenedor

Cuadro 10. *Objetivos docentes de la unidad PL6*

• **PL7. Functional programming** [optativa]

TEMAS DE CLASE
<ul style="list-style-type: none">• Motivación de los lenguajes funcionales.• Recursión sobre listas, números naturales, árboles y otros datos definidos recursivamente.• Pragmática (depuración sobre divide y vencerás; persistencia de las estructuras de datos).• Eficiencia amortizada para estructuras de datos funcionales.• Clausura y uso de funciones como datos (conjuntos infinitos, streams).

Cuadro 11. *Temas de la unidad PL7*

OBJETIVOS DOCENTES
<ul style="list-style-type: none">• Revisión del paradigma de la programación funcional.• Diseño, codificación, chequeo y depuración de programas usando programación funcional.• Explicar el uso de funciones como datos, incluyendo el concepto de clausura.

Cuadro 12. *Objetivos docentes de la unidad PL7*

• **PL11. Programming language design** [optativa]

TEMAS DE CLASE
<ul style="list-style-type: none">• Principios generales del diseño de lenguajes.• Objetivos del diseño.• Modelos de estructuras de datos.• Modelos de estructuras de control.• Mecanismos de abstracción.

Cuadro 13. *Temas de la unidad PL11*

OBJETIVOS DOCENTES
<ul style="list-style-type: none"> • Evaluar el impacto del diseño y uso de lenguajes y el proceso de traslación. • Explicar el rol de los diferentes mecanismos de abstracción en la creación de facilidades definidas por el usuario.

Cuadro 14. *Objetivos docentes de la unidad PL7*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Lenguajes y Paradigmas de Programación aparece en los actuales planes de estudio de la Escuela Politécnica Superior de la Universidad de Alicante, en la titulación de Ingeniería Informática con las características que se muestran en la siguiente tabla:

Lenguajes y Paradigmas de Programación	
Carácter	Obligatoria
Carga docente	3 créditos teóricos + 3 créditos prácticos
Curso	Segundo
Duración	Cuatrimestral (15 semanas)
Programación	2 h/sem T + 2 h/sem P
Prerrequisitos	Fundamentos de Programación I
Descriptor	Programación procedimental. Programación funcional. Programación declarativa.
Áreas de conocimiento	Programación orientada a objetos. Lenguajes de Script.. Ciencias de la Computación e Inteligencia Artificial

Cuadro 15. *Características de la asignatura Lenguajes y Paradigmas de la Programación*

Dicha asignatura tiene como incompatibilidad una única asignatura:

- **Fundamentos de Programación I:** troncal de primer curso. Descriptores: Introducción a la programación. Diseño de algoritmos.

Aunque Fundamentos de Programación I es la única asignatura prerrequisito para Lenguajes y Paradigmas de Programación, existe una relación clara y estrecha con varias asignaturas de los primeros cursos donde se imparten conceptos básicos. Concretamente, dichas asignaturas son:

- **Fundamentos de Programación II:** troncal de primer curso. Descriptores: Análisis y diseño de programas. Lenguajes de programación. Diseño descendente.
- **Programación y estructuras de datos:** troncal de segundo curso. Descriptores: Estructura de datos y algoritmo de manipulación. Tipos abstractos de datos. Diseño recursivo.

- **Programación orientada a objetos:** obligatoria de segundo curso. Descriptores: Características de la POO. Clases y objetos. Diseño orientado a objetos. Lenguajes de programación orientada a objetos. Objetos distribuidos. Herencia y generidad. Persistencia en un entorno orientado a objetos.

Por otro lado, la asignatura Lenguajes y Paradigmas de Programación, por tratar una materia introductoria a la programación y a la ciencia de la computación, sirve como fundamento para todas las asignaturas de programación y, por lo tanto, es fundamental para el desarrollo de la mayoría de los contenidos de las asignaturas. Esta asignatura expone a los estudiantes distintas técnicas de abstracción utilizando, entre otros, funciones de orden superior, programación dirigida por los datos, paso de mensajes o patrones.

Además, Lenguajes y Paradigmas de Programación aparece como prerrequisito en la asignatura troncal de cuarto curso Procesadores del Lenguaje:

- **Procesadores del lenguaje:** troncal de cuarto curso. Descriptores: Compiladores. Traductores e intérpretes. Fases de compilación. Optimización de código. Microprocesadores.

La siguiente figura recoge el esquema general de las relaciones existentes entre las asignaturas mencionadas.

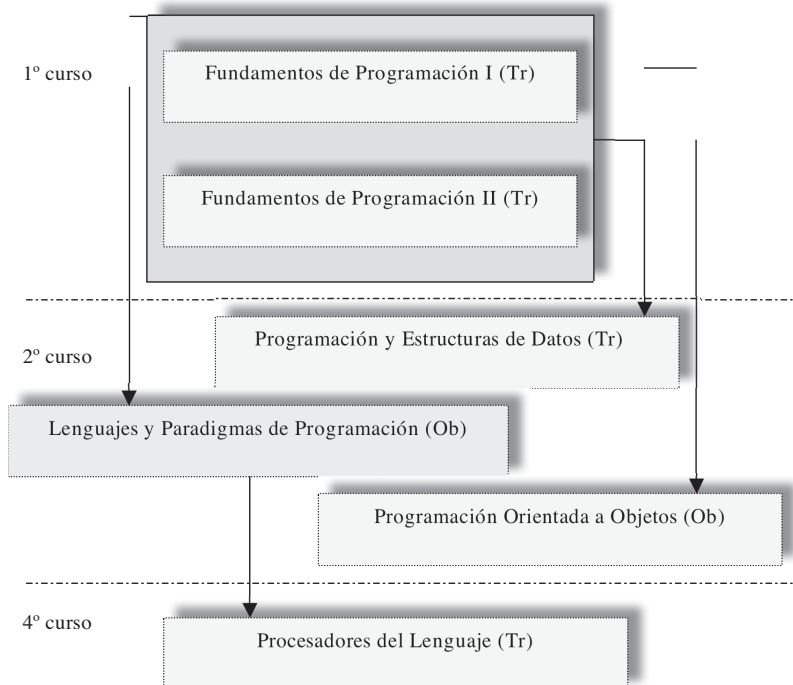


Ilustración 9. relaciones de la asignatura en el plan de estudios

2. OBJETIVOS GENERALES DE LA ASIGNATURA

2.1 CONCEPTUALES (SABER)

- OGC1. Conocer distintos modelos de computación que expliquen la semántica de los lenguajes de programación.
- OGC2. Conocer las distintas herramientas de construcción de abstracciones que proporcionan los lenguajes de programación.
- OGC3. Conocer las características básicas y esenciales de los principales paradigmas de la programación: funcional, procedural, lógico y orientado a objetos.
- OGC4. Conocer un lenguaje de programación funcional.

2.2 PROCEDIMENTALES (SABER HACER)

- OGP1. Saber diseñar barreras de abstracción.
- OGP2. Saber construir, diseñar y utilizar tipos de datos abstractos.
- OGP3. Saber diseñar procedimientos utilizando las características propias de cada paradigma de programación: funcional, procedural, lógico y orientado a objetos.
- OGP4. Saber escribir con fluidez programas en un lenguaje de programación funcional, como Scheme.

2.3 DISPOSICIONALES (SER/ESTAR)

- OGD1. Conocer y adquirir un conjunto de buenas prácticas de programación y ser capaz de argumentar sobre el estilo y la estética de la programación.
- OGD2. Desarrollar el espíritu crítico tanto para enfrentarse a un problema como para evaluar las ventajas e inconvenientes de un diseño concreto.
- OGD3. Apreciar el papel central que juega la abstracción en la tarea de programar.
- OGD4. Adquirir y utilizar con fluidez un buen lenguaje, tanto oral como escrito, para explicar procedimientos escritos en los principales paradigmas de programación (cuál es su funcionamiento, cómo se evalúa una llamada, etc.).

3. CONTENIDOS DE LA ASIGNATURA

La asignatura consta de 11 temas, con una duración de entre 2 y 3 sesiones teóricas y una sesión práctica cada uno. Es interesante resaltar el hecho de que las sesiones teóricas son de una hora de duración (y no de dos horas como suele ser lo habitual), hecho que facilita la asimilación de contenidos del alumno en dos vías: por una parte no llega a cansarse y todo el tiempo está rindiendo al 100% y, por otra parte, ha asentado los conocimientos de la sesión anterior y asi-

mila los nuevos conceptos más fácilmente. La experiencia que los profesores hemos tenido al respecto en esta asignatura es muy positiva, por lo que solicitamos todos los cursos que se mantengan las sesiones teóricas de una hora (dos sesiones de una hora a la semana). Las sesiones prácticas son de dos horas semanales.

3.1 CONTENIDO DE APRENDIZAJE

- Tema 1. Programación Funcional.
- Tema 2. Recursión e Iteración.
- Tema 3. Formulando abstracciones con procedimientos de orden superior.
- Tema 4. Abstracción de datos.
- Tema 5. Datos jerárquicos.
- Tema 6. Macros.
- Tema 7. Construcción de tipos de datos.
- Tema 8. Asignación, estado y entornos.
- Tema 9. Tipos de datos mutables.
- Tema 10. Programación orientada a objetos.
- Tema 11. Programación Lógica.

3.2 DESARROLLO DE LOS TEMAS O UNIDADES DE CONTENIDO

3.2.1 Tema 1. Programación Funcional

Descripción: breve historia de los lenguajes de programación. Explicación razonada sobre el uso del lenguaje de programación Scheme. Primera aproximación al lenguaje de programación Scheme. Conocer el paradigma básico de Programación Funcional, su modelo de computación y la abstracción de procedimientos.

Duración estimada: 2 horas

Contenidos:

- Lenguajes de programación
- Introducción a Scheme
- Expresiones
- Definiciones y entornos
- Evaluación de una expresión
- Definición de nuevas funciones
- Quote
- Algunas funciones de Scheme
- Otros ejemplos de definición de funciones
- Programación Funcional
- Modelo de sustitución
- Orden de evaluación normal vs. de aplicación

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC1.1. Comprender y conocer el Paradigma de la Programación Funcional. • CEC1.2. Conocer el funcionamiento de un programa funcional mediante el modelo de computación de sustitución. • CEC1.3. Conocer el modelo de evaluación normal y de aplicación. • CEC1.4. Aprender y familiarizarse con la sintaxis y estructura básica de programación en Scheme. • CEC1.5. Conocer la estructura básica de un procedimiento recursivo. • CEC1.6. Comprender la relación que se establece entre nombre y dato (o función) mediante el uso de la instrucción “define”.

Cuadro 16. *Análisis de coherencia: competencias conceptuales (saber) del tema 1*

COMPETENCIAS ESPECÍFICAS	
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP1.1. Implementar procedimientos en el Paradigma de la Programación Funcional. • CEP1.2. Explicar por qué un determinado procedimiento pertenece o no al Paradigma de la Programación Funcional. • CEP1.3. Ser capaz de evaluar un procedimiento en orden de evaluación normal. • CEP1.4. Ser capaz de evaluar un procedimiento en orden de aplicación. • CEP1.5. Explicar el funcionamiento de un programa funcional mediante el modelo de computación de sustitución. • CEP1.6. Definir procedimientos sencillos en Scheme. • CEP1.7. Escribir expresiones correctamente en Scheme. • CEP1.8. Implementar procedimientos recursivos sencillos. • CEP1.9. Ser capaz de utilizar nombres adecuados para funciones y datos.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED1.1. Adquirir y utilizar con fluidez un buen lenguaje con el que expresar diseños de software en el Paradigma de la Programación Funcional, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier procedimiento. • CED1.2. Conocer y adquirir un conjunto de buenas prácticas de programación funcional. • CED1.3. Familiarizarse con el intérprete de Scheme: DrScheme y con la metodología de trabajo de un lenguaje interpretado.

Cuadro 17. *Análisis de coherencia: competencias procedimentales (saber hacer) y disposicionales (ser y estar) del tema 1*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
1	1h	<ul style="list-style-type: none"> • Introducción a los Lenguajes de Programación • Introducción a Scheme: expresiones simples, expresiones compuestas, define, quote, if, cond, eval...
2	1h	<ul style="list-style-type: none"> • Introducción a la Programación Funcional: características del paradigma funcional, evaluación por sustitución, orden de evaluación normal y de aplicación

Cuadro 18. *Temporización del tema 1*

Contenidos prácticos (1 sesión de 2 h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Lanza el programa DrScheme y, después de haber cargado el fichero “simply.scm”, escribe cada una de las siguientes instrucciones en el intérprete. ¡¡Piensa en los resultados!! Intenta entender cómo interpreta Scheme lo que escribes.

3	(equal? 'hola "hola")
(+ 1 2)	(first hola)
(+ 1 2 3 4)	(first (bf 'hola))
(+)	(first 234)
(sqrt 25)	(butfirst 234)
(* (+ 2 3) 5)	(+ (first 34) (last 45))
+	(define pi 3.14159)
'+	pi
'hola	'pi
"hola"	(+ pi 7)
(first 'hola)	(* pi pi)
(first "hola")	(define (square x) (* x x))
(butfirst 'hola)	(square 5)
(butfirst "hola")	(square (+ 2 3))

• Ejercicio básico

Rellena los huecos de las siguientes expresiones para obtener el resultado esperado:

```
(if (= 5 (+ 3 2)) (_____ '(+) '(3) '(2)) (+ 3 2))
(+ 3 2)
(if (> 5 (+ 3 2)) '(+ 3 2) (_____ '(+ 3 2)))
5
(eval (append _____ '((+ 3 1))))
5
(if (= 5 (_____ '(+ 3 2))) (_____ '(+) '(3)
'(2)) (eval (_____ '(+ 3) '(2))))
(+ 3 2)
```

• Ejercicio básico

Escribe un procedimiento llamado *quitar-dupl* que, cuando se le pase una palabra como entrada, devuelva el resultado de quitar las letras duplicadas de la palabra. Debería funcionar así:

```
(quitar-dupl 'abcaedeb)
cadeb
(quitar-dupl 'abc)
abc
(quitar-dupl 'aaaabaaa)
Ba
```

- **Ejercicio de ampliación**

Escribe el procedimiento *ordenado* que tome una palabra con dígitos y devuelva cierto si los números están en orden ascendente y falso en otro caso.

- **Ejercicio de ampliación**

Supongamos que tenemos una función mágica llamada *count* que no tiene argumentos. Cada vez que se invoque, devuelvoverá 1 más que la vez anterior, empezando en 1. De modo que, (+ (*count*) (*count*)) devolverá 3. Evalúa:

```
(square (square (count)))
```

en orden **aplicativo** y en orden **normal**. Explica detalladamente tus respuestas.

- **Bibliografía básica:**

Structure and Interpretation of Computer Programs. Abelson & Sussman, the MIT Press. Apartado 1.1 (páginas 1 - 31).

How to Design Programs: An Introduction to Programming and Computing. Felleisen, Findler, Flatt & Krishnamurthi. Apartados 2, 3 y 4.

Simply Scheme: Introducing Computer Science. Brian Harvey. Apartados 1, 2, 3, 4, 5, 6 y 7.

R5RS: Revised(5) Report on the Algorithmic Language Scheme. R. Kelsey, W. Clinger, J. Rees.

- **URL's:**

Dr Scheme (intérprete y documentación): <http://www.drscheme.org/>

Definición de la Programación Funcional en la wikipedia: http://en.wikipedia.org/wiki/Functional_programming

Definición de Abstracción (Computer Science) en la wikipedia: http://en.wikipedia.org/wiki/Abstraction_%28computer_science%29

- **Bibliografía de ampliación**

- The Scheme Programming Language, 2nd ed. R. Ken Dybvig. Prentice Hall. Apartado 1.
- MIT Scheme Manual. Massachusetts Institute of Technology. Marzo 2002.
- Teach Yourself Scheme in Fixnum Days. Dorai Sitanam. 2004.

3.2.2 Tema 2. Recursión e Iteración

Descripción: Razonamiento sobre el uso de la recursión para tener una visión más abstracta de la iteración y por qué los bucles no son imprescindibles para programar. Se profundiza en el diseño de procedimientos recursivos. Se hace especial hincapié en la distinción entre procesos recursivos y procesos iterativos (cuando ambos se implementen con un procedimiento recursivo). Se hace una primera aproximación al concepto de listas en Scheme y funciones de manejo de las mismas. Aplicación de procedimientos recursivos a la implementación de nuevas funciones de manejo de listas.

Duración estimada: 2 horas

Contenidos:

- Confía en la recursión
- El coste espacial de la recursión
- Procesos recursivos e iterativos
- Listas en Scheme

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC2.1. Conocer en profundidad el diseño de procedimientos recursivos. • CEC2.2. Conocer el coste temporal y espacial de un procedimiento recursivo. • CEC2.3. Conocer las diferencias entre procesos recursivos e iterativos. • CEC2.4. Conocer las ventajas e inconvenientes de los procesos recursivos frente a los iterativos. • CEC2.5. Distinguir entre el concepto de procedimiento y de proceso. • CEC2.6. Aprender la sintaxis y las funciones básicas de manejo de las listas en Scheme.

Cuadro 19. *Análisis de coherencia: competencias conceptuales (saber) del tema 2*

COMPETENCIAS ESPECÍFICAS	
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP2.1. Ser capaz de implementar procedimientos recursivos. • CEP2.2. Saber obtener el coste temporal y espacial de procesos recursivos e iterativos. • CEP2.3. Ser capaz de implementar un mismo procedimiento de dos formas distintas: por una parte que genere un proceso recursivo, y por otra que genere uno iterativo. • CEP2.4. Ser capaz de “Confiar en la Recursión”. • CEP2.5. Ser capaz de implementar procedimientos recursivos utilizando listas: recorrido, inserción, sustitución, etc.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED2.1. Ser consciente de la necesidad de “Confiar en la Recursión” para abordar los procedimientos recursivos desde un correcto nivel de abstracción. • CED2.2. Adquirir y utilizar con fluidez un buen lenguaje, tanto oral como escrito, para explicar cualquier procedimiento recursivo. • CED2.3. Adquirir y mejorar las pautas de programación de procedimientos recursivos.

Cuadro 20. *Análisis de coherencia: competencias procedimentales (saber hacer) y disposicionales (ser y estar) del tema 2*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
3	1h	<ul style="list-style-type: none"> • Confía en la recursión. • El coste espacial de la recursión.
4	1h	<ul style="list-style-type: none"> • Procesos recursivos e iterativos. Distinción entre procedimientos y procesos recursivos o iterativos. • Listas en Scheme: list y quote para crear listas, selectores (car, cdr, length, nth), append.

Cuadro 21. *Temporización del tema 2*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Cada uno de los siguientes procedimientos define un método para sumar dos enteros positivos:

```
(define (proc+ a b)
  (if (= a 0)
      b
      (1+ (proc+ (1- a) b))))
```

```
(define (proc+ a b)
  (if (= a 0)
      b
      (proc+ (1- a) (1+ b))))
```

Ilustrar el funcionamiento (con una traza a mano con un ejemplo cada uno) de ambos procedimientos. ¿Los procesos son iterativos o recursivos?

Prueba ahora tus resultados con la traza de Scheme. Para ello escribe la instrucción (require (lib "trace.ss")) para cargar la librería que ejecuta las trazas. Para activar la traza de un procedimiento, escribe (trace procedimiento) y para desactivarla (untrace procedimiento).

• Ejercicio básico

Escribe el procedimiento (sum-elem str) que reciba una cadena de dígitos en str y devuelva la suma de sus dígitos. Escribe **dos versiones**: proceso **recursivo** y proceso **iterativo**.

```
(sum-elem '3021)
6
(sum-elem '0)
0
```

- **Ejercicio de ampliación**

Escribe un procedimiento `biggest` donde reciba una palabra compuesta por dígitos numéricos y devuelva el mayor dígito que hay en esa cadena. Di si tu proceso es recursivo o iterativo y explica por qué.

```
(biggest '3)
3
(biggest '347321)
7
```

- **Ejercicio de ampliación**

Escribe el procedimiento (`truncar elem str`) que tome un elemento y una palabra como argumento, y devuelva la primera parte de la cadena `str` hasta que encuentra el elemento `elem`. Escribe **dos versiones**: proceso **recursivo** y proceso **iterativo**.

```
(truncar 'e 'prueba)
pru
(truncar '3 '4567321 )
4567
```

Bibliografía básica:

Structure and Interpretation of Computer Programs. Abelson & Sussman, the MIT Press. Apartado 1.2 hasta 1.2.4 (páginas 31 - 72).

How to Design Programs: An Introduction to Programming and Computing. Felleisen, Findler, Flatt & Krishnamurthi. Apartado 9, 25.

Simply Scheme: Introducing Computer Science. Brian Harvey. Apartados 11, 12, 13, 14 y 15.

Bibliografía de ampliación

The Scheme Programming Language, 2nd ed. R. Ken Dybvig. Prentice Hall. Apartados 2, 3.

3.2.3 Tema 3. Procedimientos de orden superior

Descripción: en Écheme, las funciones son objetos primitivos, lo que nos permite introducir nuevos elementos de abstracción. La idea fundamental es considerar a las funciones como datos, pudiendo asignarlas a variables, pasarlas como parámetro o incluso devolverlas como resultado desde una función. Este concepto cambia completamente la visión más habitual de los lenguajes de programación procedurales, donde una función es un proceso y hay una clara distinción entre datos y programa. En este tema se profundiza también en la creación de funciones sin un nombre asociado a un entorno (`lambda`).

Duración estimada: 2 horas

Contenidos:

- Funciones como datos de primera clase
- Funciones como argumentos
- Funciones sin nombre
- Tipos de datos de primera clase
- Funciones que devuelven funciones
- Let...

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC3.1. Conocer y apreciar la potencialidad que ofrece Scheme y el resto de lenguajes funcionales a la hora de tratar a las funciones como datos de primer orden. • CEC3.2. Conocer la funcionalidad que ofrece el uso de funciones como argumentos. • CEC3.3. Conocer la funcionalidad que ofrece el uso de funciones sin nombre. • CEC3.4. Conocer la funcionalidad que ofrece el uso de funciones que devuelven funciones. • CEC3.5. Comprender en profundidad la sintaxis y la semántica de lambda. • CEC3.6. Comprender en profundidad la sintaxis y la semántica de let.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP3.1. Diseñar barreras de abstracción utilizando funciones como argumentos. • CEP3.2. Diseñar barreras de abstracción utilizando funciones sin nombre. • CEP3.3. Diseñar barreras de abstracción utilizando funciones que devuelven funciones. • CEP3.4. Implementar procedimientos utilizando lambda. • CEP3.5. Implementar procedimientos utilizando let.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED3.1. Ser consciente de la importancia de las barreras de abstracción mediante procedimientos de orden superior. • CED3.2. Adquirir y utilizar con fluidez un buen lenguaje, tanto oral como escrito, para explicar procedimientos de orden superior.

Cuadro 22. *Análisis de coherencia: competencias específicas del tema 3*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
5	1h	<ul style="list-style-type: none"> • Definición de la forma especial lambda • Funciones sin nombre
L6	1h	<ul style="list-style-type: none"> • Funciones como argumentos • Funciones que devuelven funciones • Definición de la forma especial let

Cuadro 23. *Temporización del tema 3*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Escribe un procedimiento llamado (*generic-intercambio old new*) que tome dos elementos como argumento y a su vez devuelva un procedimiento que tome una lista como argumento, y reemplace cada ocurrencia del primer elemento por el segundo. Ejemplo:

```
(define a-por-b (generic-intercambio 'a 'b))
(a-por-b '(a b c b a))
(b b c b b)

(define seis-por-siete (generic-intercambio '6 '7))
(seis-por-siete '(2 4 6 8 2 4 6))
(2 4 7 8 2 4 7)
```

• Ejercicio básico

Escribe un procedimiento llamado *diff_position* que tome como argumentos 2 listas. Tiene que devolver una lista formada por las diferencias encontradas (elementos que no coincidan en la misma posición) en la segunda lista con respecto a la primera. Ejemplos:

```
(diff_position '(hola soy luisa) '(hola soy maria))
(maria)
(diff_position '(1 2 3 4 5) '(1 2 4 6 8))
(4 6 8)
```

• Ejercicio de ampliación

Rellena los huecos del siguiente trozo de código para obtener los resultados esperados. Explícalo razonadamente.

1. (define y ___)
2. (define x y)
3. (let ((x ___) (p *) (f (lambda (p x y) (___ x ___))))
4. (f ___ x ___))
5. 6
6. x
7. ___
8. y
9. ___

• **Ejercicio de ampliación**

Define un procedimiento llamado *mystery1* que, dada la siguiente llamada, devuelva el resultado esperado:

1. ((*mystery1 mystery1 mystery1*) *mystery1* 10)
2. 10

Escribe ahora un procedimiento llamado *mystery2* que, dada la siguiente llamada, devuelva el resultado esperado:

1. (*mystery2 (mystery2 (mystery2 10 mystery2)*) *mystery2*) *mystery2*)
2. 10

Explica tus respuestas.

Bibliografía básica:

Structure and Interpretation of Computer Programs. Abelson & Sussman, the MIT Press. Apartado 1.3 (páginas 56 - 78).

How to Design Programs: An Introduction to Programming and Computing. Felleisen, Findler, Flatt & Krishnamurthi. Apartados 19, 20, 21, 22.

Simply Scheme: Introducing Computer Science. Brian Harvey. Apartados 8 y 9.

Bibliografía de ampliación:

The Scheme Programming Language, 2nd ed. R. Ken Dybvig. Prentice Hall. Apartado.

3.2.4 Tema 4. Abstracción de datos

Descripción: una de las características fundamentales de los lenguajes de programación es la posibilidad de agregar datos simples para formar estructuras más complejas. Se dan a conocer las ventajas de los datos compuestos (como el incremento de la modularidad de los diseños o la potencialidad del poder de expresión de nuestro lenguaje, y en definitiva de cualquier lenguaje) y se utiliza la forma más sencilla de acumulación de datos de Scheme, la pareja, como base para la creación de estructuras de datos complejas.

Duración estimada: 3 horas

Contenidos:

- Parejas en Scheme
- La agregación de datos no tiene por qué ser primitiva
- Abstracción de datos y barrera de abstracción
- Datos jerárquicos y la propiedad de la clausura
- Tipo abstracto de datos secuencia (o lista)
- Diagramas box-and-pointer

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC4.1. Conocer en profundidad el concepto de parejas en Scheme, su uso y su representación. • CEC4.2. Conocer las ventajas de los datos compuestos. • CEC4.3. Conocer la representación del tipo abstracto lista. • CEC4.4. Aprender a construir listas, y en general, cualquier dato jerárquico, empleando parejas. • CEC4.5. Aprender a usar la abstracción para crear tipos de datos. • CEC4.6. Conocer una de las reglas básicas de programación: acercar el programa al dominio que estamos programando y ocultar la implementación con una barrera de abstracción. • CEC4.7. Conocer la propiedad de la clausura. • CEC4.8. Conocer la representación de los diagramas box-and-pointer.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP4.1. Diseñar e implementar barreras de abstracción utilizando tipos de datos compuestos. • CEP4.2. Dibujar los diagramas box-and-pointer correspondientes a tipos de datos jerárquicos. • CEP4.3. Implementar funciones de manejo del tipo abstracto de datos secuencia o lista.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED4.1. Ser consciente de la importancia de las barreras de abstracción en la construcción de tipos de datos.

Cuadro 24. *Análisis de coherencia: competencias específicas del tema 4*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
7	1h	<ul style="list-style-type: none"> • Parejas en Scheme (cons, car, cdr) • La agregación de datos no tiene por qué ser primitiva • Abstracción de datos y barrera de abstracción
8	1h	<ul style="list-style-type: none"> • Uso de la abstracción • Datos jerárquicos y la propiedad de la clausura
9	1h	<ul style="list-style-type: none"> • Listas y secuencias • Diagramas box-and-pointer

Cuadro 25. *Temporización del tema 4*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

¿Qué devolvería Scheme en las siguientes expresiones? Si el resultado es un *pair*, dibuja el diagrama box and pointer asociado:

1. `((caddr '(1 + * 2)) 3 4)`
2. `(list? (cons 1 (cons 2 (cons nil nil))))`

3. (list (list (list (cons 1 nil))))
4. (list 'one (cons 'foo (list '())))
5. (cddadr '(a ((b)) c d e f) g)

• Ejercicio básico

Supón que estás en un concurso de televisión, donde los puntos conseguidos por los concursantes se guardan en una lista como ésta: ((domingo . 70000) (cristina . 50000) (pepe . 100000)).

a) Diseña el tipo de dato puntuación y su barrera de abstracción. Implementa al menos tres funciones de esta barrera de abstracción.

b) Un buen día, el director de la cadena decide cambiar la forma de almacenamiento de los puntos de la siguiente manera: (domingo 70000 cristina 50000 pepe 100000). Escribe un procedimiento que convierta la lista de los puntos del primer formato al segundo y modifica las funciones de la barrera de abstracción para que funcionen con el segundo formato.

• Ejercicio de ampliación

En clase de teoría has visto que hay dos formas de crear procedimientos, con lambda (sería la forma “verdadera” de la definición de los procedimientos) o sin lambda. Escribe un procedimiento (*con-lambda def*) que tome una definición de un procedimiento sin lambda, y devuelva la misma definición con lambda. Por ejemplo:

1. (con-lambda `(define (square x) (* x x)))
2. (define square (lambda (x) (* x x)))
3. (con-lambda `(define (proc a b) (* a b a)))
4. (define proc (lambda (a b) (* a b a)))

• Ejercicio de ampliación

Define un procedimiento *calculate* que tome una lista con la estructura (número operación número operación... número), donde *más* y *por* son las únicas operaciones permitidas, y calcule el resultado de la operación. Ten en cuenta que *por* tiene mayor precedencia que *mas*. Por ejemplo, (1 mas 2 por 3) se evalúa como $1 + (2 * 3)$ y no como $(1 + 2) * 3$. Se asume que la lista nunca estará vacía. Por ejemplo:

1. (calculate '(2 mas 2))
2. 4
3. (calculate '(1 mas 3 por 4 mas 1))
4. 14

Bibliografía básica:

Structure and Interpretation of Computer Programs. Abelson & Sussman, the MIT Press. Apdo. 2.1 y 2.2.1 (páginas 79—106).

How to Design Programs: An Introduction to Programming and Computing. Felleisen, Findler, Flatt & Krishnamurthi. Apdos. 9, 10, 11, 12.

Simply Scheme: Introducing Computer Science. Brian Harvey. Apdo. 17.

Bibliografía de ampliación

The Scheme Programming Language, 2nd ed. R. Ken Dybvig. Prentice Hall. Apdo. 6.

Tema 5. Datos Jerárquicos

Descripción: en este tema se profundiza en el uso de la primitiva cons (para creación de parejas) para la construcción de objetos jerárquicos tales como pseudoárboles, árboles binarios y árboles genéricos. Se estudian funciones básicas de manejo y funciones adicionales para recorrido. Se aportan ejemplos de utilidad de las estructuras arbóreas: jerarquía (para ciudades, países, ficheros,...), orden (búsqueda en árboles binarios), composición (operaciones aritméticas en árboles genéricos). Se introduce el concepto de recursión mutua para recorrido de árboles genéricos.

Duración estimada: 2 horas

Contenidos:

- Listas jerárquicas
- Árboles binarios
- Árboles genéricos

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC5.1. Conocer el tipo abstracto de datos pseudoárbol, sus constructores y sus selectores. • CEC5.2. Conocer el tipo abstracto de datos árbol binario, sus constructores y sus selectores. • CEC5.3. Conocer el tipo abstracto de datos árbol genérico, sus constructores y sus selectores. • CEC5.4. Conocer el concepto de recursión mutua.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP5.1. Saber implementar procedimientos de recorrido de pseudoárboles. • CEP5.2. Saber implementar procedimientos de recorrido de árboles binarios. • CEP5.3. Saber implementar procedimientos de recorrido de árboles genéricos. • CEP5.4. Saber dibujar los diagramas box-and-pointer de los tipos de datos estudiados en el tema. • CEP5.5. Saber utilizar la recursión mutua en el recorrido de árboles genéricos. • CEP5.6. Ser capaz de determinar, para un problema dado, la estructura jerárquica apropiada.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED5.1. Saber utilizar con fluidez la barrera de abstracción para cada tipo jerárquico de datos. • CED5.2. Entender la utilidad de las barreras de abstracción en los tipos jerárquicos de datos.

Cuadro 26. Análisis de coherencia: competencias específicas del tema 5

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
10	1h	<ul style="list-style-type: none"> Listas jerárquicas Árboles binarios
11	1h	<ul style="list-style-type: none"> Árboles genéricos

Cuadro 27. temporización del tema 5

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• **Ejercicio básico**

a) Escribe el procedimiento (*max-of-pt ptree*), que reciba un pseudoárbol numérico como argumento y devuelva el máximo de todos sus nodos.

- `(max-of-pt '(1 21 3 (42 2) (7 (91 2) (2 (4 62 8))))))`
- 91

b) Escribe ahora el procedimiento (*max-of-tree tree*) para árboles genéricos.

- `tree`
- `(40 (28 (9) (32)) (70 (52) (102)))`
- `(max-of-tree tree)`
- 102

• **Ejercicio básico**

Define el procedimiento (*total-hijos tree*) que vaya recorriendo los nodos del árbol *tree*, calculando el número de hijos de cada nodo y que devuelva total de hijos encontrado en el árbol *tree*. Por ejemplo, el árbol *tree* se corresponde con la siguiente figura:

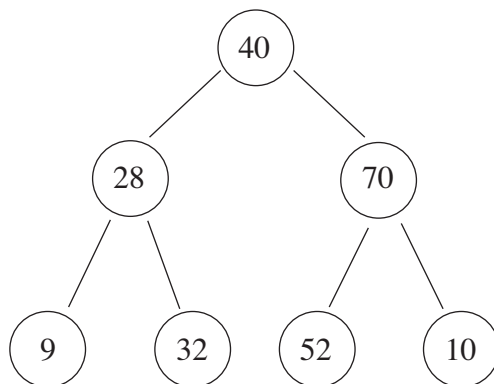


Diagrama 1

El número total de hijos es 6.

1. tree
2. (40 (28 (9) (32)) (70 (52) (10)))
3. (total-hijos tree)
4. 6

• Ejercicio de ampliación

Vamos a considerar que un árbol binario es *estrictamente balanceado* cuando todos sus nodos hoja están a la misma profundidad. Define un procedimiento (*strictly-balanced-btree? btree*) que reciba un árbol binario como parámetro y devuelva *#t* si el árbol es estrictamente balanceado y *#f* si no lo es.

Bibliografía básica:

Abelson & Sussman, *Structure and Interpretation of Computer Programs*, the MIT Press. apartado 2.2.2 (página 107) y 2.3.3 (páginas 151-161)

Brian Harvey, *Simply Scheme: Introducing Computer Science*. Apartados 18 y 19.

Felleisen, Findler, Flatt & Krishnamurthi, *How to Design Programs: An Introduction to Programming and Computing*. Apartados 14, 15, 17.

3.2.5 Tema 6. Macros

Descripción: en este tema se presenta una nueva característica de Scheme que lo distingue del resto de lenguajes de programación. Se trata de la posibilidad de extender el propio lenguaje mediante la definición de macros. Una característica fundamental para implementar esta posibilidad es el uso de un lenguaje de patrones con el que procesar las expresiones.

Duración estimada: 2 horas

Contenidos:

- Introducción
- Conceptos previos
- Funciones con un número variable de argumentos
- Quasiquotation
- Macros en Scheme
- Ejemplos de macros

Competencias específicas:

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC6.1. Conocer las posibilidades que ofrecen las macros (extensiones sintácticas) como una extensión del lenguaje. • CEC6.2. Comprender el concepto de patrón y de plantilla en el diseño de macros. • CEC6.3. Conocer la forma especial de Écheme quasiquote.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP6.1. Implementar funciones con un número variable de argumentos. • CEP6.2. Implementar una macro. • CEP6.3. Expandir una macro. • CEP6.4. Evaluar una llamada a una macro.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED6.1. Adquirir y utilizar con fluidez un buen lenguaje, tanto oral como escrito, para explicar la expansión y la evaluación de macros. • CED6.2. Sentir curiosidad por la utilización de macros y procesamiento de patrones textuales en lenguajes de programación actuales (Perl, Java, PHP, etc.)

Cuadro 28. *Análisis de coherencia: competencias específicas del tema 6*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
12	1h	<ul style="list-style-type: none"> • Introducción a macros. • Conceptos previos.
13	1h	<ul style="list-style-type: none"> • Macros en Scheme: define-syntax, syntax-rules, semántica de las macros. • Ejemplos.

Cuadro 29. *Temporización del tema 6*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

La siguiente definición de *and* es incorrecta:

1. (define-syntax my-and
2. (syntax-rules ()
3. ((_) #t)
4. ((_ e1 e2 ...)
5. (if e1 (my-and e2 ...) #f))))

La expresión (*my-and (not (= x 0)) (/ 1 x)*) debería devolver (*/ 1 x*) cuando *x* no es cero, ya que así es el funcionamiento de *and*, y sin embargo devuelve otro valor. Escribe la expansión de (*my-and (not (= x 0)) (/ 1 x)*) y explica por qué es incorrecta esta versión. A continuación escribe la versión corregida.

• Ejercicio básico

*let** es similar a *let*, pero evalúa sus declaraciones en secuencia, de manera que en una se pueden utilizar los valores declarados en el mismo conjunto de declaraciones (bindings). Ejemplo:

1. (let* ((a 5) (b (+ a a)) (c (+ a b))))
2. (list a b c)
3. (5 10 15)

*let** se podría implementar como expresiones *let* anidadas. Por ejemplo, la expresión *let** anterior se podría expresar como:

1. (let ((a 5))
2. (let ((b (+ a a)))
3. (let ((c (+ a b)))
4. (list a b c))))
5. (5 10 15)

Define *my-let** (para no sobrescribir *let** de Scheme) utilizando *define-syntax*.

• Ejercicio de ampliación

En el siguiente enlace os proporcionamos el código en scheme de un simulador del juego de cartas Twenty-One . Descárgalo y experimenta con él.

A continuación define un procedimiento *stop-at* que reciba un número como argumento y devuelva una estrategia como procedimiento. La estrategia devuelta por *stop-at* debería pedir una nueva carta si y sólo si el total de una mano es menor que el argumento pasado a *stop-at*. Por ejemplo, (*stop-at 16*) devuelve una estrategia que pide otra carta si el total de la mano es menor que 16, pero para si el total alcanza 16. Para probar tu procedimiento, juega unas cuantas veces con el simulador evaluando:

1. (twenty-one hit? (stop-at 16))

Estarás jugando contra un jugador cuya estrategia es parar en 16.

• Ejercicio de ampliación

Define el procedimiento *test-strategy* que evalúe dos estrategias jugando un número de veces con el simulador Twenty-One utilizando las dos estrategias. Deberá devolver el número de veces que ha ganado el *player-1* (y que por tanto ha perdido el *player-2*). Por ejemplo:

1. (test-strategy (stop-at 16) (stop-at 15) 10)

Deberá jugar 10 veces al Twenty-One, utilizando el valor devuelto por (*stop-at 16*) como la estrategia para el *player-1* y el valor de (*stop-at 15*) como la estrategia del *player-2*. Deberá devolver un entero no negativo indicando cuántas veces ha ganado el *player-1*

Bibliografía básica:

R. Ken Dybvig, *The Scheme Programming Language*, 2nd ed. Prentice Hall. Apartados 3 y 8.

R. Ken Dybig, *Chez Scheme User Guide*. Apartado 9.

URL's sobre macros en Scheme:

- <http://schemewiki.org/view/Cookbook/GettingStartedMacros>
- <http://www.scheme.com/tspl3/further.html>
- http://download.plt-scheme.org/doc/301/html/r5rs/r5rs-Z-H-7.html#%25_sec_4.3
- <http://home.comcast.net/~prunesquallor/macro.txt>
- <http://perldoc.perl.org/perlrequick.html>
- <http://perldoc.perl.org/perlintro.html>

Bibliografía de ampliación:

Clinger, W.D. (1991) “Macros in Scheme”. *Lisp Pointers*. IV(4). December, 1991.

Culpepper, R. and Felleisen, M. (2003). “Well-Shaped Macros”. *Scheme Workshop 2003*. Noviembre, 2003.

Dybvig, R.K.; Friedman, D.P. and Haynes, Ch.T. (1988) “Expansion-Passing Style: A General Macro Mechanism”. *LISP and Symbolic Computation*. 1(1). June, 1988.

Kohlbecker, E.; Friedman, D.; Felleisen, M. and Duba, B. (1986) “Hygienic macro expansion”. *Proceedings of the 1986 ACM Conference on LISP and Functional Programming*.

Oleg Kiselyov. “Re-writing abstractions, or Lambda: the ultimate pattern macro”. Diciembre, 2001.

3.2.6 Tema 7. Construcción de tipos de datos

Descripción: los beneficios de introducir tipos de datos en un lenguaje de programación son muy grandes. Entre ellos se encuentran la posibilidad de detectar y prevenir errores en las llamadas a funciones o en la asignación de valores a variables. En este tema se ven tres estrategias para introducir tipos de datos en Scheme: tipos de datos etiquetados, programación dirigida por los datos y paso de mensajes.

Duración estimada: 3 horas

Contenidos:

- Introducción
- Números complejos
- Operaciones con números complejos
- Tipos de datos etiquetados

- Representación de tipos datos con etiquetas
- Programando con números complejos
- Programación dirigida por los datos
- Paso de mensajes
- Estructuras en MzScheme

Competencias específicas:

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC7.1. Conocer las ventajas de incluir tipos de datos en los lenguajes de programación. • CEC7.2. Comprender la necesidad de agregar etiquetas (con el tipo de dato) a los datos en un lenguaje débilmente tipeado como Scheme. • CEC7.3. Conocer diferentes estrategias para introducir tipos de datos en Scheme. • CEC7.4. Conocer las estructuras con campos en Scheme.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP7.1. Saber utilizar la estrategia “datos etiquetados”. • CEP7.2. Saber utilizar la estrategia “programación dirigida por los datos”. • CEP7.3. Saber utilizar la estrategia “paso de mensajes”. • CEP7.4. Conocer la implementación de una tabla hash. • CEP7.5. Construir estructuras de datos en Scheme.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED7.1. Utilizar con fluidez las estrategias para introducir tipos de datos en Scheme. • CED7.2. Adquirir como buena práctica de programación el chequeo de tipos de datos en los argumentos de las funciones.

Cuadro 30. *Análisis de coherencia: competencias específicas del tema 7*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
14	1h	<ul style="list-style-type: none"> • Introducción • Tipos de datos etiquetados
15	1h	<ul style="list-style-type: none"> • Programación dirigida por los datos
16	1h	<ul style="list-style-type: none"> • Paso de mensajes

Cuadro 31. *Temporización del tema 7*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Implementa los tipos de datos *circulo2d* y *rectangulo2d* para las dos estrategias: datos etiquetados (implementando todas las funciones o utilizando programación dirigida por los datos) y paso de mensajes con la siguiente sintaxis:

1. (make-circulo2d x y radio): donde x,y indican las coordenadas del centro del círculo.
2. (make-rectangulo2d x y base altura): donde x,y indican las coordenadas del punto inferior izquierdo del rectángulo.

• Ejercicio básico

a) Añade los siguientes selectores a los tipos de datos anteriores para las estrategias que hayas implementado:

1. (posX obj-geom)
2. (posY obj-geom)
3. (radio obj-geom)
4. (base obj-geom)
5. (altura obj-geom)
6. (tipo obj-geom)

Nota: si se pregunta por el radio de un rectángulo, el selector devolverá el radio del círculo inscrito en el rectángulo, es decir, la diagonal/2. Lo mismo ocurre si se pregunta por la base y la altura del círculo, se devolverá la base y la altura del rectángulo inscrito en él.

b) Para mostrar gráficos por pantalla, Scheme dispone de un paquete de enseñanza llamado draw.ss. Después de añadirlo mediante la opción del menu Lenguaje->Añadir un paquete de enseñanza, implementa el procedimiento *genérico* (*draw-obj obj-geom*) para las estrategias que hayas implementado. Deberá dibujar el *obj-geom* pasado como argumento (del tipo circulo2d o rectangulo2d). Ejemplos de utilización del paquete draw.ss:

1. ;Para abrir el canvas (zona de dibujo) de tamaño 500x500:
2. (start 500 500)
3. ;Para dibujar un círculo en la posición (100 100) de radio 50 y color rojo:
4. (draw-solid-disk (make-posn 100 100) 50 'red)
5. ;Para dibujar un rectángulo en la posición (200 200) con anchura 30 y altura 70 y color azul:
6. (draw-solid-rect (make-posn 200 200) 30 70 'blue)
7. ;make-posn es un tipo de dato predefinido que crea una posición-pasándole las coordenadas X,Y como argumento.

• Ejercicio de ampliación

Implementa, en el juego del Twenty-One, un procedimiento llamado *watch-player* que reciba una estrategia como argumento y devuelva una estrategia. Esta estrategia devuelta deberá hacer lo mismo que la estrategia que se le pasa como argumento, pero además, deberá imprimir toda la información que se le pasa a la estrategia (la mano y la carta del oponente) y la decisión de la estrategia.

Ejemplo:

1. (twenty-one (watch-player (stop-at 16)))
2. (watch-player (stop-at 15)))
- 3.
4. Carta del adversario: 3
5. Total: 5
6. Decisión: Otra carta
7. Carta del adversario: 3
8. Total: 9
9. Decisión: Parar
- 10.
11. Carta del adversario: 5
12. Total: 3
13. Decisión: Otra carta
- 14.
15. Carta del adversario: 5
16. Mi total: 12
17. Decisión: Otra carta
- 18.
19. 2

Bibliografía básica:

- Abelson & Sussman, *Structure and Interpretation of Computer Programs*, the MIT Press. apartado 2.4 (páginas 169-179)
- Felleisen, Findler, Flatt & Krishnamurthi., *How to Design Programs: An Introduction to Programming and Computing*. Apartado 6.

Bibliografía de ampliación

- Tucker, A.; Noonam, R. (2002) *Mc Graw Hill. Programming Languages: Principles and Paradigms*. Apartado 3.

3.2.7 Tema 8. Asignación, Estado y Entornos

Descripción: Si un sistema está compuesto por objetos, éstos difícilmente serán independientes completamente a no ser que se estructuren correctamente y de forma modular. Además cada objeto computacional debería tener sus propias variables de estado local que describen el estado del objeto en cada momento. Al cambiar los estados a lo largo del tiempo, las variables de estado también deben cambiar. De esta forma, que si se desean modelar variables de estado, el lenguaje debe proporcionar operadores de asignación para modificar los valores asociados a las variables. Cuando una asignación está presente, una variable no es sólo considerada como un nombre asociado a un valor, sino que la variable

debe estar asociada a un “lugar” donde los valores se guarden. En el modelo de evaluación de Scheme, esos “lugares” se denominan entornos. Los entornos permiten formalizar también el concepto de ámbito de variable.

En este tema se introduce, por tanto, un nuevo paradigma de computación, el paradigma procedural. Como modelo computacional de este paradigma, se utiliza el citado modelo computacional basado en entornos.

Duración estimada: 4 horas

Contenidos:

- Formas especiales set!
- Entornos, estado global y estado local
- El modelo de computación de Scheme basado en entornos
- Ejemplos de evaluación

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC8.1. Conocer la forma especial set!. • CEC8.2. Conocer el modelo de computación basado en entornos. • CEC8.3. Conocer el concepto de estado local de una función.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP8.1. Saber evaluar una expresión utilizando el modelo computacional basado en entornos. • CEP8.2. Saber implementar procedimientos con estado local en Scheme. • CEP8.3. Ser capaz de extraer la expresión que generó un determinado diagrama de entorno.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED8.1. Aprender a utilizar con fluidez el vocabulario necesario para explicar la evaluación de expresiones utilizando el modelo computacional basado en entornos.

Cuadro 32. *Análisis de coherencia: competencias específicas del tema 8*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
17	1h	<ul style="list-style-type: none"> • Forma especial set! • Entornos y estado local
18	1h	<ul style="list-style-type: none"> • Entornos y estado local
19	1h	<ul style="list-style-type: none"> • El modelo de computación de Scheme basado en entornos
20	1h	<ul style="list-style-type: none"> • Ejemplos de evaluación basado en entornos

Cuadro 33. *Temporización del tema 8*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Modifica cualquier versión de *make-account* para que, cuando se le pase como mensaje *balance*, devuelva el balance actual de la cuenta y, dado el mensaje *init-balance*, devuelva la cantidad con la que la cuenta fue creada. Por ejemplo:

1. (define acc (make-account 100))
2. ((acc 'withdraw) 50)
3. 50

• Ejercicio básico

Modifica *make-account* para que, dado el mensaje *transactions*, devuelva una lista con todas las transacciones que se han realizado desde que la cuenta se abrió. Por ejemplo:

1. (define acc (make-account 100))
2. ((acc 'withdraw) 50)
3. 50
4. ((acc 'deposit) 10)
5. 60
6. (acc 'transactions)
7. ((withdraw 50) (deposit 10))

• Ejercicio de ampliación

Implementa un procedimiento llamado *parkímetro* que simule una máquina de control de parking con las siguientes consideraciones:

Se paga con tarjeta de crédito y el importe lo elige el usuario. Cuesta 0.01€/minuto.

El medidor es digital. El usuario puede consultar el tiempo que le queda. Proporciona una manera para poder consultarlo.

Al dueño del parking le interesa saber cuánto dinero ha recogido un determinado parkímetro. Proporciona una manera para devolver la cantidad total recaudada.

Cuando el tiempo ha llegado a 0, un mensaje llamado “multar” debería devolver #t. Se utiliza para saber quién necesita un nuevo ticket.

El tiempo se decrementa manualmente con un mensaje llamado “tick” (simula el tick de reloj).

• Ejercicio de ampliación

Dibuja el siguiente diagrama de entorno y explícalo detalladamente:

1. (define x 2)
2. (define (changer x y)
3. (y)
4. (y)
5. x)
6. (changer 16 (lambda () (set! x (* x x))))

Bibliografía básica:

Abelson & Sussman, *Structure and Interpretation of Computer Programs* MIT Press. Apartado 3.1 (páginas 218-229) y 3.2 (páginas 236-249)
 Felleisen, Findler, Flatt & Krishnamurthi, *How to Design Programs: An Introduction to Programming and Computing*. Apartados 34, 35, 36, 37.

Bibliografía de ampliación:

Tucker, A.; Noonam, R. (2002) *Programming Languages: Principles and Paradigms*. Mc Graw Hill. Apartado 4.

3.2.8 Tema 9. Tipos de datos mutables

Descripción: el modelado de sistemas compuestos por objetos cuyos estados cambian en el tiempo, obliga al diseño de abstracciones de datos que incluyan, además de los selectores y constructores, unos operadores llamados mutadores. Los mutadores modifican los objetos, y los objetos a los que se aplican los mutadores se denominan objetos mutables.

Además de los mutadores anteriores, también es necesario introducir la posibilidad de modificar las estructuras de datos jerárquicas únicamente cambiando enlaces y contenidos de las mismas.

En este tema se introducen las primitivas de Scheme `set!`, `set-car!` y `set-cdr!` que permiten realizar estas operaciones sobre estructuras jerárquicas estudiadas en temas anteriores.

Duración estimada: 2 horas

Contenidos:

- Primitivas `set-car!` y `set-cdr!`
- Tipos abstractos de datos con datos mutables
- Listas. Árboles
- Tablas hash

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC9.1. Conocer las primitivas <code>set-car!</code> y <code>set-cdr!</code>. • CEC9.2. Conocer la diferencia entre igualdad de referencia e igualdad de contenido. • CEC9.3. Conocer la aplicación de los mutadores a los tipos abstractos de datos.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP9.1. Saber diseñar las primitivas <code>set-car!</code> y <code>set-cdr!</code>. • CEP9.2. Saber dibujar el diagrama box-and-pointer de un tipo de dato mutable. • CEP9.3. Saber implementar procedimientos mutadores de tipos de datos jerárquicos.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED9.1. Adquirir y mejorar destrezas en el diseño de procedimientos mutadores, ya que es necesaria una visión abstracta, en ocasiones elevada, para crear un buen diseño de un mutador.

Cuadro 34. Análisis de coherencia: competencias específicas del tema 9

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
21	1h	Primitivas set-car! y set-cdr!. Manejo de listas con mutadores.
22	1h	Manejo de árboles con mutadores. Manejo de tablas hash con mutadores.

Cuadro 35. *Temporización del tema 9*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Tenemos el siguiente procedimiento para añadir listas:

1. (define (append x y)
2. (if (null? x)
3. y
4. (cons (car x) (append (cdr x) y))))

append forma una lista nueva formando sucesivamente nuevas parejas de elementos de x con elementos de y. El procedimiento append! es similar a append, pero es un mutador, más que un constructor. Añade las listas modificando el final de la última pareja de x para que su cdr sea ahora y (Sería un error llamar a append! con un x vacío).

1. (define (append! x y)
2. set-cdr! (last-pair x) y)
3. x)

Donde last-pair es un procedimiento que devuelve la última pareja de su argumento:

1. (define (last-pair x)
2. (if (null? (cdr x))
3. x
4. (last-pair (cdr x))))

Considera las siguientes instrucciones:

1. (define x (list 'a 'b))
2. (define y (list 'c 'd))
3. (define z (append x y))
4. z -> (a b c d)
5. (cdr x) -> _____
6. define w (append! x y)
7. w -> (a b c d)
8. (cdr x) -> _____

Rellena los huecos y a continuación dibuja diagramas de caja y puntero para explicar las respuestas.

• Ejercicio básico

Escribe el procedimiento (`remove-dupls! ls`) que tome una lista como argumento y la mute de forma que se eliminen sus elementos duplicados

1. `(define ll (list 1 2 3 5 4 3 5 2 3 4 1))`
2. `(remove-dupls! ll)`
3. `ll`
4. `(5 2 3 4 1)`

• Ejercicio de ampliación

Escribe el procedimiento (`deep-subst! ls old new`) que tome una lista y dos elementos como argumento, y mute la lista de forma que cada ocurrencia de `old` se sustituya por `new`.

1. `lista`
2. `(1 (2 3 4 (4 5) (3 (5 6)) 4))`
3. `(deep-subst! lista '4 'A)`
4. `lista`
5. `(1 (2 3 A (A 5) (3 (5 6)) A))`

• Ejercicio de ampliación

Escribe el procedimiento (`reverse-list! ls`) que tome una lista como argumento, y mute la lista de forma que quede invertida.

1. `lista`
2. `(1 (2 3 4 (4 5) (3 (5 6)) 4))`
3. `(reverse-list! lista)`
4. `lista`
5. `(4 ((6 5) 3) (5 4) 4 3 2) 1)`

Bibliografía básica:

Abelson & Sussman, *Structure and Interpretation of Computer Programs*, the MIT Press. Apartado 3.3 (páginas 251-266)

Felleisen, Findler, Flatt & Krishnamurthi, *How to Design Programs: An Introduction to Programming and Computing*. Apartados 39, 40, 41.

Bibliografía de ampliación

Tucker, A.; Noonam, R. (2002) *Programming Languages: Principles and Paradigms*. Mc Graw Hill. Apartado 5.

3.2.9 Tema 10. Programación Orientada a Objetos

Descripción: introducción en Scheme a un paradigma de programación que amplía las posibilidades de abstracción mediante el uso de clases, objetos, interfaces y herencia. La Programación Orientada a Objetos (POO) se dispone en Scheme como un módulo adicional, donde se están escribiendo programas en Scheme, pero con el vocabulario extendido de la POO: clases, herencia, campos, etc. Este módulo se incorpora como un estándar en la implementación DrScheme que se utiliza en la asignatura.

Las ideas clave que se estudian en este tema son el paso de mensajes (un objeto puede pedir a otro la realización de ciertas tareas), el estado local (un objeto puede recordar cosas sobre su historia pasada) y la herencia.

Duración estimada: 3 horas

Contenidos:

- Breve historia de la Programación Orientada a Objetos
- Conceptos básicos de Programación Orientada a Objetos
- Características avanzadas del módulo class.ss
- El modelo de entornos y las clases

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC10.1. Conocer el Paradigma de Programación Orientada a Objetos. • CEC10.2. Conocer los conceptos de clase, campo, método, interfaz y objeto. • CEC10.3. Conocer el mecanismo de abstracción de la herencia. • CEC10.4. Conocer los distintos modificadores de los métodos en una relación de herencia. • CEC10.5. Conocer el concepto de estado local de un objeto.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP10.1. Ser capaz de diseñar una jerarquía de clases e interfaces utilizando el módulo de Scheme class.ss. • CEP10.2. Saber crear interfaces, clases, campos, campos de inicialización, métodos y objetos. • CEP10.3. Saber enviar mensajes entre objetos. • CEP10.4. Saber utilizar el mecanismo de herencia.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED10.1. Capacidad de aplicación de los conocimientos, métodos y destrezas del paradigma orientado a objetos al diseño de soluciones a problemas sencillos. • CED10.2. Adquirir y utilizar con fluidez un buen lenguaje de diseño software, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier interacción o relación entre elementos del sistema.

Cuadro 36. *Análisis de coherencia: competencias específicas del tema 10*

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
23	1h	Breve historia de POO Elementos básicos de POO
24	1h	Características del módulo class.ss Herencia
25	1h	Interfaces Funciones de bajo nivel de acceso a los campos Definición de clases y modelo de entornos

Cuadro 37. *Temporización del tema 10*

Contenidos prácticos (1 sesión de 2h.)

Ejemplos de ejercicios básicos y de ampliación.

• Ejercicio básico

Define una interfaz dibujable que defina un método llamado dibuja. Define la clase objeto_grafico que implemente la interfaz dibujable. Tiene las siguientes características:

El campo de inicialización identificador almacena el nombre del objeto gráfico (en este caso será objeto_grafico).

El campo color almacena el color del borde del objeto gráfico.

El método dibuja de la interfaz dibujable. En el canvas deberá dibujar la cadena de texto que contiene su identificador. Para ello puedes utilizar la función (draw-solid-string Posn String) del paquete de enseñanza draw.ss.

• Ejercicio básico

Define 2 subclases de la clase objeto_ gráfico. Cada clase debería representar un tipo distinto de figura, como por ejemplo círculo, rectángulo o triángulo. Todas las clases tendrán las siguientes características comunes:

Los campos de inicialización posX y posY que almacenan las coordenadas del objeto.

Los métodos (area) devuelven el área de la figura y (traslada incX incY), que modifica sus coordenadas realizando una traslación con respecto al origen de coordenadas.

El método (bounding-box),. que devuelve el objeto rectángulo que incluye de forma estricta la figura.

Cada subclase debe sobrescribir el método dibuja de la clase padre, de forma que, utilizando las funciones del paquete de enseñanza draw.ss, dibujen cada objeto en el canvas, con el color definido en el campo color de la clase padre. Si el campo color estuviese vacío, utilizar el color por defecto (para ello no poner el último argumento en la llamada a la función de draw.ss).

- **Ejercicio de ampliación**

Define la clase grupo-objetos para que se puedan definir objetos compuestos. Para ello, la clase va a tener un método añade-objeto que deben ser guardados en alguna variable de instancia del nuevo objeto compuesto recién creado. Define el método (desagrupa) que devuelve una lista con todos los objetos concretos que forman un objeto compuesto.

- **Ejercicio de ampliación**

Añade a la clase grupo-objetos el método (bounding-box) que devuelva el rectángulo, que incluye de forma estricta el objeto compuesto.

Bibliografía básica:

Tutorial sobre POO en MzScheme:

<http://schemecookbook.org/Cookbook/IntroductionToMzlibClasses>

Timothy Budd (2001) *An Introduction to Object Oriented Programming*. Ed. Addison Wesley. Apartados 1, 2, 8.

Bibliografía de ampliación

Friedman, D.P. (2003) “Object-Oriented Style”. *International LISP Conference*. October, 2003.

Jonathan A. Rees and Norman I. Adams IV. (1988) “Object-oriented programming in Scheme”. *Proceedings of the 1988 ACM Conference on Lisp and Functional Programming*.

3.2.10 Tema 11. Programación Lógica

Descripción: se utiliza un módulo de Scheme de programación lógica implementada por Abelson & Sussman. Este módulo tiene algunas notaciones diferentes a los lenguajes de programación lógica estándares, como Prolog, pero el concepto es el mismo.

Duración estimada: 2 horas

Contenidos:

- Breve historia de la Programación Lógica
- Conceptos básicos de la Programación Lógica: reglas, aserciones, preguntas, relaciones, base de conocimientos
- Soluciones múltiples, backtracking, query system (unificación y pattern matching)
- Funcionamiento del query system

COMPETENCIAS ESPECÍFICAS	
Conceptuales (saber)	<ul style="list-style-type: none"> • CEC11.1. Conocer el Paradigma de la Programación Lógica. • CEC11.2. Conocer los conceptos de reglas, preguntas, base de conocimientos, relaciones, aserciones. • CEC11.3 Conocer el funcionamiento del query system.
Procedimentales (saber hacer)	<ul style="list-style-type: none"> • CEP11.1. Saber implementar y modificar un programa mediante el Paradigma de la Programación Lógica. • CEP11.2. Saber cómo funciona el mecanismo del query system (unificación y pattern matching), backtracking. • CEP11.3. Saber cómo extraer soluciones múltiples.
Disposicionales (ser/estar)	<ul style="list-style-type: none"> • CED11.1. Manejar con fluidez los conceptos de la Programación Lógica.

Cuadro 38. Análisis de coherencia: competencias específicas del tema 11

TEMPORIZACIÓN		
Sesión	Duración	Contenidos
23	1h	Historia de la Programación Lógica Conceptos básicos de la Programación Lógica: reglas, aserciones, preguntas, relaciones, base de conocimientos
24	1h	Soluciones múltiples, backtracking, query system (unificación y pattern matching) Query system

Cuadro 39. Temporización del tema 11

Bibliografía básica:

Abelson & Sussman, *Structure and Interpretation of Computer Programs*, the MIT Press. Apartado 4.4 (páginas 438-468)

Bibliografía de ampliación:

Alferes, J.J, y Pereira, L.M (1996) *Reasoning with Logic Programming*. Springer-Verlag.

Amble, Tore (1987) *Logic Programming and Knowledge Engineering*. Ed. Addison-Wesley.

Apt, K. R. (1990) “Introduction to Logic Programming”. En J. van Leeuwen, *Handbook Of Theoretical Computer Science*. Vol B: Formal Models and Semantics. Elsevier, Amsterdam y The MIT Press, Cambridge.

4. METODOLOGÍA DOCENTE

Las actividades que se proponen son:

Clases de teoría con apoyo de material audiovisual: las explicaciones teóricas se refuerzan con ejemplos prácticos. El profesor dispone de un ordenador portátil en clase, conectado a un proyector. Mientras explica los conceptos teóricos, los ilustra con pequeños programas de ejemplo. De esta forma, el alumno puede entender y asimilar mejor lo que se está explicando.

Prácticas de laboratorio: se realiza una práctica semanal con los contenidos del temario que se ha explicado en esa semana. Constan de ejercicios cortos que el alumno puede resolver en las dos horas del turno de prácticas. Las prácticas tienen dos partes, una parte básica que se considera imprescindible para la consecución de los objetivos y una parte de ampliación, para aquellos alumnos que tengan un interés especial en la materia o quieran aumentar su nota.

Actividades en grupos pequeños / tutorías docentes: estas actividades estarán relacionadas con la realización de problemas y cuestiones teórico-prácticas vinculadas con la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos estudiados.

Tutorías de atención al alumnado: el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como dudas concretas sobre el contenido de la asignatura.

4.1 ESTRATEGIAS DE APRENDIZAJE

Los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador no son los únicos medios sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web ofrecen innumerables posibilidades que no hay que dejar pasar. Se ha elaborado una web de la asignatura que incluye toda la información que el alumno necesita.

Entre otras cosas, en dicha página se puede encontrar:

- **Novedades:** desde aquí, el alumno puede estar perfectamente informado de cualquier novedad relacionada con la asignatura, además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, o novedades.
- **Tutorías:** indica el horario de atención al alumnado.
- **Objetivos:** resume los objetivos que se pretenden alcanzar en esta asignatura.
- **Temario:** especifica el temario de esta asignatura.
- **Apuntes de la asignatura:** aquí se puede encontrar todo el material que, como mínimo, va a ser necesario para el seguimiento de las clases teóricas.
- **Ejemplos de clase:** enlace a todos los ejemplos que el profesor de teoría presenta en clase. Están disponibles para que los alumnos muestren una mayor participación en clase y no estén pendientes de copiar el código.
- **Prácticas de la asignatura:** material necesario para el seguimiento de las clases prácticas.

- **Bibliografía complementaria:** aparece un listado de bibliografía complementaria que pueden consultar para profundizar en la asignatura o preparar los trabajos complementarios.
- **Horarios de teoría y prácticas:** contiene información sobre los grupos de teoría y prácticas, así como el profesorado que los imparte.
- **Exámenes:** el alumno dispone de los exámenes de otros años para tener mayor variedad de ejercicios.
- **Enlaces de interés:** aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en los contenidos de la materia.
- **Recursos:** enlaces al software utilizado en la asignatura (DrScheme), etc.
- **Foro de la asignatura:** enlace al foro de la asignatura, donde se establece una relación entre profesores y alumnos de forma casi constante. Pretende ser una herramienta de consulta entre el alumnado y el profesorado. El alumno puede consultar, preguntar, responder y sugerir sobre cualquier tema relacionado con la asignatura. Es una herramienta muy positiva que estimula el interés del alumno por la asignatura por varios motivos:

Si tiene alguna duda puntual, puede preguntar en el foro y cualquier alumno o profesor puede responder con relativa rapidez, de forma que en cuestión de minutos (o pocas horas) puede continuar con su ejercicio, y no tener que esperar varios días a ver al profesor en tutorías.

Si tiene alguna duda que le ha surgido a otro alumno previamente, puede leer la explicación pertinente y de esta forma solventar el problema de forma inmediata.

Si lee una pregunta lanzada por otro compañero y conoce la respuesta, puede escribir la solución al problema permitiéndole mejorar su vocabulario al respecto y afianzar sus conocimientos, con la satisfacción de saber que su respuesta será leída por la mayoría de sus compañeros y por el profesorado.

Le permite pensar en nuevos ejercicios y en su solución, estableciéndose competiciones entre diversos alumnos y argumentaciones sobre qué solución es mejor y por qué.

En base a todo esto, la estrategia de aprendizaje que se propone, se compone de las siguientes fases:

Planificación de las clases teóricas:

Como se ha explicado al principio del capítulo 7.2 de este proyecto docente, las clases teóricas son de una hora de duración.

Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar contrastarlo con otros compañeros (de forma personal o utilizando el foro de la asignatura) o utilizar la bibliografía recomendada. Si esto no es suficiente se acudirá a tutorías para intentar solucionar el problema.

Planificación de las clases prácticas:

Se realizarán 10 prácticas guiadas con ejercicios cortos y pequeños programas en Scheme sobre el temario de teoría. Estas prácticas reforzarán los conceptos allí vistos. Las prácticas se deberán realizar por parejas por varios motivos, por un lado, para poder corregir todas las prácticas de forma presencial en la segunda parte de la sesión de prácticas y, por otra parte, porque el alumno aprende a trabajar en equipo y resulta conveniente que los alumnos comenten y analicen los métodos a seguir y las posibles soluciones.

Las sesiones prácticas se dividen en dos partes. En una primera parte, el profesor explicará detalladamente la práctica a realizar en esa sesión y resolverá las dudas que surjan. En una segunda parte, mientras los alumnos realizan la práctica, el profesor irá pasando por todos los puestos revisando y corrigiendo la práctica lanzada en la sesión anterior.

Las prácticas están diseñadas para que se puedan realizar en el transcurso de las dos horas de laboratorio. No obstante, si el alumno necesita más tiempo para su consecución, puede terminarla en horas no presenciales. El profesorado corregirá la práctica en la siguiente sesión de prácticas y realizará una corrección individual para cada pareja, indicando cuáles han sido los errores cometidos y explicando cómo sería la solución correcta. La corrección consiste en revisar el funcionamiento de los ejercicios propuestos y hacer preguntas sobre los mismos, de forma que el alumno defienda su trabajo y perfeccione su vocabulario sobre los contenidos de la asignatura.

Planificación de las actividades en grupos pequeños:

Una vez entendidas las explicaciones de las clases teóricas, se leerá la actividad a realizar en grupos pequeños para poder preguntar las dudas surgidas en el entendimiento del enunciado.

En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida en la propia aula entre todos.

Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Evaluación final: El estudiante deberá realizar el examen de teoría y, si en base a los criterios de evaluación ha aprobado, habrá superado la asignatura.

5. PLAN DE APRENDIZAJE DEL ALUMNO

En las siguientes tablas se esquematiza el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial, y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

5.1 Planificación del programa de Teoría

ACTIVIDAD	Horas Presenciales		Horas No Presenciales	TOTAL
	Trabajo en aula	Tutorías organizadas	Aprendizaje autónomo y colaborativo	
Tema 1	2	0	0	2
Tema 2	2	1	2	5
Tema 3	2	1	2	5
Tema 4	3	1	3	7
Tema 5	2	1	2	5
Tema 6	2	1	2	5
Tema 7	3	1	3	7
Tema 8	4	1	3	8
Tema 9	2	1	2	5
Tema 10	3	1	2	6
Tema 11	2	1	2	5
Preparación del examen final:	2		15	17
Examen final:	2		0	2
TOTAL:	31	10	38	79

Cuadro 40. Planificación del programa de teoría

5.2 Planificación del programa de Prácticas

ACTIVIDAD	Prácticas en aula (Horas Presenciales)	Realización de las prácticas fuera del horario de la asignatura (Horas no Presenciales)	TOTAL
Práctica 1	2	0	2
Práctica 2	2	1	3
Práctica 3	2	1	3
Práctica 4	2	1	3
Práctica 5	2	3	5
Práctica 6	2	3	5
Práctica 7	2	2	4
Práctica 8	2	1	3
Práctica 9	2	3	5
Práctica 10	2	1	3
TOTAL:	20	16	36

Cuadro 41. Planificación del programa de prácticas

5.3 Planificación temporal Teoría y Prácticas

SEMANA	Sesión Teoría	Sesión Prácticas
1 2	Tema 1	Inscripción a Prácticas Explicación Práctica
3	Tema2	Explicación Práctica 2 Corrección Práctica 1
4	Tema 3	Explicación Práctica 3 Corrección Práctica 2
5 6	Tema 4	Explicación Práctica 4 Corrección Práctica 4
7	Tema 5	Explicación Práctica 5
8	Tema 6	Explicación Práctica 6 Corrección Práctica 5
9 10	Tema 7	Explicación Práctica 7 Corrección Práctica 6 Corrección Práctica 7
11 12	Tema 8	Explicación Práctica 8 Corrección Práctica 8
13	Tema 9	Explicación Práctica 9
14	Tema 10	Explicación Práctica 10 Corrección Práctica 9
15	Tema 11	Corrección Práctica 10

Cuadro 42. Planificación temporal de teoría y prácticas

5.4 Planificación (resumen)

Tras realizar la planificación de la asignatura obtenemos los siguientes resultados:

- Número total de horas presenciales: 61
- Número total de horas no presenciales: 54
- Número total de horas: 115

6. BIBLIOGRAFÍA

6.1 BIBLIOGRAFÍA BÁSICA

Abelson & Sussman, *Structure and Interpretation of Computer Programs*, the MIT Press. Apartado 1.1 (páginas 1 - 31).

Brian Harvey, *Simply Scheme: Introducing Computer Science*.

Timothy Budd (2001) *An Introduction to Object Oriented Programming*. Ed. Addison Wesley.

6.2 BIBLIOGRAFÍA COMPLEMENTARIA

Tucker, A.; Noonam, R. (2002) *Programming Languages: Principles and Paradigms*. Mc Graw Hill.

Felleisen, Findler, Flatt & Krishnamurthi, *How to Design Programs: An Introduction to Programming and Computing*.

Dybvig, R.K., *The Scheme Programming Language*, 2nd ed. Prentice Hall.

6.3 OTROS RECURSOS

Foro de debate:

- Foro de debate de Lenguajes y Paradigmas de Programación en la Universidad de Alicante: <http://www.rvg.ua.es/moodle>

Libros online:

- Structure and Interpretation of Computer Programs. Edición online. <http://mitpress.mit.edu/sicp/full-text/book/book.html>
- How to Design Programs. Edición online. <http://www.htdp.org/>
- The Scheme Programming Language, Second Edition. Edición online. <http://www.scheme.com/tspl2d/index.html>
- Chez Scheme User's Guide. Edición online. <http://www.scheme.com/csug/index.html>

Asignaturas similares a LPP (contenidos y estructura):

- CS61A. Structure and Interpretation of Computer Programs. Asignatura de la Universidad de Berkeley con un estilo muy similar a LPP. <http://www-inst.eecs.berkeley.edu/~cs61a/sp06/>
- 6.001. Structure and Interpretation of Computer Programs. Asignatura del MIT con una estructura muy similar a LPP. <http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-001Spring-2005/CourseHome/>

Scheme:

- PLT Online. Recopilación de textos online sobre Lenguajes de Programación. <http://homepages.cwi.nl/~atanasso/ref>
- Scheme Cookbook. Documentación sobre Scheme, librerías y módulos. <http://schemecookbook.org/Cookbook>
- Lambda the Ultimate: The Programming Languages Weblog. <http://lambda-the-ultimate.org>

Sitios donde se distribuyen librerías para PLT Scheme:

- PLT: <http://www.plt-scheme.org/software>
- PLanetT: <http://planet.plt-scheme.org>
- Schematics: <http://sourceforge.net/projects/schematics>
- Scheme Cookbook Library Index: <http://schemecookbook.org/Cookbook/LibraryIndex>
- Neil Van Dyke: <http://www.neilvandyke.org>
- www.cs.utah.edu/plt/develop/
- Bibliography of Scheme-related Research: <http://library.readscheme.org>

Historia de los lenguajes de programación:

- Computer Languages History: <http://www.levenez.com/lang/>
- The History of Computer Programming Languages: http://www.princeton.edu/~ferguson/adw/programming_languages.shtml
- History of Programming Languages: <http://www.cs.iastate.edu/~leavens/ComS541Fall97/hw-pages/history/>
- The Development of the C Language: <http://cm.bell-labs.com/cm/cs/who/dmr/chist.html>
- Historic Documents in Computer Science: <http://www.fh-jena.de/~kleine/history/>
- Reporte del ALGOL-60 publicado en ACM: <http://www.fh-jena.de/~kleine/history/languages/Algol60-ACM.pdf>
- Computer museum tonh <http://www.tonh.net/museum/rsbooksproglang.html>

7. Evaluación de los procesos y resultados de aprendizaje

Una asignatura como la que nos ocupa resulta compleja de evaluar con los patrones tradicionales. Es conveniente realizar una evaluación continua que permita conocer el grado de conocimientos que los estudiantes van adquiriendo a lo largo del proceso educativo. En este sentido, dividimos la evaluación de la asignatura en dos módulos, uno de ellos mediante la realización, al final del curso, de un examen, y otro como consecuencia de una evaluación continua en la realización de tareas teórico-prácticas. Las prácticas se evalúan de forma continua y pretenden premiar al alumno que trabaja semana a semana. De esta forma que establece un sistema de puntos, que el alumno va sumando semanalmente. Por este motivo, los puntos de las prácticas sólo se pueden conseguir de esta forma, es decir, semana a semana y no se podrán modificar para las siguientes convocatorias. Las prácticas no son obligatorias pero son convenientes, ya que un alumno que realice las prácticas, aparte de conseguir puntos que le beneficiarán, habrá aprendido los contenidos necesarios para el examen final.

Bajo estos condicionantes, se propone el siguiente esquema de evaluación:

- Evaluación de la Parte Teórica (50 puntos). Prueba escrita compuesta de cuestiones cortas teórico-prácticas con un estilo similar a las prácticas realizadas durante el curso.
- Evaluación de la Parte Práctica (10 puntos). Suma de las calificaciones obtenidas semanalmente en las prácticas de laboratorio. Cada una de las 10 prácticas se evaluará con una puntuación entre 0 y 1, con una granularidad de 0.25 (las notas posibles serán: 0, 0.25, 0.5, 0.75 y 1). Por tanto, la suma de todas las prácticas será como máximo de 10 puntos
- La nota final se obtiene sumando los puntos obtenidos en prácticas (máximo de 10) y los obtenidos en teoría (máximo de 50). Esta puntuación de 0 a 60 se escalará al rango 0.0-10.0 para obtener la calificación final. El alumno que consiga 30 puntos o más habrá superado la asignatura.

8. Análisis de coherencia de la guía docente

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Lenguajes y Paradigmas de Programación. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Conceptuales (saber)			
OGC1	CEC1.2 CEC1.3 CEC8.2	1 8	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos los modelos de computación.

Cuadro 43. Análisis de coherencia: competencias conceptuales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Conceptuales (saber)			
OGC2	CEC1.6 CEC3.1 CEC3.2 CEC3.3 CEC3.4 CEC4.1 CEC4.2 CEC4.3 CEC4.4 CEC4.5 CEC4.6 CEC4.7 CEC4.8 CEC5.1 CEC5.2 CEC5.3 CEC5.4 CEC7.1 CEC7.2 CEC7.3 CEC8.3 CEC9.3 CEC10.3	1 3 4 5 7 8 9 10	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a las herramientas de construcción de abstracciones que proporcionan los lenguajes de programación.
OGC3	CEC1.1 CEC2.1 CEC2.2 CEC2.3 CEC2.4 CEC2.5 CEC6.1 CEC6.2 CEC10.1 CEC10.2 CEC10.3 CEC10.4 CEC10.5 CEC11.1 CEC11.2 CEC11.3	1 2 6 10 11	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los distintos paradigmas de programación.
OGC4	CEC1.5 CEC2.6 CEC3.5 CEC3.6 CEC6.3 CEC9.1 CEC9.2	1 2 3 6 9	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Prácticas. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los contenidos teóricos en el lenguaje de programación Scheme.

Cuadro 44. *Análisis de coherencia: competencias conceptuales (saber)*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Procedimentales (saber hacer)			
OGP1	CEP1.3 CEP1.4 CEP1.5 CEP8.1 CEP8.3	1 8	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza de los conceptos relativos a los modelos de computación.
OGP2	CEP1.9 CEP3.1 CEP3.2 CEP3.3 CEP3.4 CEP4.1 CEP4.2 CEP4.3 CEP5.1 CEP5.2 CEP5.3 CEP5.4 CEP5.5 CEP5.6 CEP7.1 CEP7.2 CEP7.3 CEP9.2 CEP9.3	1 3 4 5 7 9	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza en la aplicación de los conceptos relativos a las herramientas de construcción de abstracciones que proporcionan los lenguajes de programación.
OGP3	CEP1.1 CEP1.2 CEP2.1 CEP2.2 CEP2.3 CEP2.4 CEP2.5 CEP6.2 CEP6.3 CEP6.4 CEP7.4 CEP7.5 CEP10.1 CEP10.2 CEP10.3 CEP10.4 CEP11.1 CEP11.2 CEP11.3	1 2 6 7 10 11	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza sobre los distintos paradigmas de programación.

Cuadro 45. Análisis de coherencia: competencias procedimentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Procedimentales (saber hacer)			
OGP4	CEP1.6 CEP1.7 CEP1.8 CEP3.5 CEP3.6 CEP6.1 CEP8.2 CEP9.1	1 3 5 8 9	Enseñanza presencial (lección magistral / Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas).	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza sobre el lenguaje de programación Scheme.

Cuadro 46. Análisis de coherencia: competencias procedimentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Disposicionales (ser/estar)			
OGD1	CED1.2 CED1.3 CED2.1 CED2.3 CED3.1 CED4.1 CED6.2 CED7.1 CED10.1	1 2 3 4 6 7 10	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Corrección individualizada. Nivel de actuación en las destrezas de las buenas prácticas de programación.
OGD2	CED2.3 CED4.1 CED5.1 CED6.2 CED7.1 CED11.1	2 4 5 6 7 11	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Corrección individualizada. Nivel de actuación frente a un problema y ventajas y análisis de un diseño determinado.

Cuadro 47. Análisis de coherencia: competencias disposicionales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS DE CONTENIDOS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Disposicionales (ser/estar)			
OGD3	<p>CED2.1 CED3.1 CED4.1 CED5.1 CED5.2 CED7.1 CED9.1</p>	<p>2 3 5 5 7 9</p>	<p>Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.</p>	<p><u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Corrección individualizada. Nivel de actuación en las destrezas sobre la abstracción en la tarea de programar.</p>
OGD4	<p>CED1.1 CED2.2 CED3.2 CED6.1 CED8.1 CED10.2 CED11.1</p>	<p>1 2 3 6 8 10 11</p>	<p>Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.</p>	<p><u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Corrección individualizada. Nivel de actuación en las destrezas de explicación de procedimientos (oral o escrito).</p>

Cuadro 48. *Análisis de coherencia: competencias disposicionales (ser y estar)*

GUÍA DOCENTE DE PROGRAMACIÓN ORIENTADA A OBJETOS

Cristina Cachero Castro; Estela Saquete Boró

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
(ccachero; stela)dlsi.ua.es*

1. CONTEXTUALIZACIÓN

Actualmente, los títulos en Informática pertenecen al área de las Tecnologías de la Información y Comunicaciones (TIC). En España, la profesión Informática está muy difuminada, ya que existe una multiplicidad de títulos relacionados directamente con la disciplina, que da lugar a profesionales con formaciones muy dispares. Aún más, profesionales formados en otras disciplinas como por ejemplo Ingeniería en Telecomunicaciones, Ingeniería Industrial, Licenciados en Ciencias Físicas y en Matemáticas, etc., también trabajan en este sector, por lo que no se puede hablar de un perfil unificado.

El intento de justificar esta disparidad de títulos en el área de las TIC mediante el establecimiento de unos contenidos básicos y unas atribuciones profesionales bien definidas, diferenciadas y lo suficientemente estancas, ha resultado infructuoso. Este hecho ha provocado que, en el nuevo marco europeo, se haya optado por la reducción a dos únicas titulaciones, que se relacionan con los campos propios de la Informática y de las Telecomunicaciones respectivamente. En este marco, los objetivos primordiales en la formación de un Ingeniero en Informática hacen referencia tanto al ámbito cognoscitivo como a las habilidades y aptitudes que permiten aplicar los conocimientos adquiridos en el ejercicio de la profesión. Así, un Ingeniero en Informática debería ser capaz de abordar problemas desconocidos y adaptarse a la rápida evolución del sector.

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

Los lenguajes de programación han sido desde siempre piedra angular de todas las propuestas docentes en titulaciones relacionadas con la informática. Dichas propuestas han ido evolucionando a medida que se ha producido un avance histórico considerable de los lenguajes de programación, que ha estado estrechamente relacionado con el desarrollo del hardware y software de los ordenadores. Actualmente, es muy común hablar de generaciones de ordenadores y, de una forma similar, de generaciones de lenguajes de programación. La historia de los lenguajes de programación puede ser dividida en cuatro (o cinco, según algunos autores) generaciones. Las dos primeras generaciones siguen una sucesión en el tiempo, mientras que las dos últimas siguen evolucionando en la actualidad de forma casi paralela, debido a que propugnan distintos paradigmas, cada uno de ellos especialmente pensado para la resolución de un tipo de problema distinto.

Estas generaciones son:

- *Primera generación:* años previos a la segunda guerra mundial. Problemas numéricos. Lenguajes máquina y ensamblador.
- *Segunda Generación:* principio de los años cincuenta. Notaciones simbólicas. FORTRAN (1957), ALGOL (década de los sesenta), COBOL (finales de los cincuenta), LISP (principio de los sesenta), BASIC (1965). BASIC fue el primer lenguaje incluido en el software distribuido con los primeros ordenadores personales.
- *Tercera generación.* La creciente complejidad, tamaño e importancia de los sistemas y aplicaciones, las necesidades de fiabilidad en los programas y, al mismo tiempo, la falta de adecuación de los métodos tradicionales a dichas necesidades ha provocado a finales de los sesenta la denominada *crisis del software*. Como respuesta surgen los conceptos de abstracción funcional, método de diseño descendente (top-down), y se propone un conjunto de principios de programación estructurada (Dijkstra) que se aplican a nuevos lenguajes: PASCAL (Wirth, 1971). C (lenguaje en el que se desarrolló el sistema operativo UNIX a principios de los setenta), paradigma lógico (PROLOG (1972) y **paradigma orientado a objetos** (SIMULA (1967), Smalltalk (1980), Eiffel (1986), C++ (1986), Java (1995),...). Este paradigma modifica el tradicional enfoque a procesos por un enfoque a datos e introduce nuevos conceptos como los de clase, objeto y herencia.
- *Cuarta Generación.* Se sigue aumentando el nivel de abstracción. Aparecen lenguajes que pretenden acelerar el proceso de construcción de aplicaciones y facilitar su mantenimiento, incluso posibilitan que el propio usuario final pueda resolver sus propios problemas usando el lenguaje de programación. Estos lenguajes están orientados sobre todo a la gestión y en espe-

cial al tratamiento de bases de datos. Algunos de estos lenguajes de programación tienen su propia estructura y algunos otros realmente lo que hacen es el de permitir al programador crear programas en un lenguaje de tercera generación utilizando muchas menos sentencias.

Esta evolución se ha visto plasmada en el número y tipo de materias propuestas (relacionadas con la programación) por las principales recomendaciones curriculares internacionales de Informática. Entre ellas, quizás una de las de más impacto sea el Computing Curricula. En su versión del 2001 [ACM2001] ya se enfatizaba que uno de los cambios fundamentales respecto a versiones anteriores era el aumento significativo de la importancia de ciertos temas curriculares, entre los que destacaba la Programación Orientada a Objetos. De hecho, en su definición del Cuerpo de Conocimiento de Computer Science, la Programación orientada a objetos aparece como un core topic, con un número de horas asignadas igual al asignado a la enseñanza de los Algoritmos de Computación Básicos, por poner un ejemplo. Esta importancia está plenamente justificada. Por un lado el paradigma orientado a objetos ha demostrado sobradamente su grado de madurez y la viabilidad de sus propuestas. Por otro, la aplicación de este paradigma supone un cambio cualitativo en el modo de construir aplicaciones ya que, tal y como enfatiza Meyer [MEYER97], proporciona mecanismos que permiten aumentar la calidad del producto software resultante (aplicaciones más correctas, más robustas, más extensibles y más reutilizables).

La importancia del aprendizaje de los distintos paradigmas de programación en las titulaciones de Informática se refleja de manera muy clara en la última revisión de los distintos perfiles que componen esta recomendación (Computing Curricula 2004 [ACM2004]), en la que se proponen cinco grandes perfiles genéricos de Informática: Computer Science[CC2001], Computer Engineering [CE2004], Software Engineering[SE2004], Information Systems[IS2002] e Information Technology[IT2005]. De entre ellos, el área de conocimiento Programming Fundamentals (dentro de la cual podemos englobar la Programación Orientada a Objetos) aparece con relevancia máxima (nivel 5) en los perfiles Computer Science, Computer Engineering y Software Engineering, que son precisamente los perfiles en los que se enmarcan las actuales titulaciones de Informática en España. Esto implica que los nuevos planes de estudio deberían poner el máximo énfasis (a través de las distintas asignaturas de especialización ofertadas por cada plan más allá de los mínimos exigibles por el currículo de grado) en la enseñanza de esta área de conocimiento en relación con el resto de áreas y perfiles impartidos.

Por otro lado, hay que tener en cuenta (tal y como hace el Computing Curricula 2004) que la Programación Orientada a Objetos implica no sólo un lenguaje que soporte clases, objetos, herencia y polimorfismo sino una nueva forma de diseñar el programa, y de hecho nos encontramos con que su enseñanza se afian-

za a menudo dentro del marco de asignaturas relacionadas con la disciplina de Ingeniería del Software. A decir verdad, existe cierta controversia en cuanto a si se debería enseñar antes Ingeniería del Software y sólo más adelante introducir la programación o si, por el contrario, se debería comenzar con cursos de programación y en cursos posteriores profundizar en conceptos propios de la Ingeniería del Software. Igualmente, existen distintas propuestas en cuanto al orden en que se deberían enseñar los distintos paradigmas: mientras unos autores propugnan comenzar por el paradigma imperativo, otros proponen comenzar directamente con el paradigma orientado a objetos. En cualquier caso, tal y como se afirma en el perfil de Software Engineering "... La programación es una habilidad básica requerida por todos los ingenieros software; también es una habilidad que requiere de mucha práctica para su correcta adquisición. Mientras más pronto los estudiantes practiquen la programación, mayor habilidad adquirirán..." En este perfil [SE2004], la programación orientada a objetos aparece de manera explícita en las siguientes asignaturas:

CS102I (The Object-Oriented Paradigm): introduce el concepto de programación orientada a objetos a alumnos con un bagaje en el paradigma imperativo. El curso comienza con una revisión de las estructuras de control y los tipos de datos, con énfasis en los tipos de datos estructurados y en el procesamiento de arrays. Posteriormente, introduce el paradigma orientado a objetos, focalizándose en la definición y uso de clases junto con los fundamentos del diseño orientado a objetos. Otros temas incluyen una visión general de los principios generales de algún lenguaje de programación, análisis simples de algoritmos, técnicas básicas de ordenación y búsqueda, y una introducción de principios de ingeniería del software.

De los contenidos propuestos para CS102I, todos los referidos a la programación orientada a objetos (diseño orientado a objetos, encapsulación y ocultación de información, separación de comportamiento e implementación, clases, subclases y herencia, polimorfismo, jerarquías de herencia) están cubiertos en la asignatura objeto de este estudio.

CS 103 (Data Structures and Algorithms): basada en los fundamentos proporcionados por la secuencia de asignaturas CS101I-102I. Esta materia introduce los conceptos fundamentales de estructuras de datos y los algoritmos que proceden de ellas. Algunos de los temas tratados son recursividad, la filosofía subyacente de la programación orientada a objetos, los fundamentos de las estructuras de datos (pilas, colas, listas enlazadas, tablas asociativas, árboles y grafos), las bases del análisis de algoritmos y una introducción a los principios de traducción de lenguajes.

Además de una profundización en los conceptos OO introducidos en la asignatura anterior, esta materia proporciona una visión general de los lenguajes y paradigmas de programación, que es un tema también incluido en nuestra asignatura.

SE201 (Introduction to Software Engineering): curso que presenta los principios y conceptos básicos de la Ingeniería del Software y da una base para muchos otros cursos posteriores. Cubre de manera extensa la mayor parte de la terminología y conceptos de la ingeniería del software. Al terminar el curso, los alumnos serán capaces de realizar un análisis y diseño básico OO, particularmente utilizando UML. Además, tendrán un conocimiento básico de requisitos, arquitectura software y testeo.

La conexión entre diseño orientado a objetos utilizando UML y su implementación es un tema cubierto de manera extensiva por la asignatura objeto de este estudio.

SE312 (Low-Level Design of Software): esta asignatura abarca el diseño y la construcción detallada del software. También cubre de manera profunda los patrones de diseño y la refactorización. Se introducen aproximaciones formales al diseño. Se analizan diseños basados en criterios de calidad internos. Se mejora el rendimiento y la mantenibilidad de programas. Se realiza ingeniería inversa. Se presentan aproximaciones disciplinadas al cambio en el diseño.

En esta asignatura, entre otras cosas, el alumno aprende a utilizar patrones de diseño, así como a programar con un alto nivel de eficiencia. Estos objetivos de aprendizaje están también presente en la asignatura objeto de este estudio.

Por último, destacar que en el perfil de Software Engineering, las distintas posibilidades en cuanto al orden de aprendizaje de estas asignaturas se reflejan en distintos ‘itinerarios’, es decir, distinta ordenación temporal de las materias en el plan de estudios.

Versiones anteriores de estas recomendaciones han influido en la elaboración de las directrices generales propias de las titulaciones de informática (Reales Decretos 1459/1990 1460/1990 1461/1990 del 26 Octubre, BOE 20 Noviembre 1990). Lamentablemente, debido a su momento de elaboración, en estos reales decretos no se incluye ningún epígrafe específico para la Programación Orientada a Objetos. Según estas directrices, las materias más relacionadas con la asignatura Programación Orientada a Objetos son:

- Metodología y Tecnología de la Programación: materia troncal (y por tanto de obligatoria inclusión) para todas las titulaciones en todos los planes de estudios conducentes a la obtención de los títulos de Ingeniero Informático, Ingeniero Técnico en Informática de Gestión, Ingeniero Técnico en Informática de Sistemas. 15 créditos. Contenido: Diseño de Algoritmos, Análisis de Algoritmos, Lenguajes de Programación, Diseño de programas: descomposición modular y documentación, Técnicas de verificación y pruebas de programas.
- Estructura de Datos y de la Información: materia troncal (y por tanto de obligatoria inclusión) para todas las titulaciones en todos los planes de estudios conducentes a la obtención de los títulos de Ingeniero Informáti-

co, Ingeniero Técnico en Informática de Gestión, Ingeniero Técnico en Informática de Sistemas. 12 créditos. Contenidos: Tipos Abstractos de Datos, Estructura de datos y algoritmos de manipulación, Estructuras de información: Ficheros, Bases de datos.

Siguiendo estas directrices, en la Universidad de Alicante, la materia troncal Metodología y Tecnología de la Programación se estructura en las siguientes asignaturas, troncales para todas las titulaciones, definidas por la universidad:

- Fundamentos de Programación I. Asignatura básica que facilita una visión inicial del campo de la programación a través del estudio de los elementos básicos de un lenguaje de programación. Primer curso. Primer cuatrimestre.
- Fundamentos de Programación II. Asignatura que facilita la aplicación de los conceptos generales de un lenguaje mediante el estudio y uso de un lenguaje de programación de tipo imperativo. Primer curso. Segundo Cuatrimestre.
- Diseño y Análisis de Algoritmos. Asignatura que estudia los conceptos de diseño análisis y verificación de algoritmos y sus técnicas asociadas. Tercer curso. Primer cuatrimestre.

Las tres asignaturas tienen actualmente una carga docente de 6 créditos, lo que hacen un total de 18 créditos, es decir, tres más de los quince mínimos que establecen los Reales Decretos.

Asimismo, en la Universidad de Alicante la materia troncal Estructura de datos y de la Información se materializa en dos asignaturas anuales de 9 créditos cada una:

- Programación y Estructuras de Datos. Asignatura que introduce el diseño recursivo, así como los fundamentos del paradigma orientado a objetos de una manera pragmática, a través del estudio de los principales TAD's (pilas, colas, listas, árboles y grafos).
- Bases de Datos I. Asignatura básica que proporciona una visión general de los principios subyacentes de las bases de datos relacionales, y que introduce tanto el cálculo relacional como el modelo EER.

Ambas asignaturas se imparten en segundo curso y, debido a su carga docente, son anuales. Juntas, suman los 18 créditos mínimos establecidos para esta materia troncal en los Reales Decretos de las tres titulaciones.

A pesar de esta ausencia significativa del contenido Programación Orientada a Objetos en los Reales Decretos españoles, cabe destacar cómo los planes de estudio de la Universidad de Alicante ponen los medios para subsanar esta carencia mediante la oferta de un número elevado de asignaturas obligatorias, con una carga de 4,5 créditos cada una, que complementan los contenidos de las materias troncales. Dichas asignaturas son:

- Programación Orientada a Objetos. Asignatura que introduce de una manera sistemática los principios del paradigma orientado a objetos, y pro-

fundiza en los conceptos de clase, objeto, mensaje, herencia, polimorfismo, gestión de errores y persistencia. Segundo curso. Primer cuatrimestre.

- **Herramientas de Programación.** Asignatura que introduce los entornos de desarrollo, los estándares de nomenclatura, la organización, indentado y comentario del código fuente, la gestión de proyectos, la programación por contrato, la internalización de aplicaciones y el control de versiones. Segundo curso. Primer cuatrimestre.
- **Lenguajes y Paradigmas de Programación.** Asignatura que da una visión general de los distintos paradigmas de programación, sus puntos fuertes y sus debilidades, y proporciona al alumno una base crítica sobre la cual discernir qué lenguaje puede adecuarse mejor a cada proyecto. Segundo curso. Segundo cuatrimestre.
- **Computabilidad:** Asignatura que, como su nombre indica, estudia la computabilidad los algoritmos, y hace especial énfasis en máquinas de Turing y funciones recursivas. Segundo curso. Segundo Cuatrimestre.
- **Diseño y Programación Avanzada de Aplicaciones.** Asignatura que profundiza en el paradigma orientado a objetos, e introduce los objetos distribuidos, las aplicaciones cliente/servidor y las aplicaciones para Internet. Tercer curso. Segundo Cuatrimestre.
- **Algoritmia Avanzada.** Asignatura que profundiza en conceptos presentados en Diseño y Análisis de Algoritmos, como son los algoritmos de búsqueda exhaustiva y estocástica, programación dinámica y algoritmos de codificación y compresión. Cuarto Curso. Primer Cuatrimestre.

Al analizar la adecuación del perfil de la asignatura a los perfiles de la titulación, debemos destacar que la Programación Orientada a Objetos juega un papel fundamental a la hora de construir sistemas software donde la calidad, fiabilidad y productividad deben estar garantizadas. Los descriptores de esta asignatura cubren los fundamentos de este paradigma, y son básicos para la formación de un informático independientemente de su perfil, ya que cubren un conjunto de técnicas de abstracción básicas para el diseño y la implementación de sistemas de información complejos, con un alto grado de fiabilidad y reuso.

Por tanto el contenido de la Programación Orientada a Objetos es relevante en los tres perfiles profesionales generales [EICE04]:

Perfil profesional de desarrollo de software:

- Un ingeniero con este perfil “sabe diseñar la arquitectura y detallar las especificaciones de funcionamiento; conoce la naturaleza y posibilidades de los distintos lenguajes de codificación y es capaz de realizar la implementación, de todo o parte del producto, mediante el uso de las diferentes metodologías y paradigmas de desarrollo que estén a su alcance. Está preparado para realizar la verificación modular de los desarrollos parciales, la

integración parcial o completa y las pruebas modulares y de sistema; está en disposición de validar el producto para la aceptación del cliente, de implantarlo y de ponerlo en explotación...” [EICE04]. Tal y como indica esta descripción, la programación orientada a objetos es fundamental en este perfil, ya que es el paradigma predominante en la actualidad para el desarrollo software y aplicaciones de tamaño medio.

Perfil profesional de sistemas:

- “Un Ingeniero en Informática, con perfil Sistemas, es capaz de especificar, modelar, diseñar, implantar, verificar, integrar, configurar, mantener y evaluar el rendimiento de cualquier sistema informático así como cada uno de sus componentes o partes. Por ello debe contar con sólidos conocimientos de las técnicas, dispositivos y herramientas propias del ámbito que le capaciten para la especificación, diseño, montaje, depuración, mantenimiento y evaluación del rendimiento del hardware de computadores y sus periféricos habituales. Asimismo, debe ser competente para el desarrollo del software del sistema que posibilita una gestión eficaz de los recursos hardware del sistema informático...” [EICE04] La Programación Orientada a Objetos proporciona de manera natural mecanismos de integración, seguridad etc. que facilitan el desarrollo de sistemas distribuidos. De hecho es un paradigma que está en la base de muchas de las plataformas de desarrollo actuales (J2EE, .NET, etc).

Perfil profesional de gestión y explotación de tecnologías de información:

- “Un Ingeniero en Informática con perfil profesional de Gestión y Explotación de Tecnologías de la Información es responsable de asegurar que las necesidades de Gestión de la Información y del Conocimiento de las organizaciones se satisfacen con el desarrollo y la implantación de soluciones informáticas. Conoce la estrategia empresarial y las diferentes soluciones de las Tecnologías de la Información y de las Comunicaciones necesarias para apoyar dicha estrategia.”. Aunque quizás este sea el perfil en que menos importancia relativa tenga la Programación Orientada a Objetos, un ingeniero de estas características debe poder comunicarse con el ingeniero software en términos comprensibles para ambos, y el paradigma orientado a objetos proporciona una buena base para la discusión de la arquitectura de los sistemas a desarrollar.

Por último, cabe destacar que la utilización en esta asignatura de artefactos aprendidos en otras materias que cubren el área de Ingeniería del Software ayuda a comprender el proceso de desarrollo de software, al mismo tiempo que enfatiza la utilidad de los artefactos software (en nuestro caso diagramas UML) como vehículo de reflexión y comunicación entre programadores.

1.2 LA PROGRAMACIÓN OO EN EL EEES

La asignatura Programación Orientada a Objetos es además perfectamente coherente con la propuesta del Libro Blanco que recoge los cambios que deben efectuarse en los planes de estudio de las titulaciones universitarias de informática en el nuevo espacio europeo de educación superior [EICE04]. Muy sucintamente, a partir del 2010, se prevé que exista una sola titulación de Grado denominada Ingeniería en Informática, que recoge los contenidos generales y básicos de la enseñanza superior informática. Estos estudios de Grado constarán de 240 créditos ECTS y están organizados en 4 años. Los contenidos Formativos Comunes de la titulación representan un 60% de la carga de los estudios, incluyendo la carga asignada al Proyecto Fin de Carrera, dejando el 40% restante para materias ofertadas discrecionalmente por cada Universidad. El título de Grado dará acceso a un segundo ciclo de carácter puramente profesional, denominado Master. El Master está destinado a la especialización profesional de los Ingenieros en Informática, o bien a su preparación para la investigación. Los estudios de Máster constarán de entre 60 y 120 créditos ECTS, y permitirán el acceso a la realización de la tesis doctoral con el objeto de obtener el grado de Doctor.

Los Contenidos Formativos Comunes (CFC), que representan el 60% de la carga de los estudios (al menos 144 ECTS de los 240 de los que consta la titulación), y dentro de los que se engloba la materia objeto de este estudio, quedan distribuidos de la siguiente forma:

1. Fundamentos científicos (10% - 15%)

- Fundamentos matemáticos de la Informática
- Fundamentos físicos de la Informática

2. Contenidos Generales de la Ingeniería (5% - 10%)

- Gestión de las organizaciones
- Ética, legislación y profesión
- Destrezas profesionales

3. Contenidos específicos de la Ingeniería en Informática (35% - 40%)

- Programación
- Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes
- Sistemas operativos, Sistemas Distribuidos y Redes
- Ingeniería de Computadores

4. Proyecto Fin de Carrera (6%)

El libro blanco detalla además los contenidos de cada categoría. En concreto, para los contenidos específicos se especifica lo siguiente:

- Programación (P): fundamentos y metodología de la programación, algoritmia, computabilidad, lenguajes de programación, paradigmas de programación, estructuras de datos.
- Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes (IS): desarrollo de software, procesos, requisitos, especificación y diseño,

gestión de proyectos, calidad del software, interacción persona-computadora, bases de datos, inteligencia artificial.

- Sistemas Operativos, Sistemas Distribuidos y Redes (SO): sistemas operativos, sistemas distribuidos, sistemas de tiempo real, arquitectura e infraestructura de redes y servicios telemáticos, seguridad.
- Ingeniería de los computadores (IC): fundamentos, estructura y arquitectura de computadores, tecnología de computadores.

La materia de Programación Orientada a Objetos (POO) se engloba, dentro de los Contenidos Específicos de la Ingeniería en Informática, en las subcategoría Programación, donde cubre parcialmente los epígrafes fundamentos y metodología de la programación, paradigmas de programación y lenguajes de programación. De hecho, en el libro blanco se identifica la Programación Orientada a Objetos como un cambio “revolucionario”, que influye de forma determinante en la enseñanza de la Informática, junto con otras áreas como la web, las nuevas tecnologías de red, los gráficos y multimedia, las técnicas de simulación, los sistemas empujados, las bases de datos relacionales o el uso de sofisticados interfaces para el programador de aplicaciones.

En este contexto, todas las asignaturas relacionadas con la Programación Orientada a Objetos forman parte de los Contenidos Específicos, subcategoría Programación, tal y como mostramos en la siguiente tabla:

EPÍGRAFE	ASIGNATURA
Fundamentos y metodología de la programación	Fundamentos Programación I Fundamentos Programación II Herramientas de Prgramación
Algoritmia, computabilidad	Diseño y Análisis de Algoritmos Algoritmia Avanzada Computabilidad
Lenguajes de Programación	Lenguajes Gramáticas y Autómatas Procesadores del Lenguaje
Paradigmas de Programación	Lenguajes y Paradigmas de Programación Programación Orientada a Objetos Diseño y Programación Avanzada de Aplicaciones
Estructura de datos	Programación y Estructuras de Datos

Cuadro 1. Relación de asignaturas relacionadas con Programación Orientada a Objetos

1.3 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Programación Orientada a Objetos es obligatoria en el segundo curso de las tres ingenierías ofertadas por la Universidad de Alicante. Esta asignatura está estrechamente relacionada con otras asignaturas troncales, obligatorias y optativas, que presentamos en la tabla 1 junto con sus correspondientes descriptores. En la siguiente tabla la primera columna indica el curso en el que se imparte la asignatura (en caso de ser troncal u obligatoria) y el cuatrimestre (primero, segundo o asignatura anual).

CURSO/ CUATR	TIPO	ASIGNATURA	DESCRIPTOR
1/1	T	Fundamentos programación I (fp1)	<ul style="list-style-type: none"> • Introducción a la programación. • Diseño de algoritmos
1/2	T	Fundamentos programación II (fp2)	<ul style="list-style-type: none"> • Análisis y diseño de programas. • Lenguajes de programación.diseño descendente.
2/A	T	Programación y estructuras de datos (ped)	<ul style="list-style-type: none"> • Estructura de datos i algoritmo de manipulación. • Tipos abstractos de datos. • Diseño recursivo.
2/1	O	Programación orientada a objetos (poo)	<ul style="list-style-type: none"> • Metodología.características de la poo. • Clases y objetos Diseño orientado a objetos. Lenguajes de programación orientado a objetos. Objetos distribuidos. Herencia y genericidad. • Persistencia en un entorno orientado a objetos.
2/1	O	Herramientas de programación (hp)	<ul style="list-style-type: none"> • Entornos de desarrollo. • Estándares de nomenclatura. • Indentado y comentario en el código fuente. • Organización del código fuente. • Gestión de proyectos. Programación por contrato. Internalización de aplicaciones. • Control de versiones
2/1	T	Lenguajes, gramáticas y autómatas	<ul style="list-style-type: none"> • Máquinas secuenciales y autómatas finitos. • Gramáticas y lenguajes formales. • Redes neuronales
2/2	O	Computabilidad (c)	<ul style="list-style-type: none"> • Máquinas de turing. Funciones recursivas
2/2	O	Lenguajes y paradigmas de programación (lpp)	<ul style="list-style-type: none"> • Programación procedimental. Programación funcional. Programación declarativa. Programación orientada a objetos. • Lenguajes de script
3/1	T	Diseño y análisis de algoritmos (daa)	<ul style="list-style-type: none"> • Diseño de programas: descomposición modular y documentación. • Técnicas de verificación y pruebas de programas.la eficiencia de los algoritmos. • Divide y vencerás. Algoritmos voraces. Algoritmos con retroceso.

CURSO/ CUATR	TIPO	ASIGNATURA	DESCRIPTOR
3/2	O	Diseño y programación avanzada de aplicaciones (dpaa)	<ul style="list-style-type: none"> • Aplicaciones distribuidas. Aplicaciones internet. Sistemas abiertos. • Objetos distribuidos. Cliente/servidor
4/1	O	Algoritmia avanzada (aa)	<ul style="list-style-type: none"> • Búsqueda exhaustiva y estocástica. programación dinámica. algoritmos de codificación y compresión
4/A	T	Procesadores de lenguaje (pl)	<ul style="list-style-type: none"> • Compiladores. • Traductores e intérpretes. • Fases de compilación. Optimización de código. • Microprocesadores
—	OPT	Programación concurrente (pc)	<ul style="list-style-type: none"> • Procesos. • Sincronización. Competencia y Cooperación. Exclusión mutua. • Memoria Compartida y distribuida. CSP
—	OPT	Programación en entornos interactivos (pei)	<ul style="list-style-type: none"> • Programación visual. Programación dirigida a eventos. Interfaces gráficos de usuarios
—	OPT	Programación en internet (pi)	<ul style="list-style-type: none"> • Desarrollo y programación de sistemas de acceso a bases de datos de internet. Planificación, diseño y administración de sitios web. • Migración de aplicaciones a entornos en internet. • Herramientas de desarrollo Diseño y programación de elementos multimedia en internet

Cuadro 2. Relaciones de Programación Orientada a Objetos en el plan de estudios

Todas estas asignaturas están claramente vinculadas, tal y como se muestra en el siguiente diagrama de actividad, donde los estados reflejan las asignaturas troncales/obligatorias y las transiciones reflejan los prerrequisitos existentes entre ellas.

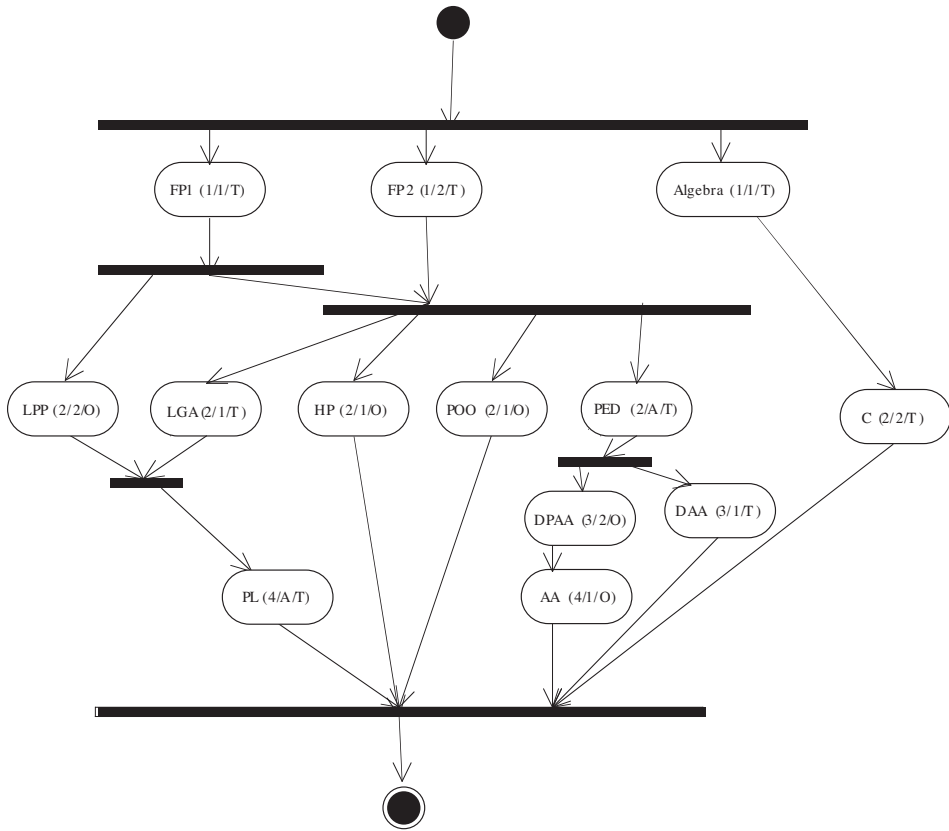


Ilustración 10. Diagrama de actividad

La asignatura de POO supone, junto con PED y LPP (impartidas de manera simultánea), el primer contacto que el alumno tiene con el paradigma OO y con C++, aunque cuando cursa esta asignatura el alumno ya viene con nociones de programación estructurada y del lenguaje C, adquiridas en FP1 y FP2. Tras cursar POO, el alumno debe cursar otras asignaturas que requieren dominar la programación orientada a objetos para su correcta asimilación: DPAA, DAA y AA.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes:

Objetivos Instrumentales Generales (saber y saber hacer)

- Dominar y utilizar la terminología usual de la asignatura no sólo en castellano/valenciano sino también en inglés, lengua franca de la Informática, tal y como reconoce el Libro Blanco de la titulación [EICE04].
- Ser capaz de discernir los tipos de aplicación y las situaciones en las que es posible y necesario aplicar el paradigma orientado a objetos.
- Comprender aquellos conceptos básicos y métodos relacionados con la Programación Orientada a Objetos que son recomendaciones o prerrequisitos para otras asignaturas.
- Comprender, interpretar y analizar el cambio de enfoque en el modo de resolver problemas que supone el uso del paradigma orientado a objetos respecto a otros paradigmas.
- Comprender la enorme importancia de crear software fiable, reutilizable y mantenible.
- Dominar estrategias básicas de reuso como son el uso de librerías y de patrones software.
- Desarrollar su capacidad de abstracción de cara a futuras tareas de análisis y diseño software.
- Experimentar con entornos y herramientas de desarrollo orientado a objetos de libre distribución.
- Ser capaz de comparar distintos lenguajes de programación orientados a objetos y apreciar sus ventajas e inconvenientes en base a su grado de cumplimiento de las principales características del paradigma orientado a objetos.

Objetivos Interpersonales Generales (ser y estar)

- Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso: cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

Objetivos Sistémicos Generales

- Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS ACADÉMICAS Y PROFESIONALES

2.2.1 Competencias Instrumentales Específicas (saber y saber hacer)

- **Habilidades Cognitivas (saber):** además de las habilidades cCIC1 y cCIC2, desarrolladas en el artículo introductorio de este libro, tendremos en cuenta las siguientes:

Bloque 1: Introducción al Paradigma Orientado a Objetos

- CIC1: Entender el concepto de paradigma y sus implicaciones en el modo de resolver problemas.
- CIC2: Conocer y entender el proceso de evolución de los distintos paradigmas de programación
- CIC3: Entender el tipo de problemas de desarrollo software que palía un uso correcto del paradigma orientado a objetos
- CIC4: Conocer el modo en que el paradigma orientado a objetos ayuda a mejorar las capacidades de reuso del software.

Bloque 2: Conceptos Básicos de la Programación Orientada a Objetos

- CIC5: Entender los conceptos de clase, atributo, operación, interfaz y objeto.
- CIC6: Entender el mecanismo de paso de mensajes.
- CIC7: Comprender el modo en que se deben implementar los caminos de comunicación entre clases para permitir el paso de mensajes entre ellas.
- CIC8: Entender y ser capaz de implementar los distintos tipos de relaciones que se pueden establecer a nivel de objeto entre dos clases: asociaciones, agregaciones y composiciones.
- CIC9: Entender el concepto de estado de un objeto.
- CIC10: Entender la relación entre diagramas de clase UML y código.

Bloque 3: Herencia y Polimorfismo

- CIC11: Entender el mecanismo de abstracción de la herencia.
- CIC12: Saber discernir entre jerarquías de herencia seguras (bien definidas) e inseguras.
- CIC13: Comprender los costes de la herencia.
- CIC14: Saber decidir cuándo usar herencia y cuándo optar por composición.
- CIC15: Entender el concepto de polimorfismo.
- CIC16: Entender la diferencia entre ligadura estática y ligadura dinámica en los lenguajes de programación.
- CIC17: Entender la relación a nivel de implementación entre herencia y polimorfismo.
- CIC18: Saber identificar los distintos tipos de polimorfismo: sobrecarga, sobrescritura, variables polimórficas y genericidad.
- CIC19: Entender las relaciones entre los distintos tipos de polimorfismo.

Bloque 4: Otras características del Paradigma Orientado a Objetos

- CIC20: Entender los mecanismos de gestión de errores que ofrecen algunos lenguajes de programación.
- CIC21: Entender el concepto de persistencia.
- CIC22: Entender el concepto de concurrencia.

Capacidades Metodológicas (saber hacer): además de las capacidades metodológicas generales cCIM1, cCIM2 y cCIM3, introducidas en el artículo introductorio de este libro, para esta asignatura consideramos importante:

- CIM1: Ser capaz de interpretar un diagrama de clases UML para, a partir de él, proceder a la codificación de la aplicación.
- CIM2: Ser capaz de aplicar las distintas técnicas de reuso de software (composición, herencia, polimorfismo), interpretando cuál es la más adecuada.

Destrezas Tecnológicas (saber hacer): además de la destreza tecnológica cCIT1, introducida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con fluidez las herramientas de programación de libre distribución gdb, doxygen y el compilador g++ de C++.

Destrezas Lingüísticas (saber hacer): además de las destrezas lingüísticas cCIL1 y cCIL2, introducidas en el artículo introductorio de este libro, se consideran las siguientes dos destrezas lingüísticas:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje de diseño software, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier interacción o relación entre elementos del sistema.
- CIL2: Comprender sinónimos y expresiones en otras lenguas utilizadas por otros autores para referirse a conceptos tratados en la asignatura.

2.2.2 Competencias Interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo. En ambos casos las competencias definidas para la asignatura coinciden con las aportadas el artículo introductorio de este libro: cCIPTC1, cCIPTC2, CIPTR1, cIPTR2, cIPTR3 y cIPTR4.

2.2.3 Competencias Sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5. Además, consideramos competencias sistémicas de la asignatura:

- CS1: Capacidad de aplicación de los conocimientos, métodos y destrezas prácticas del paradigma orientado a objetos para el diseño y desarrollo de sistemas de información más robustos y mantenibles.
- CS2: Extrapolación los conocimientos adquiridos a otros lenguajes de programación.
- CS3: Capacidad de prueba exhaustiva de los programas implementados.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Entender el concepto de paradigma.
- Entender el modo de abordar la resolución de problemas en el paradigma orientado a objetos.
- Comprender la información contenida en un diagrama de clases UML y ser capaz de codificar dicha información en un lenguaje de programación orientado a objetos.
- Entender el concepto de relación entre clases, y cómo a través de las relaciones se produce el paso de mensajes entre objetos del sistema.
- Entender el concepto de generalización y cómo este se puede plasmar en jerarquías de herencia seguras.
- Entender el concepto de polimorfismo y ser capaz de aplicarlo para mejorar la reusabilidad del código.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los prerrequisitos necesarios para el estudio y entendimiento de esta asignatura se cubren en las asignaturas FP1 y FP2, que se imparten en el primer curso de las tres ingenierías de informática. Sin embargo, el alumnado de esta asignatura no es totalmente homogéneo, ya que con frecuencia hay estudiantes que, sin haber superado dichas asignaturas, se matriculan en Programación Orientada a Objetos. Para ellos, y para todo aquel alumno que desee comprobar/reforzar los conocimientos adquiridos sobre la construcción de algoritmos y el manejo de memoria en C++, en el Campus Virtual, se actualiza todos los años una pequeña serie de libros y direcciones de Internet donde se recogen tutoriales de C que tratan los principales problemas con los que el alumno se puede encontrar a la hora de cursar nuestra signatura, principalmente el manejo de punteros. Además, en las horas de laboratorio se imparte un seminario de 3 sesiones (4,5 horas) donde, a nivel práctico, se refrescan conceptos básicos de programación: construcción de algoritmos, manejos de arrays, etc. Además, se intenta que los temas sean lo más autocontenidos posibles, para lo cual se recuerda al principio de cada tema aquellos conceptos que van a necesitar para entenderlo, y se refuerzan dichos conceptos al final. De esta forma, aquellos estudiantes que hayan olvidado estos conceptos podrán recordarlos en clase y, si lo creen necesario, profundizar en ellos con la ayuda del material bibliográfico.

4. BLOQUES Y TEMAS DE CONTENIDO

En este punto y, siguiendo los objetivos y competencias concretos de la programación orientada a objetos, enunciamos el contenido del programa de teoría y práctica de la asignatura de **Programación Orientada a Objetos**.

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque I: Introducción al Paradigma Orientado a Objetos

- Unidad 0: Presentación y objetivos.
- Unidad 1: Introducción al paradigma Orientado a Objetos.

Bloque II: Conceptos Básicos de la Programación Orientada a Objetos

- Unidad 2: Fundamentos de la POO.

Bloque III: Herencia y Polimorfismo

- Unidad 3: Herencia.
- Unidad 4: Polimorfismo.

Bloque IV: Otras características del Paradigma Orientado a Objetos

- Unidad 5: Características Avanzadas del Paradigma Orientado a Objetos.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque I: Introducción al Paradigma Orientado a Objetos

- Unidad 0: Presentación y objetivos.
- Unidad 1: Introducción al paradigma Orientado a Objetos.
 1. El progreso de la abstracción.
 2. Principales Paradigmas de Programación.
 3. El Paradigma Orientado a Objetos.
 4. Historia de los Lenguajes Orientados a Objetos.
 5. Metas del Paradigma Orientado a Objetos.
 6. Conclusiones.
 7. Ejercicios Propuestos.

Bloque II: Conceptos Básicos de la Programación Orientada a Objetos

- Unidad 2: Fundamentos de la POO.
 1. Clase.
 2. Atributos.
 3. Operaciones.
 4. Interfaces.
 5. Objetos.
 6. Metaclases.
 7. Diseño de Aplicaciones OO.
 8. Relaciones entre clases y objetos.
 9. Conclusiones.
 10. Ejercicios Resueltos.
- Ejercicios Propuestos.

Bloque III: Herencia y Polimorfismo

- Unidad 3: Herencia.
 1. Introducción a la Herencia.
 2. Herencia Simple.
 3. Herencia Múltiple.
 4. Herencia de Interfaz.
 5. Herencia de Implementación.
 6. Beneficios y costes de la herencia.
 7. Elección de la técnica de reuso.
 8. Conclusiones.
 9. Ejercicios propuestos.
- Unidad 4: Polimorfismo.
 1. Polimorfismo y reuso.
 2. Sobrecarga.
 3. Sobreescritura.
 4. Variables Polimórficas.
 5. Genericidad.
 6. Caso de estudio.
 7. Conclusiones.
 8. Ejercicios Propuestos.

Bloque IV: Otras características del Paradigma Orientado a Objetos

- Unidad 5: Características Avanzadas del Paradigma Orientado a Objetos
 1. Gestión de Errores.
 2. Persistencia.
 3. Concurrencia.
 4. Conclusiones.
 5. Ejercicios Propuestos.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente se ha desarrollado en el artículo introductorio de este libro desde el punto de vista general para el segundo curso de la titulación de Informática. Dicho desarrollo, aunque genérico, se considera válido para el caso particular de la asignatura Programación Orientada a Objetos. Lo que cabe destacar serían las prácticas de laboratorio y las actividades en grupos pequeños, puesto que jugarán un papel fundamental. Las actividades que se proponen son:

- **Clases de teoría con apoyo de material audiovisual:** en lo que se refiere a las clases de teoría, cabe mencionar que éstas se apoyan de material audiovisual disponible para el alumnado y que le puede servir de guía sobre los contenidos más importantes de la asignatura. Además, los pro-

fesores de la asignatura hemos creado unos apuntes y estamos editando un libro que incluye todos los contenidos de la misma. De esta forma, el alumnado puede entender y asimilar mejor lo que se está explicando. Dichas explicaciones teóricas se intercalarán con la realización de problemas, ejemplos prácticos y aplicaciones siempre que el contenido lo requiera.

- **Actividades en grupos pequeños / tutorías docentes:** estas actividades estarán vinculadas con la realización de problemas y cuestiones teórico-prácticas relacionadas con la asignatura, de manera que se intente reforzar y aplicar los conceptos básicos a situaciones reales concretas y fomentar la capacidad de análisis, síntesis y autoevaluación del alumnado.
- **Prácticas de laboratorio:** en cuanto a las prácticas de laboratorio, cabe mencionar que éstas se preparan durante los meses previos al inicio de las prácticas, y son implementadas y testeadas por varios profesores antes del comienzo del curso. Las primeras sesiones se dedicarán a un seminario del lenguaje de programación utilizado en las mismas (C++ en nuestro caso) para permitir al alumno la toma de contacto con dicho lenguaje. El alumno deberá utilizar el material audiovisual y los apuntes de la asignatura como apoyo al desarrollo de dichas prácticas.
- **Trabajos complementarios:** en cuanto a los trabajos complementarios, comentar que dichos trabajos incidirán en la nota final de la asignatura y pueden ser de índole teórica, de índole práctica o de índole teórico-práctica y deberán realizarse de forma individual o por parejas.
- **Tutorías de atención al alumnado:** el alumnado tiene a su disposición unas horas de tutorías en las cuales puede consultar cualquier duda relacionada con la organización y planificación de la asignatura, así como dudas concretas sobre el contenido de la asignatura. Además de dichas tutorías individualizadas, se programarán varias tutorías en grupo, al menos una para cada bloque de la asignatura correspondiente.

5.2 ESTRATEGIAS DE APRENDIZAJE

Las estrategias de aprendizaje se han establecido a nivel común en el artículo introductorio de este libro. Tal y como se indicó, los medios tradicionales como las transparencias, apuntes o presentaciones por ordenador, no son los únicos medios sobre los que nos apoyaremos en nuestra docencia. Concretamente, las páginas web y el campus virtual ofrecen innumerables posibilidades que no hay que dejar pasar. Así, se ha elaborado en el campus virtual una página de la asignatura, que incluye toda la información que el alumno necesita. El uso de la misma ha sido mayoritario en las experiencias llevadas a cabo hasta el momento.

Entre otras cosas, en dicha página podemos encontrar:

- Tablón de anuncios: desde aquí, el alumno puede estar perfectamente informado de cualquier novedad relacionada con la asignatura, además de recordar los plazos de entrega de cualquier trabajo, fechas de exámenes, etc.
- Tutorías: indica el horario de atención al alumnado.
- Objetivos: resume los objetivos que se pretenden alcanzar en esta asignatura.
- Temario: especifica el temario de esta asignatura.
- Material docente de la asignatura: aquí se puede encontrar todo el material que como mínimo va a ser necesario para el seguimiento de las clases teóricas y prácticas.
- Bibliografía complementaria: aparece un listado de bibliografía complementaria que pueden consultar para profundizar en la asignatura o preparar los trabajos complementarios. Toda la bibliografía se encuentra disponible en la Biblioteca de la Universidad de Alicante.
- Plan de prácticas: se especifica semana a semana qué trabajo debe realizar el alumnado tanto en las horas presenciales como en las no presenciales, de forma autónoma.
- Horarios de teoría y prácticas: contiene información sobre los grupos de teoría y prácticas, así como el profesorado que los imparte.
- Exámenes de autoevaluación: actualmente sólo desarrollados para el seminario de C++, impartido durante las primeras tres sesiones prácticas de la asignatura, con el fin de que el alumno testee si posee el nivel de conocimientos mínimo que le permita afrontar la elaboración de la primera práctica de la asignatura con ciertas garantías de éxito.
- Enlaces de interés: aquí aparecen una serie de enlaces interesantes que pueden servir para profundizar en algunos contenidos de la materia.

En base a todo esto, la estrategia de aprendizaje que se propone se compone de las siguientes fases:

Recopilación de toda la documentación de la asignatura.

Planificación de las clases teóricas:

Lectura previa del guión correspondiente a la sesión de teoría que se trate.

Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido. En caso de no entender algo intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Planificación de las actividades en grupos pequeños:

Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la actividad a realizar en grupos pequeños para, al inicio de la actividad, poder preguntar las dudas surgidas en el entendimiento del enunciado.

En las actividades en grupos pequeños, cada subgrupo tendrá que hacer la actividad propuesta que será corregida entre todos, en el aula o por el profesor, fuera de ella.

Una vez corregida la actividad propuesta, los grupos deben analizar cuáles han sido los errores cometidos para intentar no volverlos a realizar. Si es necesario, se pedirá ayuda al profesor correspondiente.

Planificación de las clases prácticas:

Una vez entendidas las explicaciones de las clases teóricas se leerá, de forma independiente, la práctica de laboratorio que se debe realizar en la sesión correspondiente para, al inicio de la sesión, poder preguntar las dudas surgidas en el entendimiento del enunciado.

Parte de las prácticas se realizarán en los laboratorios y parte en horas no presenciales. Se deberá cumplir el calendario de entrega de prácticas. El profesorado corregirá en un plazo máximo de dos semanas dichas prácticas, indicando una vez corregidas los fallos más comunes y publicando los programas utilizados en la corrección. Cada estudiante, de forma individual, debe analizar cuáles han sido los errores cometidos y solucionarlos para la entrega siguiente. Si es necesario se pedirá ayuda al profesor correspondiente.

Autoevaluación: una vez realizadas todas las actividades previas relacionadas con un tema concreto, el estudiante debe discernir si cree que dicho tema ha sido totalmente entendido. En algunos temas se podrá ayudar de los tests de autoevaluación que hayan sido publicados por los profesores en el campus virtual. En caso de no haber asimilado los contenidos mínimos, el alumno debe incidir en el estudio de los contenidos que crea tener más flojos, utilizando, si lo cree conveniente, las tutorías y realizando algunos problemas de ampliación, bien de los propuestos en las hojas de problemas o bien haciendo uso de la bibliografía.

Evaluación final: si el resultado de las prácticas propuestas ha sido satisfactorio, el estudiante deberá realizar el examen de teoría, y una vez aprobado, habrá superado la asignatura.

De forma opcional, se podrá hacer trabajos complementarios individuales o por parejas para subir la nota, siempre y cuando el trabajo realizado a lo largo del curso se considere satisfactorio.

6. PLAN DE TRABAJO DE LOS ALUMNOS: ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DEL APRENDIZAJE

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial, y horas

no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

6.1 PLANIFICACIÓN DEL PROGRAMA DE TEORÍA

ACTIVIDAD		Horas Presenciales			Horas No Presenciales	
		Trabajo en aula	Tutorías organizadas	Aprendizaje Actividades complementarias	TOTAL autónomo y colaborativo	
BLOQUE 1	Unidad 0	1,5	0		0	1,5
	Unidad 1	3	1		3	7
BLOQUE 2	Unidad 2	4,5	2		6	12,5
BLOQUE 3	Unidad 3	4,5	2	1	6	13,5
	Unidad 4	6	2	1	10	19
BLOQUE 4	Unidad 5	1,5	3		2	6,5
Preparación del examen final:		1,5			15	16,5
Examen final:		2			0	2
TOTAL:		24,5	10	2	42	78,5

Cuadro 3. Planificación del programa de teoría

6.2 PLANIFICACIÓN DEL PROGRAMA DE PRÁCTICAS

ACTIVIDAD		Horas Presenciales		Horas No Presenciales	TOTAL
		Prácticas en aula	Tutorías organizadas	Realización de las prácticas fuera del horario de la asignatura	
Unidad 0: Seminario C++		4,5		0	4,5
Práctica 1		4,5		5	9,5
Práctica 2		6	0,5	8	14,5
Práctica 3		6	1	8	15
TOTAL:		21	1,5	21	43,5

Cuadro 4. Planificación del programa de prácticas

6.3 PLANIFICACIÓN (RESUMEN)

Tras realizar la planificación de la asignatura, obtenemos los siguientes resultados:

- Número total de horas presenciales: 59
- Número total de horas no presenciales: 63
- Número total de horas: 122

7. BIBLIOGRAFÍA Y MATERIALES RECOMENDADOS

7.1 BIBLIOGRAFÍA BÁSICA

Teoría

Budd, T. (2001) *An Introduction to Object-oriented Programming*. Ed. Addison Wesley

Eckel, B., *Thinking in C++*. 2nd Edition. Libro electrónico gratuito.
<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>

Prácticas

Harvey, M.; Deitel, P. (2005) *C++: How to Program*. 5ª Edición. Ed. US Imports & PHIPES

Aprenda C++ como si estuviera en primero www.cplusplus.com

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Teoría

Fundamentals of Object Oriented Design in UML. Mailir Page-Jones, 2000. Ed. Addison Wesley

Requirements Analysis and System Design. Developing Information Systems with UML. L.A. Maciaszek, 2001. Ed. Addison-Wesley

Object-oriented Software Construction (Prentice Hall International Series in Computer Science). Bertrand Meyer, 2000. Ed. Prentice Hall

The Object Primer 3rd Edition Agile Model Driven Development with UML 2. Scott Ambler. 2004. Ed. Cambridge University Press, ISBN#: 0-521-54018-6

Prácticas

Simply C++ An Application-Driven Tutorial Approach. 2005. Deitel & Deitel.
Programación en C++: algoritmos, estructuras de datos y objetos. L. Joyanes Aguilar, 2000. Ed. Mc Graw Hill

The C++ Primer. 3rd. Edition. S.B. Lippman, 1999. Ed. Addison Wesley.

7.3 OTROS RECURSOS

Páginas web de interés

- <http://se.inf.ethz.ch/touch/TOUCH.pdf>
- <http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>
- <http://www.hibernate.org/>

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE

8.1 PROCEDIMIENTO DE EVALUACIÓN

Según [MS04] el proceso evaluativo se ubica, al menos, en cuatro espacios de aprendizaje:

- Los contenidos: cuestiones y redes conceptuales
- Las prácticas: trabajos realizados dentro y fuera del aula durante el desarrollo de los créditos teóricos y prácticos
- La participación en el trabajo de aula, aprendizaje virtual y tutorías
- El proyecto o memoria final, portfolio, etc.

Para la evaluación del alumno en Programación Orientada a Objetos se sigue un tipo criterial, i.e. una evaluación que pretende determinar las competencias que posee el estudiante tras el desarrollo del proceso de aprendizaje. En este tipo de evaluación se recoge información mediante un instrumento, prueba, procedimiento o actividad para poder describir e identificar las competencias adquiridas por los estudiantes acerca de un dominio de referencia, descrito en la planificación docente inicial. Así, los criterios de evaluación deben correlacionarse con los objetivos/competencias y especificar principalmente un dominio conceptual claro, la adquisición de procedimientos, técnicas, instrumentos, habilidades y destrezas de ejecución profesional y académica, que concretan el nivel de capacidades personales y profesionales necesarias para el ejercicio profesional.

8.2 CRITERIOS DE EVALUACIÓN

Teniendo en cuenta estas premisas, los criterios de evaluación aplicados a Programación Orientada a Objetos son:

- Teoría: examen final. 40% de la asignatura
- Práctica: Tres entregas. 60% de la asignatura
- Primera entrega: 20%
- Segunda entrega 50%
- Tercera entrega: 30%
- Se deben aprobar al menos dos de las tres prácticas para que se haga media
- Trabajos optativos: HASTA 2 PUNTOS (valorable por el profesor en función tanto de la defensa que haga el alumno como de la síntesis, practicidad y relevancia para la asignatura del trabajo)
- Obligatoria la defensa del trabajo en clase: presentación de 10-15 minutos
- Ejemplos de trabajos:
- Uso de STL en C++
- Ejemplo de implementación de persistencia en C++
- Conexión de C++ con Bases de Datos
- eXtreme Programming
- Implementación de la práctica con NCurses

Se debe aprobar la práctica para presentarse a teoría. La nota de teoría y práctica NO COMPENSAN (se deben aprobar por separado). La realización de ejercicios propuestos en clase es útil para decidir nota en caso de duda.

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos de resolución de problemas se ajustan a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre POO cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. Los problemas relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos de resolución de problemas son generalmente ajustados a la naturaleza del problema. • Las destrezas experimentales son generalmente buenas y muestran un análisis y evaluación de los resultados aceptables. • La actuación en las destrezas transferibles es generalmente buena. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las prácticas de laboratorio estándares son usualmente desarrolladas con éxito razonable aunque el significado y análisis de los resultados pueden no ser entendidos completamente. • Las destrezas transferibles están a un nivel básico. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspense	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas relacionados con la asignatura no son, generalmente, resueltos de forma adecuada. • Las prácticas de laboratorio estándares no son usualmente desarrolladas satisfactoriamente y el significado y análisis de los resultados no son entendidos generalmente. • Las destrezas transferibles están a un nivel deficiente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 5. Criterios de evaluación

Queremos hacer notar que para la obtención de matrícula de honor es necesario obtener un sobresaliente alto y hacer un trabajo complementario de calidad.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Programación orientada a Objetos. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)		
OI1	De CIC1 a CIC110 cCIC1 cCIC2	1	2			Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a la programación orientada a objetos.
OI2	De CIC11 a CIC22 cCIC1 cCIC2	1		3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de comprensión, análisis y aplicación de los conceptos relativos a la programación orientada a objetos.
OI3	De CIC1 a CIC22 cCIC1 cCIC2	1	2	3,4	5	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. <u>Criterios:</u> Grado de comprensión conceptual.
OI4	De CIC1 a CIC22 cCIC1 cCIC2	1	2	3,4	5	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. <u>Criterios:</u> Grado de comprensión conceptual.
OI6	De CIC1 a CIC22 cCIC1 cCIC2	1	2	3,4	5	Enseñanza presencial (lección magistral). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. <u>Criterios:</u> Grado de comprensión conceptual.
OI6	De CIC1 a CIC21	1	2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 6. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)		
OI5	CIM1 CIM2 cCIM1 cCIM2 cCIM3		2	3,4	5	Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOI1	CIM1 CIM2 cCIM1 cCIM2 cCIM3	1	2	3,4	5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental, eficiencia y precisión en la resolución de problemas usando programación orientada a objetos.
cOI2	CIT1 cCIT1	1	2	3,4	5	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/realización de prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. <u>Actividades</u> en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI3	CIL1 cCIL1	1	2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de rigurosidad en las explicaciones de los procedimientos aplicados.
cOI4	cCIL2	1	2	3,4	5	Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Y complementario. <u>Criterios:</u> Grado de conocimiento de los distintos términos de programación orientada a objetos, en castellano y/o valenciano y en inglés.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)		
cO15	De CIM1 a CIM2 cCIM3	2	2	3,4	5	Enseñanza no presencial (aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, análisis y aplicación de los conceptos.

Cuadro 8. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)		
cOIP1	CIPTC1 cCIPTC1 cCIPTC2		2	3,4	5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios/ prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2		2	3,4	5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios/prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR2		2	3,4	5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios/ prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 9. Análisis de coherencia: competencias interpersonales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS				PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)		
cOS1	cCS1 cCS4 cCS5	1	2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3	1	2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los problemas resueltos con programación orientada a objetos.
cOS3	cCS1 cCS2 cCS4 cCS5	1	2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2		2	3,4	5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los problemas resueltos con programación orientada a objetos

Cuadro 10. Análisis de coherencia: competencias sistémicas

10. ANEXO A: CLASIFICACIÓN DE LAS ASIGNATURAS DEL PLAN DE ESTUDIOS VIGENTE EN LA UNIVERSIDAD DE ALICANTE SEGÚN SUBCATEGORÍAS DEL LIBRO BLANCO (CONTENIDOS ESPECÍFICOS DE LA INGENIERÍA)

P: Programación

IS: Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes

SO: Sistemas operativos, Sistemas Distribuidos y Redes

IC: Ingeniería de Computadores

Curso: 1		
Tipo: TRONCAL		
Asignatura	Créditos	Área
ÁLGEBRA (9168)	6	-
CÁLCULO INFINITESIMAL (9169)	9	-
ESTADÍSTICA (9162)	6	-
FUNDAMENTOS DE PROGRAMACIÓN I (9171)	6	P
FUNDAMENTOS DE PROGRAMACIÓN II (9172)	6	P
FUNDAMENTOS FÍSICOS DE LA INFORMÁTICA (9167)	10,5	-
INFORMÁTICA BÁSICA (9165)	12	IC
MATEMÁTICA DISCRETA (9170)	6	-
Tipo: OBLIGATORIA		
Asignatura	Créditos	Área
LÓGICA COMPUTACIONAL (9188)	6	-

Cuadro 11. Clasificación de asignaturas de 1^{er} curso

Curso: 2		
Tipo: TRONCAL		
Asignatura	Créditos	Área
BASES DE DATOS I (9164)	9	IS
COMPUTABILIDAD (9177)	4,5	P
ESTRUCTURAS DE COMPUTADORES (9166)	6	IC
LENGUAJES, GRAMÁTICAS Y AUTÓMATAS (9176)	4,5	P
PROGRAMACIÓN Y ESTRUCTURAS DE DATOS (9163)	9	P
SISTEMAS OPERATIVOS I (9174)	4,5	SO
Tipo: OBLIGATORIA		
Asignatura	Créditos	Área
HERRAMIENTAS DE PROGRAMACIÓN (9452)	6	P
LENGUAJES Y PARADIGMAS DE PROGRAMACIÓN (9189)	6	P
PROGRAMACIÓN ORIENTADA A OBJETOS (9190)	4,5	P

Cuadro 12. Clasificación de asignaturas de 2^o curso

Curso: 3			
Tipo: TRONCAL			
Asignatura		Créditos	Área
DISEÑO Y ANÁLISIS DE ALGORITMOS (9173)		6	P
SISTEMAS OPERATIVOS II (9175)		4,5	SO
Tipo: OBLIGATORIA			
Asignatura		Créditos	Área
BASES DE DATOS II (9192)		6	IS
DISEÑO Y PROGRAMACIÓN AVANZADA DE APLICACIONES (9193)		4,5	IS
FUNDAMENTOS DE ARQUITECTURAS DE COMPUTADORES (9194)		6	IC
GRÁFICOS POR COMPUTADOR (9191)		4,5	P
SISTEMAS DE INFORMACIÓN EN LA EMPRESA I (9195)		6	-
SISTEMAS DE INFORMACIÓN EN LA EMPRESA II (9196)		6	-

Cuadro 13. Clasificación de asignaturas de 3^{er} curso

Curso: 4			
Tipo: TRONCAL			
Asignatura		Créditos	Área
ANÁLISIS Y ESPECIFICACIÓN DE SISTEMAS DE INFORMACIÓN (9179)		6	IS
ARQUITECTURA E INGENIERÍA DE COMPUTADORES (9178)		12	IC
FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL (9182)		4,5	IS
INGENIERÍA DEL SOFTWARE I (9180)		6	IS
PROCESADORES DE LENGUAJE (9184)		9	P
REDES (9185)		7,5	SO
SISTEMAS DE TRANSPORTE DE DATOS (9186)		6	SO
TÉCNICAS DE INTELIGENCIA ARTIFICIAL (9183)		4,5	IS
Tipo: OBLIGATORIA			
Asignatura		Créditos	Área
ALGORITMIA AVANZADA (9197)		4,5	P

Cuadro 14. Clasificación de asignaturas de 4^o curso

Curso: 5			
Tipo: TRONCAL			
Asignatura		Créditos	Área
INGENIERÍA DEL SOFTWARE II (9181)		6	IS
SISTEMAS INFORMÁTICOS (9187)		15	-
Tipo: OBLIGATORIA			
Asignatura		Créditos	Área
SISTEMAS OPERATIVOS EN RED (9198)		9	SO

Cuadro 15. Clasificación de asignaturas de 5^o curso

11. ANEXO B: ESTIMACIÓN DE CRÉDITOS ECTS PARA POO

Tal y como podemos ver en la siguiente tabla, en el plan de estudios actual, conducente a la obtención del título de Ingeniero en Informática por la Universidad de Alicante, la carga docente acumulada asignable a los Contenidos Formativos Comunes suma un total de 255 créditos (de los 364,5 que componen la titulación), divididos de la siguiente forma:

División de créditos actuales según categorías de los cfc	Créditos	%T/O
Fundamentos científicos	43,5	17,06%
Contenidos generales ing	12	4,71%
Proyecto	15	5,88%
Contenidos específicos ing	184,5	72,35%
TOTAL	255	100%

Cuadro 16. *División de créditos actuales según categorías de los Contenidos Formativos Comunes*

Dentro de los Contenidos Específicos, tenemos el siguiente desglose en las cuatro grandes subcategorías:

Programación	70,5	38,21%
Ingeniería Sw, SI y Stmas Inteligentes	46,5	25,20%
SO, SD y Redes	31,5	17,07%
Ingeniería Computadores	36	19,51%

Cuadro 17. *Desglose de créditos por subcategorías*

Si suponemos que los porcentajes de la tabla anterior se van a mantener en la nueva distribución de créditos ECTS por asignatura, y teniendo en cuenta el cálculo de 140,4 ECTS (39% del total de 240 créditos) que es el máximo recomendado por el libro blanco (tomando mínimos para el resto de categorías) para el conjunto de contenidos específicos, obtenemos los siguientes créditos para cada subcategoría:

CONTENIDOS ESPECÍFICOS ING	39,00%	140,4
Programación	38,21%	53,65
Ingeniería Sw, SI y Stmas Inteligentes	25,20%	35,39
SO, SD y Redes	17,07%	23,97
Ingeniería Computadores	19,51%	27,40

Cuadro 18. *Desglose de créditos*

Si ahora adaptamos los créditos de las asignaturas incluidas en la subcategoría de Programación a esos 53,65 ECTS, mediante una sencilla regla de tres nos quedaría:

ASIGNATURA	Cred Act	ECTS
Fundamentos programación i	6	4,77
Fundamentos programación ii	6	4,77
Herramientas de programación	4,5	3,58
Diseño y análisis de algoritmos	6	4,77
Algoritmia avanzada	4,5	3,58
Computabilidad	4,5	3,58
Lenguajes, gramáticas y autómatas	4,5	3,58
Procesadores del lenguaje	9	7,15
Lenguajes y paradigmas de programación	4,5	3,58
Programación orientada a objetos	4,5	3,58
Diseño y programación avanzada de aplicaciones	4,5	3,58
Programación y estructuras de datos	9	7,15
TOTAL	67,5	53,65

Cuadro 19. *Relación entre créditos actuales y créditos ECTS*

Este cálculo proporciona una asignación de 3,58 créditos ECTS a la asignatura de Programación Orientada a Objetos (unas 108 horas), lo que es ligeramente inferior a las 122 horas calculadas en la sección 6. Aún más, el plan de estudios actual se basa en un decreto que no está actualizado, y por tanto no recoge la importancia que los currícula más actuales reconocen al paradigma Orientado a Objetos. Es por ello que consideramos necesario, con el fin de reconocer el esfuerzo que el alumno debe realizar para la asimilación de esta materia, y con el objeto de impartirla con la profundidad y rigor que los Currícula Internacionales sugieren, aumentar significativamente la carga docente de esta asignatura en los nuevos planes de estudio, y asignarle un mínimo de 6 créditos ECTS (es decir, entre 150 y 180 horas de trabajo, si asumimos que cada crédito ECTS equivale a 25-30 horas).

Además, debemos tener en cuenta que la carga lectiva actual de la titulación de Ingeniería Informática en la UA es claramente excesiva (364,5 créditos frente a los poco más de 300 de otras universidades, i.e. un 21% superior). Aún más, con la conversión realizada en este trabajo, hemos encajado la docencia impartida actualmente en 5 años, como mucho, en cuatro años (tres de asignaturas más uno de trabajos tutorizados) de la nueva titulación. Una alternativa (que no hemos aplicado por desconocer cuál va a ser la política a aplicar en los títulos de Máster) sería haber considerado no 240 ECTS (que pensamos que será lo que se convalide a los Ingenieros Informáticos) sino 300 (grado+máster) para aplicar los porcentajes. Creemos que es absolutamente necesaria una reflexión profunda en el seno de la universidad para discernir cuáles son en verdad contenidos fundamentales y cuáles se podrían trasladar a un futuro título de Máster, para de este modo no sobrecargar al alumno con contenidos. En esta futura reorganización de asignaturas, la Programación Orientada a Objetos es claramente un contenido del título de grado, ya que su temario supone la primera aproximación específica a este paradigma de programación.

GUÍA DOCENTE DE PROGRAMACIÓN Y ESTRUCTURAS DE DATOS

Jesús Peral Cortés; Sergio Luján Mora; Antonio Ferrández Rodríguez

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
{jperal;slujan;antonio}@dlsi.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

El área de conocimiento que enmarca la disciplina de Programación y Estructuras de Datos es Lenguajes y Sistemas Informáticos. Ésta es un área donde se engloban materias como ingeniería del software, lenguajes informáticos y naturales, sistemas de información, inteligencia artificial y bases de datos. Así, según la Nomenclatura internacional de la UNESCO para los campos de ciencia y tecnología⁴, la asignatura de Programación y Estructuras de Datos formaría parte del campo científico de las matemáticas (12) y, más concretamente, dentro de él en la ciencia de los ordenadores (1203) que engloba a la teoría de la programación (120324) y a los lenguajes de programación (120323). Según otros autores, podría formar parte del campo científico de las ciencias tecnológicas (33) y, dentro de él, en el campo de la tecnología de los ordenadores (3304), sin poder especificar el campo concreto, sino que se englobaría en otros (330499), definido para estos casos.

1.1.1 Análisis de la disciplina

El problema fundamental en el diseño e instrumentación de grandes proyectos o aplicaciones informáticas es reducir su complejidad. Los atributos que permiten obtener esta característica deseable son los siguientes:

⁴ Propuesta en 1973 y 1974 por las Divisiones de Política Científica y de Estadística de la Ciencia y Tecnología de la UNESCO. En 1983, por Resolución de 23 de septiembre de 1983 (B.O.E. n. 14 de octubre), pasa a ser la clasificación utilizada por el Ministerio de Educación y Ciencia.

Legibilidad. Se trata de una cualidad imprescindible, especialmente cuando se trabaja en equipo, que facilitará la consecución del resto de las características deseables. Al margen de cuestiones de estilo, algunas propiedades que influyen en la legibilidad son la simplicidad, la estructuración y la elección de estructuras de datos adecuadas.

Corrección. Esta propiedad constituye un criterio fundamental de la programación. Los programas deberán estar conformes con la especificación para cualquier condición. En este sentido se manejan técnicas de especificación y derivación que dan lugar a programas correctos.

Eficiencia. El coste de un programa se mide en términos de los recursos consumidos. Se trata de hacer estudios sobre la complejidad temporal y espacial de los algoritmos y comparar soluciones alternativas.

Facilidad de mantenimiento. Cabe destacar la importancia de que un programa pueda modificarse para ser adaptado a nuevos requerimientos. Esta propiedad está íntimamente ligada a la que se menciona seguidamente y, por tanto, a la descomposición modular de los programas.

Reutilización. Éste es uno de los factores que claramente reducen el tiempo y el coste de la producción de programas. El objetivo de obtener programas reutilizables nos lleva a dos consideraciones: por un lado a la necesidad de diseñar algoritmos que sean generales dentro de lo posible, por otro lado, al hecho de que la reutilización redundará en la composición modular y en la capacidad de identificar los problemas a resolver con los ya resueltos.

A través de la abstracción de datos, se pueden obtener estos objetivos (Liskov, 1987). Además, la abstracción de datos permitirá el cambio de representación de los datos sin afectar por ello a los programas que los utilicen. Es además particularmente importante porque simplifica la estructura de los programas que la usan debido a que presenta una interfaz de alto nivel.

Pero ¿qué es la abstracción? Ésta se podría definir como:

Una idea general que se concentra en las cualidades esenciales de algo, en lugar de en las realizaciones concretas o casos.

La evolución de la asignatura de Programación y Estructuras de Datos se aprecia en el desarrollo de los mecanismos de abstracción utilizados en programación a lo largo del tiempo, los cuales se pueden ordenar del siguiente modo:

Procedimiento, módulo, tipos de datos, tipo abstracto de dato y objeto.

El primer mecanismo de abstracción en programación fue el concepto de procedimiento. Un procedimiento realiza cierta tarea o función, que será realizada en el programa a través de una llamada a dicho procedimiento. Para utilizarlo, el programador sólo se preocupa de qué hace y no cómo está instrumentado. Fortran es el primer lenguaje que soporta este mecanismo, apareciendo posteriormente Algol-60, Algol-68, C y Pascal.

Con el paso de los años, el énfasis en el diseño de programas fue cambiando del diseño basado en procedimientos a la organización de los datos. En la década de los 60, los tipos de datos se podían definir como:

Conjunto de valores que sirve de dominio de ciertas operaciones

Los tipos de datos permitían clasificar los objetos de los programas (variables, parámetros y constantes) y determinar los valores que podían tomar y las operaciones que se les podía aplicar. Sin embargo, los tipos de datos eran insuficientes en el desarrollo de software a gran escala. En el desarrollo de estos grandes proyectos o aplicaciones informáticas sólo existían las restricciones impuestas por el compilador y ocurría que los tipos de datos y las operaciones se encontraban mezclados. Esta idea surgió a finales de los años 70 cuando empezó a considerarse que un tipo de datos no debía definirse solamente a partir de un conjunto de valores admisibles, sino que el concepto de tipo debía estar íntimamente ligado a las operaciones que tenían significado para sus objetos. La segunda idea fue que la protección de los datos no debía dejarse al buen criterio del programador o al papel estratégico de los sistemas operativos, sino que debería resolverse en el propio lenguaje. Las violaciones de la protección se deberían detectar en la fase de compilación, en lugar de hacerse en la de ejecución, con consecuencias potencialmente más graves (Bishop, 1989).

Para evitar estos problemas aparece el concepto de tipo abstracto de datos, propuesto por Guttag en 1976 (Guttag et al., 1976):

Un tipo abstracto de datos es una colección de valores y operaciones que se definen mediante una especificación que es independiente de cualquier representación

En esta definición se introduce el concepto de abstracto indicando que la manipulación de los objetos o valores de un tipo sólo depende del comportamiento descrito en su especificación y es independiente de su representación e instrumentación. Dicho de otra forma, la abstracción de datos permite separar el qué (especificación) del cómo (representación e instrumentación 2) (Liskov, 1987).

Los tipos abstractos de datos se definen mediante entidades abstractas y operaciones que se encapsulan de forma tal que la única vía para acceder o para modificar las entidades es a través de las operaciones abstractas. Esta idea ya estaba implícita en el lenguaje de programación SIMULA 67, en el que la designación sintáctica clase denota una colección de esas operaciones. A este encapsulamiento se le llama Tipo Abstracto de Dato (TAD), haciendo referencia con el calificativo abstracto, a la idea de que, cuando se especifica, lo único fundamental son las cualidades, es decir, el qué se puede hacer, y no cómo se hace.

Con posterioridad a SIMULA 67 aparecieron otros lenguajes que también incorporaban la noción de abstracción de datos, como Alphard (Wulf et al., 1976), CLU (Liskov et al., 1978), Modula-2 (Wirth, 1982), ADA (Barnes, 1982),

C++ (Stroustrup & Ellis, 1990; Stroustrup, 1993; Stroustrup, 2002), Eiffel (Switzer, 1993) y Java (Arnold & Gosling, 1997). Estos lenguajes permiten el encapsulamiento de la representación del TAD; al mismo tiempo, proporcionan la posibilidad de ocultar la información, característica que es fundamental para conseguir que sólo se manipulen los objetos de un TAD mediante una interfaz, y que nunca se pueda acceder a la instrumentación del mismo desde el exterior. ADA, C++ y Eiffel proporcionan, además un nivel mayor de abstracción al permitir declarar tipos abstractos de datos genéricos, definiendo las operaciones independientemente de los elementos que van a constituir el TAD. Esta característica, llamada genericidad, ofrece la posibilidad de definir varios conceptos abstractos semejantes desarrollando un único TAD parametrizado, que se podrá instanciar con diferentes tipos de elementos.

La especificación sirve para expresar las cualidades de un tipo abstracto de datos, es decir, qué debe producir, y no cómo lo hace. Es una descripción breve y precisa de los datos y funciones de un módulo. La especificación nos va a permitir separar la realización del uso, es un enlace entre el usuario y el implementador. La especificación debe proporcionar al usuario toda la información necesaria para que pueda utilizar el programa correctamente, y nada más. También debe proporcionar al implementador toda la información, acerca del uso que se pretende, necesaria para completar el programa y ninguna información adicional más, en concreto, no debe ser portadora de ninguna información sobre la estructura de los programas de llamada.

Para que sea útil, una especificación ha de ser:

- Precisa: sólo tiene que decir lo realmente imprescindible.
- General: adaptable a diferentes contextos.
- Legible: que sirva como instrumento de comunicación entre el especificador y los usuarios del tipo.
- No ambigua: que evite problemas de interpretación.

La especificación de los Tipos Abstractos de Datos se ha planteado habitualmente de dos formas: informal o formal. La informal no está sujeta a ninguna notación ni estructura y las características del objeto a especificar se describen en lenguaje natural. Sin embargo, la especificación debe ser clara y sin ambigüedades, ya que frecuentemente es el único medio de comunicación entre los usuarios y los diseñadores. La especificación en lenguaje natural tiene la ventaja de ser muy fácil de leer debido a la gran amplitud de diferentes puntos de vista, y los diferentes niveles de detalle a los que se quiera llegar. El problema es la ambigüedad, que conduce a diferentes interpretaciones dando lugar a multitud de errores cuando se diseñan sistemas.

La especificación formal evita estas ambigüedades, basándose, principalmente, en un lenguaje matemático, como por ejemplo las dos propuestas siguientes:

1. La propuesta en (Dahl et al., 1972) en la que se plantea la introducción del concepto de módulo independiente, donde se lleva a cabo la realización de las operaciones asociadas a un tipo. La invocación del módulo sólo permite utilizar las operaciones, estando ocultos los detalles asociados a la instrumentación de éstas. Además, el control de las pre y post condiciones introduce la abstracción en el nivel de la instrumentación.
2. La aproximación algebraica (Guttag et al., 1976), en la que la especificación de un TAD consta de dos partes: sintaxis, en la que se determinan los conjuntos de objetos que se van a utilizar y el perfil de las operaciones, y la semántica, es decir, un conjunto suficientemente completo de axiomas que especifique cómo interactúan las funciones. Se denomina algebraica a esta aproximación por fundamentarse en la teoría de álgebras heterogéneos (Horebeek, 1985).

Una vez obtenida la especificación del TAD, y la herramienta para poder instrumentarla, el programador decide qué estructura de datos utilizará para representar el tipo, y qué algoritmos utilizará para realizar las operaciones determinadas por la especificación, es decir, la instrumentación o implementación de un TAD. Con el término estructuras de datos nos referimos a la forma en que está organizada lógicamente la información en un programa, esto es, cómo se instrumentan los objetos abstractos, de acuerdo a las restricciones dadas por el lenguaje de programación elegido. Por tanto, dependiendo del punto de vista, un objeto se caracteriza por su tipo (para el usuario) o por su estructura (para el que lo instrumenta). Típicamente, una estructura de datos se construye a partir de los tipos de datos disponibles en el lenguaje. En cuanto a los algoritmos seleccionados, deben ser tales que proporcionen a las operaciones el comportamiento definido por la especificación. Además es aconsejable hacer un estudio previo de las complejidades espacial y temporal de los algoritmos con el fin de lograr la mayor eficiencia posible.

La implementación de un tipo abstracto de datos es totalmente transparente a los usuarios del tipo y no se debe escribir hasta haber determinado claramente su especificación. El cambio de una implementación por otra que respete el comportamiento definido en la especificación podrá realizarse cuando las circunstancias lo requieran, sin que afecte a los usuarios del TAD. Para que una implementación sea útil, ha de ser:

- Estructurada: para facilitar su desarrollo.
- Eficiente: para optimizar el uso de recursos del ordenador.
- Legible: para facilitar su modificación y mantenimiento.

Por último, comentar el auge que está teniendo actualmente la Programación Orientada a Objetos (POO) (Stroustrup, 1988) debido a las deficiencias del resto de técnicas de desarrollo, sobre todo, para la realización de grandes proyectos. La POO es una metodología de programación práctica y potente

que fomenta el diseño modular y la reutilización de código, además de otras características deseables como el encapsulamiento, portabilidad y mantenimiento (Cardelli & Wegner, 1985). Una de las principales características es la de permitir la abstracción de datos y la definición de nuevos tipos de datos cuyo comportamiento se define de una forma abstracta sin tener en cuenta cómo se instrumentan. Se dice que un lenguaje de programación es orientado a objetos si soporta la abstracción de datos, las clases de objetos y la herencia de clases.

Los principios de la programación orientada a objetos son:

- Encapsulamiento. En un sistema orientado a objetos se manipulan objetos de ciertas clases. Estas clases son un instrumento para representar tipos abstractos de datos. Concretamente, el concepto de clase permite agrupar conjuntamente como una única entidad a los datos, así como a las operaciones que permiten trabajar con ellos. Contienen toda la información necesaria para construir ejemplares de ellas (objetos).
- Herencia. Este mecanismo permite la creación de una nueva clase que sea una extensión o especialización de otra ya existente. Los objetos de la nueva clase pueden acceder a variables y métodos contenidos en la clase previamente definida, sin tener que volver a realizar estas definiciones.
- Polimorfismo. Es una propiedad según la cual un nombre (de variable) puede referirse a objetos de distinta clase, relacionados todos ellos por alguna clase base común. Esto permite que, por ejemplo, cualquier objeto referido por esta variable sea capaz de responder a un conjunto de operaciones de forma distinta.

Actualmente, existe un gran número de lenguajes de programación que permiten este tipo de metodología. Algunos de estos lenguajes son Smalltalk (Goldberg & Robson, 1985), Objective C (Cox, 1991), C++, Eiffel y Java.

1.1.2 Comparación con los modelos curriculares internacionales en Informática

Para justificar el programa propuesto en la asignatura de Programación y Estructuras de Datos, a continuación se analizan las propuestas curriculares de los principales organismos internacionales. Se analiza el currículo ACM/IEEE y el currículo de ACM/AIS/AITP. Posteriormente, se realiza la comparación con el programa aquí propuesto.

Modelo curricular de ACM/IEEE

La Association for Computing Machinery (ACM), fundada en 1947, junto con la Institute of Electrical and Electronics Engineers (IEEE) Computer Society (IEEE-CS), fundada en 1946, son dos de las asociaciones profesionales internacionales más importantes en el campo de la informática.

Estas asociaciones se han caracterizado por la publicación periódica de informes cuyo propósito básico ha sido establecer criterios y efectuar recomendaciones para el diseño de planes de estudio de distintos niveles en Informática.

En concreto, el Curriculum Committee on Computer Science de la ACM publicó sus primeras recomendaciones en marzo de 1968, con el título Curriculum 68 (ACM, 1968), y en marzo de 1978, se publica una versión actualizada de las mismas con el título Curriculum 78 (ACM, 1978). La sociedad IEEE por su parte, en la publicación Model Program in Computer Science and Engineering de diciembre de 1983 (IEEE, 1983) propone también unas áreas de conocimiento que combinadas total o parcialmente pueden constituir diversas especialidades o currículos para estudios de Informática.

En la primavera de 1988, la ACM y el IEEE-CS formaron el Joint Curriculum Task Force, con objeto de presentar recomendaciones para el diseño e implementación del currículo para aspirantes a licenciados en informática. Como resultado de estos esfuerzos se publica un primer informe titulado Computing as a Discipline (Denning et al., 1989a; Denning et al., 1989b).

De acuerdo a este informe, la informática se concibe basada en tres paradigmas:

Teoría, de fuerte base matemática y, por tanto, contempla los aspectos de ciencia formal de la informática.

En el desarrollo de una teoría coherente y válida se contemplan cuatro pasos:

- Definición. Caracterización de los objetos de estudio.
- Teorema. Hacer hipótesis sobre las posibles relaciones entre los objetos.
- Demostración. Determinando qué relaciones son ciertas.
- Interpretación de los resultados.

Abstracción. Es propio de las ciencias experimentales. La investigación de un fenómeno se caracteriza por los pasos:

- Proposición de una hipótesis.
- Construcción de un modelo y realización de predicciones.
- Diseño de experimentos y recogida de datos.
- Análisis de resultados.

Diseño, propio de las ingenierías (ciencias aplicadas). La construcción de un sistema para resolver un problema dado se caracteriza por los pasos:

- Establecimiento de los requisitos.
- Establecimiento de las especificaciones.
- Diseño y realización del sistema.
- Prueba del sistema.

La influencia de los tres paradigmas está interrelacionada de forma que no se puede hablar de que alguno de ellos prime sobre los demás. De todo lo reseñado anteriormente, la informática puede definirse como el estudio sistemático de procesos algorítmicos que describen y transforman la información, su teoría, análisis

sis, diseño, eficiencia, implementación y aplicación. La cuestión fundamental de la informática es: ¿Qué puede ser automatizado (eficientemente)?

Dada la extensión de la disciplina, ésta se divide en subáreas, cada una de las cuales tiene una base teórica sustancial, una abstracción y diseño significativos, así como un contenido importante de instrumentación. Así se propone un paradigma para la enseñanza de la informática que satisface las normas científicas tradicionales y enfatiza el desarrollo de competencia en la materia e integra armoniosamente teoría, experimentación y diseño.

Modelo curricular de ACM/IEEE (1991).

Tras este primer informe se propone el Computing Curricula, 1991 (ACM/IEEE, 1991) que contiene las recomendaciones de currículo en la disciplina de la computación. Este informe incluye programas para titulaciones de ciencia de la computación, ingeniería de la computación, ciencia e ingeniería de la computación y otros títulos similares.

Este informe hace uso de las diferentes áreas y subáreas temáticas definidas en el informe anterior, a las que se añaden las siguientes características:

Dentro de las áreas temáticas se establecen los requisitos comunes, conocimientos que deben considerarse como esenciales para cualquier estudiante. Estas exigencias son formalizadas en las llamadas unidades de conocimiento, las cuales permiten una mayor flexibilidad en la definición de una asignatura, curso o plan de estudios. Los requisitos comunes junto con una estimación de las horas lectivas que incluiría su enseñanza se describen a continuación:

- AL: Algoritmos y estructuras de datos (aprox. 47 horas lectivas).
- AR: Arquitectura (aprox. 59 horas lectivas).
- AI: Inteligencia artificial y robótica (aprox. 9 horas lectivas).
- DB: Bases de datos y recuperación de información (aprox. 9 horas lectivas).
- HU: Comunicación hombre-máquina (aprox. 8 horas lectivas).
- NU: Computación numérica y simbólica (aprox. 7 horas lectivas).
- OS: Sistemas operativos (aprox. 31 horas lectivas).
- PL: Lenguajes de programación (aprox. 46 horas lectivas).
- SE: Metodología e ingeniería del software (aprox. 44 horas lectivas).
- SP: Aspectos sociales, éticos y profesionales (aprox. 11 horas lectivas).

Se mantienen los tres paradigmas: teoría, abstracción y diseño, aunque ahora se les llama procesos.

Se definen los conceptos recurrentes que pueden aparecer en cualquier área temática y en cualquier proceso, permitiendo obtener un factor común en la definición de las asignaturas. Los conceptos recurrentes son: enlace o binding (asociaciones proceso-procesador, creación de instancias concretas a partir de descripciones abstractas, etc.), complejidad de grandes problemas, modelos conceptuales y formales, consistencia y completitud, eficiencia, evolución, niveles

de abstracción, ordenación en el espacio, ordenación en el tiempo, reutilización, seguridad, y consideraciones (trade-o) y consecuencias. Además, establece la necesidad de la obtención de conocimientos referentes a los hechos culturales, sociales, éticos y legales que se relacionan con la Informática.

Vuelve a insistir en el papel primordial de los laboratorios, los cuales demuestran la aplicación del diseño, implementación y pruebas de los sistemas hardware y software. Diferencia entre laboratorio abierto (no supervisado, con o sin instrumentación para su realización) y cerrado (supervisado, incluye el uso de instrumentación necesaria para su realización).

Hace referencia a otras experiencias educacionales como el trabajo en grupo, aduciendo que tal experiencia añade amplitud, profundidad y realismo al currículo, además de ser un importante conductor del desarrollo de habilidades de comunicación tanto oral como escrita.

El documento señala que uno de los objetivos de los programas de licenciatura en informática debería ser preparar a los estudiantes para aplicar sus conocimientos a problemas específicos y concretos y producir soluciones. Esto incluye la capacidad de definir un problema claramente; determinar su tratabilidad; determinar cuándo es apropiada la consulta con expertos externos; evaluar y elegir una estrategia adecuada de solución; estudiar, especificar, diseñar, implementar, probar, modificar y documentar esa solución; evaluar alternativas y realizar un análisis de riesgo sobre ese diseño; integrar tecnologías alternativas en esas soluciones y comunicar esas soluciones a los colegas, profesionales de otros campos y al público en general, así como la habilidad para trabajar dentro de un ambiente de grupo durante todo el proceso de solución del problema.

Se proporciona, además, en una sección aparte, el material avanzado y suplementario para profundizar en líneas de intensificación de cada una de las áreas fundamentales.

Dentro de los requisitos comunes especificados se destaca en concreto el de AL: Algoritmos y estructuras de datos y PL: Lenguajes de programación para la asignatura de PED. A continuación se describen cada uno de ellos:

AL: Algoritmos y estructuras de datos. Hay aproximadamente 47 horas lectivas recomendadas para este requisito común. Éste se divide en 9 unidades de conocimiento, siendo las unidades AL1, AL2 y AL3 las más relacionadas con la asignatura de PED:

AL1: Estructuras de datos básicas

- Descripción: introducción a la definición, implementación y aplicación de estructuras de datos básicas. Éstas incluyen listas, vectores, tablas, pilas, colas, árboles y grafos.

- Conceptos recurrentes: Modelos conceptuales y formales, consistencia y completitud, eficiencia, niveles de abstracción, reutilización, consideraciones y consecuencias.
- Temas lectivos: (mínimo 13 horas)
 1. Definición de estructuras de datos básicas.
 2. Uso de estructuras de datos básicas.
 3. Implementaciones secuenciales y enlazadas. Estudio de las complejidades espacial y temporal en la resolución de problemas.
- Prácticas de laboratorio:
 1. Uso de pilas, colas y otras estructuras de datos en aplicaciones típicas de Informática, tales como vuelta atrás, análisis sintáctico, simulaciones discretas, etc.
 2. Estudio de diferentes estrategias de implementación, como por ejemplo listas enlazadas frente a vectores.
 3. Diseño de nuevas estructuras de datos y su implementación.
 4. Los estudiantes se familiarizan con el uso apropiado de las estructuras de datos para la resolución de problemas, y la creación de implementaciones alternativas de las estructuras de datos.
- Requisito para: OS7, PL6.
- Relaciones: AL2, PL3.

AL2. Tipos abstractos de datos

- Descripción: introducción al concepto de tipo abstracto de dato y su motivación como una metodología para separar los detalles de implementación de las aplicaciones que los usan.
- Conceptos recurrentes: complejidad de grandes problemas, modelos conceptuales y formales, niveles de abstracción, reutilización.
- Temas lectivos: (mínimo 2 horas)
 1. Propósito de los tipos abstractos de datos.
 2. Implementación de los tipos abstractos de datos en un lenguaje de alto nivel, con ejemplos.
- Prácticas de laboratorio:
 1. Desarrollo de un programa usando un tipo abstracto de dato proporcionado. Ejecutarlo posteriormente usando implementaciones alternativas.
 2. Los estudiantes se familiarizan con los conceptos de abstracción, modularidad y sustitución de diferentes implementaciones.
- Requisito para: PL11, SE2, SE3.
- Relaciones: AL1, PL2, PL3, PL5.

AL3. Algoritmos recursivos

- Descripción: Introducción a los fundamentos y uso de algoritmos recursivos para la resolución de problemas.

- Conceptos recurrentes: modelos conceptuales y formales, consistencia y completitud, niveles de abstracción, consideraciones y consecuencias.
- Temas lectivos: (mínimo 3 horas)
 - 1 Introducción a los algoritmos recursivos.
 2. Conexión con la inducción matemática.
 3. Comparación entre algoritmos recursivos e iterativos.
- Prácticas de laboratorio:
 1. Realización de la traza de un algoritmo recursivo, con la visualización de variables locales y parámetros.
 2. Desarrollo de una solución recursiva a un problema de programación.
- Requisito para: AL4, AL7, PL1, PL6, PL7.
- Relaciones.

PL: Lenguajes de programación.

Hay aproximadamente 46 horas lectivas recomendadas para este requisito común. Éste se divide en 12 unidades de conocimiento, siendo la unidad PL3 la más relacionada con la asignatura de PED:

PL3: Representación de tipos de datos

- Descripción: tipos de datos elementales y estructurados. Creación de tipos de datos definidos por el usuario y sus aplicaciones.
- Conceptos recurrentes: modelos conceptuales y formales, niveles de abstracción, ordenación en el espacio.
- Temas lectivos : (mínimo 2 horas)
 1. Elección y representación de tipos de datos elementales; entero, real, booleano, carácter.
 2. Especificación y representación de tipos de datos estructurados; vectores, registros, conjuntos, punteros, almacenamiento dinámico de la memoria.
- Prácticas de laboratorio:
 1. Resolución de problemas que requieren variables estructuradas con almacenamiento dinámico de la memoria.
 2. Los estudiantes adquieren experiencia con lenguajes que soportan tipos de datos estructurados con memoria dinámica.
- Requisito para: PL9.
- Relaciones: AL1, AL2, AR3, NU1.

Modelo curricular de ACM/IEEE (2001).

Los enormes cambios que ha sufrido la disciplina informática durante la pasada década ha ocasionado que se haya decidido realizar un nuevo currículo. Además, existe la percepción entre algunos educadores en esta disciplina que la

influencia del currículo ACM/IEEE (1991) no ha tenido tanta influencia como sus predecesores, ya que a pesar de ser éste más detallado que los anteriores, era más difícil de implantar.

La rápida y continua evolución de la disciplina hace necesaria la revisión de algunas áreas que pueden quedar obsoletas en pocos años e introducir nuevas áreas que no se habían tenido en cuenta como tales en currículos previos. Por ejemplo, cuando se desarrolló el currículo de 1991 no se tuvo en cuenta el área de redes e Internet como tal, simplemente era un tema de 3 horas en los requisitos comunes. Después de todo, las redes de ordenadores aún no eran un fenómeno de masas, y la Web era poco más que una idea en la mente de sus creadores. Los avances técnicos experimentados durante la última década han aumentado la importancia de algunos temas, tales como:

- WWW y sus aplicaciones.
- Tecnología de redes, particularmente las basadas en TCP/IP.
- Gráficos y multimedia.
- Sistemas integrados.
- Bases de datos relacionales.
- Interoperabilidad.
- Programación Orientada a Objetos.
- El uso de interfaces de programadores de aplicaciones sofisticados.
- Interacción hombre-máquina.
- Seguridad del software.
- Seguridad y criptografía.
- Dominio de aplicaciones.

El incremento de estos temas debería de incluirse como requisitos comunes. Desafortunadamente, las restricciones de los programas de licenciatura hace difícil añadir nuevos temas sin quitar alguno de los existentes. A menudo, es imposible cubrir nuevas áreas sin reducir la cantidad de tiempo dedicada a temas tradicionales como lenguajes de programación, semántica formal y análisis numérico.

También han afectado en la enseñanza de la disciplina informática cambios en el contexto cultural y sociológico. Los cambios que han tenido influencia en la naturaleza del proceso educacional son:

- Cambios pedagógicos ocasionados por las nuevas tecnologías.
- Crecimiento del uso del ordenador a nivel general.
- Influencia del crecimiento económico en la tecnología informática.
- Mayor aceptación de la disciplina de la informática como una carrera universitaria.
- Crecimiento en amplitud de la disciplina.

Como consecuencia de los cambios anteriores, a finales de 1998, la Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) y

la Association for Computing Machinery (ACM) establecen el Joint Task Force on Computing Curricula 2001 (CC2001) para revisar las líneas maestras en creación de programas universitarios para la disciplina Informática. Así, en la declaración de intenciones de esta Task Force se establece que: se revisará el Joint ACM and IEEE/CS Computing Curricula de 1991 y se desarrollará una versión revisada y mejorada para el año 2001 que contemplará los últimos desarrollos en tecnologías informáticas de la última década, y que serán soportados durante la próxima década.

El documento final del Joint ACM and IEEE/CS Computing Curricula de 2001 se aprobó el 15 de diciembre de 2001 (ACM/IEEE, 2001). A continuación se exponen las principales conclusiones que se extraen de dicha propuesta en relación a la asignatura de PED.

Según este documento, se siguen manteniendo áreas de conocimiento divididas en unidades que, a su vez, se dividen en tópicos (temas). Además, se mantiene el concepto de requisito común que se implementa estableciendo unidades troncales (unidades esenciales para cualquier alumno que se gradúe en Informática), y se añade el concepto de unidad opcional (unidad de conocimiento que opcionalmente puede incluirse en la elaboración de los Planes de Estudio para un determinado centro de enseñanza).

Las áreas definidas en este currículo son las siguientes:

- AL: Algoritmos y complejidad (31 horas troncales + opcionales).
- AR: Arquitectura y organización (36 horas troncales + opcionales).
- CN: Ciencia computacional y métodos numéricos (horas opcionales).
- DS: Estructuras discretas (43 horas troncales).
- GV: Gráficos y computación visual (5 horas troncales + opcionales).
- HC: Interacción hombre-máquina (6 horas troncales + opcionales).
- IM: Gestión de la información (10 horas troncales + opcionales).
- IS: Sistemas inteligentes (10 horas troncales + opcionales).
- NC: Computación en red (15 horas troncales + opcionales).
- OS: Sistemas operativos (18 horas troncales + opcionales).
- PF: Fundamentos de programación (54 horas troncales).
- PL: Lenguajes de programación (6 horas troncales + opcionales).
- SE: Ingeniería del software (30 horas troncales + opcionales).
- SP: Aspectos sociales y profesionales (16 horas troncales + opcionales).

De acuerdo a esta propuesta, un currículo completo se puede organizar con tres tipos de cursos: introductorios, intermedios y avanzados. Los cursos introductorios se corresponden con los que se imparten en el primer y segundo año, los intermedios se corresponden con los de segundo y tercer año y los avanzados el resto de años. Aunque en los cursos introductorios e intermedios se concentran la mayoría de los tópicos troncales (el equivalente a las materias troncales en el Plan de Estudios), se deja a disposición de los diferentes centros

de enseñanzas la posibilidad de completar estos tópicos con algunos tópicos opcionales (que sería el equivalente a las materias obligatorias del Plan de Estudios).

A continuación nos centraremos en concreto en las áreas y unidades relacionadas con PED: PF: Fundamentos de programación, AL: Algoritmos y complejidad y PL: Lenguajes de programación.

PF: Fundamentos de programación. Esta área está formada por 5 unidades troncales. Destacamos las siguientes:

PF3: Estructuras de datos fundamentales

- Duración mínima: 14 horas.
- Carácter: Troncal.
- Tópicos:
 1. Tipos primitivos.
 2. Vectores.
 3. Registros.
 4. Cadenas y su procesamiento.
 5. Representación de los datos en memoria.
 6. Pilas estáticas.
 7. Almacenamiento dinámico.
 8. Punteros y referencias.
 9. Estructuras enlazadas.
 10. Estrategias de implementación para pilas, colas y tablas hash.
 11. Estrategias de implementación para grafos y árboles.
 12. Estrategias para la correcta elección de las estructuras de datos.
- Objetivos:
 1. Discutir la representación y uso de las primitivas de tipos de datos.
 2. Describir el manejo de estructuras de datos dinámicas.
 3. Describir aplicaciones comunes para cada estructura de datos.
 4. Implementar los distintos tipos de datos en un lenguaje de alto nivel.
 5. Comparar distintas implementaciones de estructuras de datos y su rendimiento.
 6. Escribir programas que usen: vectores, registros, cadenas, listas, pilas, colas y tablas hash.
 7. Comparar y contrastar los costes y beneficios de las estructuras dinámicas frente a las estáticas.
 8. Elegir la estructura de datos apropiada para un problema dado.

PF4: Recursión

- Duración mínima: 5 horas.

- Carácter: Troncal.
- Tópicos:
 - 1.El concepto de recursión.
 - 2.Funciones matemáticas recursivas.
 - 3.Procedimientos recursivos simples.
 - 4.Estrategias de divide-y-vencerás.
 - 5.Backtracking recursivo.
 - 6.Implementación de la recursión.
- Objetivos:
 - 1.Describir el concepto de la recursión y dar ejemplos de su uso.
 - 2.Identificar los casos base y general de un problema definido recursivamente.
 - 3.Comparar las soluciones iterativa y recursiva para problemas elementales, como el factorial.
 - 4.Describir la aproximación divide-y-vencerás.
 - 5.Implementar, comprobar y hacer la traza de funciones y procedimientos simples recursivos.
 - 6.Describir cómo la recursión se puede implementar usando una pila.
 - 7.Discutir problemas para los que el backtracking es una solución apropiada.
 - 8.Determinar cuándo una solución recursiva es apropiada para un problema.

AL: Algoritmos y complejidad. Esta área está formada por 5 unidades troncales y 6 opcionales. Destacamos las siguientes:

AL3: Algoritmos fundamentales para computación

- Duración mínima: 12 horas.
- Carácter: Troncal.
- Tópicos:
 - 1.Algoritmos numéricos simples.
 - 2.Algoritmos de búsqueda secuencial y binaria.
 - 3.Algoritmos de ordenación cuadráticos (selección, inserción).
 - 4.Algoritmos de ordenación logarítmicos (quicksort, heapsort, mergesort).
 - 5.Tablas hash; estrategias para evitar las colisiones.
 - 6.Árboles de búsqueda binarios.
 - 7.Representación de grafos (listas y matrices de adyacencia).
 - 8.Recorridos en profundidad y en anchura.
 - 9.Algoritmos de camino mínimo (algoritmos de Dijkstra y Floyd).
 10. Clausura transitiva (Floyd).
 11. Árbol de extensión de coste mínimo (algoritmos de Prim y Kruskal).
 12. Ordenación topológica.

- **Objetivos:**
 1. Implementación de los algoritmos de ordenación más comunes cuadráticos y logarítmicos.
 2. Diseñar e implementar una función hash, apropiada para una aplicación.
 3. Diseñar e implementar un algoritmo para la resolución de las colisiones de una tabla hash.
 4. Discutir la eficiencia computacional de los principales algoritmos para ordenación, búsqueda y hashing.
 5. Discutir otros factores que influyen en la eficiencia computacional para la elección de un algoritmo (tiempo de programación, mantenimiento, etc.).
 6. Resolver problemas utilizando los algoritmos fundamentales de grafos.
 7. Demostrar las siguientes habilidades: evaluar algoritmos, seleccionarlo de entre varias opciones, justificar la selección e implementar el algoritmo en un contexto de programación.

PL: Lenguajes de programación. Esta área está formada por 6 unidades troncales y 5 opcionales. Destacamos las siguientes:

PL6: Programación orientada a objetos

- Duración mínima: 10 horas.
- Carácter: Troncal.
- Tópicos:
 1. Diseño orientado a objetos.
 2. Encapsulamiento y ocultación de la información.
 3. Separación de comportamiento e implementación.
 4. Clases y subclasses.
 5. Herencia.
 6. Polimorfismo.
 7. Jerarquía de clases.
 8. Colección de clases.
 9. Representación interna de objetos y métodos.
- **Objetivos:**
 1. Justificar la filosofía del diseño orientado a objetos y los conceptos de encapsulamiento, abstracción, herencia y polimorfismo.
 2. Diseñar, implementar, probar y hacer una traza de programas sencillos en un lenguaje de programación orientado a objetos.
 3. Describir cómo las clases soportan el encapsulamiento y la ocultación de la información.
 4. Diseñar, poner en funcionamiento y probar la implementación de una relación es-un entre objetos, usando una jerarquía de clases y herencia.

5. Comparar y contrastar las nociones de sobrecarga en un lenguaje orientado a objetos.
6. Explicar la relación entre la estructura estática de una clase y la estructura dinámica de las instancias de la clase.
7. Describir el manejo de iteradores.

PL9: Sistemas tipo

- Carácter: Opcional.
- Tópicos:
 1. Los tipos de datos como conjunto de valores y sus operaciones.
 2. Tipos de datos: elementales, producto y coproducto, algebraicos, recursivos, funciones, parametrizados.
 3. Modelos para comprobar los tipos.
 4. Modelos semánticos de tipos definidos por el usuario: tipos abstractos de datos.
 5. Polimorfismos.
 6. Algoritmos para la comprobación de tipos.
- Objetivos:
 1. Formalizar la noción de tipo.
 2. Describir los tipos de datos elementales.
 3. Explicar el concepto de tipo abstracto de datos.
 4. Reconocer la importancia de los tipos de datos para la abstracción y la seguridad.
 5. Diferenciar entre tipos estáticos y dinámicos.
 6. Diferenciar entre declaraciones de tipos e inferencias de tipos.
 7. Evaluar lenguajes con respecto a los tipos.

PL10: Semántica de los lenguajes de programación

- Carácter: Opcional.
- Tópicos:
 1. Semántica informal.
 2. Semántica formal.
 3. Semántica denotacional.
 4. Semántica axiomática.
 5. Semántica operacional.
- Objetivos:
 1. Explicar la importancia de la semántica formal.
 2. Diferenciar entre semántica formal e informal.
 3. Describir las diferentes aproximaciones a la semántica informal.
 4. Describir las diferentes aproximaciones a la semántica formal.

Modelo curricular de ACM/AIS/AITP en Sistemas de Información (IS'97)

Este modelo curricular es uno de los que mayor número de revisiones ha sufrido (hasta un total de siete, incluida la actual) desde que se publicaron los primeros modelos curriculares en Sistemas de Información a principios de los 70; los editores lo presentan como un modelo válido hasta el año 2005. Los participantes elegidos para su desarrollo provienen de todos los ámbitos de interés en los que los Sistemas de Información tienen cabida: comunidades internacionales de la industria, las empresas, las finanzas y las universidades, incluyendo tanto a las que ofertan títulos de 2 como de 4 años (lo que en España equivale a Ingeniero Técnico e Ingeniero en Informática).

El informe inicial en el que se basa este modelo curricular fue revisado en 11 encuentros americanos e internacionales, en los que participaron alrededor de 1000 personas pertenecientes tanto a la industria como a la academia, hasta su aprobación definitiva en 1997 por las tres asociaciones colaboradoras: ACM, AIS (Association for Information Systems) y AITP (Asociación for Information Technology Professionals), conocida anteriormente como DPMA y asociación que más esfuerzos ha realizado en el diseño de modelos curriculares en los Sistemas de Información a todos los niveles (1981, 1984, 1985, 1990, 1994).

El informe IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (ACM-AIS-AITP, 1997) está fuertemente motivado por la creciente demanda en la industria de Estados Unidos y Canadá de profesionales experimentados en la gestión de recursos y servicios de la tecnología de la información y su aplicación a la empresa.

El informe se estructura en 10 capítulos y un total de 8 apéndices. En los primeros se delimita el ámbito de los sistemas de información, justificando este área como un campo de estudio académico por sí mismo, independiente y diferente de las Ciencias de la Computación. Además, al igual que los dos informes previos, se efectúan recomendaciones (en este caso muy detalladas, concisas y bien justificadas) acerca de los recursos materiales (laboratorios, aulas y biblioteca) y humanos (profesorado y personal de laboratorio) necesarios para obtener titulados de calidad en los sistemas de información. En los ocho apéndices, se describe con toda clase de detalles la estructura y composición del modelo curricular (para titulados de grado medio y superior en sistemas de información), aportando datos sobre contenidos detallados, el nivel de competencia de cada uno de los temas de estudio y la métrica utilizada para obtener ese nivel de competencia, entre otros.

Una de las características principales del informe IS'97 es que reflexiona y delimita el ámbito de los Sistemas de Información. Como campo académico, los Sistemas de Información (SI) abarcan tanto la adquisición, desarrollo y gestión

de los recursos y servicios de la tecnología de la información (funciones de los sistemas de información), como el desarrollo y mantenimiento de la infraestructura y los sistemas para su uso en las organizaciones (desarrollo de sistemas). Así, los profesionales en los sistemas de información tienen bajo su responsabilidad, por un lado, el desarrollo, implementación y gestión de una infraestructura de tecnología de información (computadores y comunicaciones), datos (internos y externos) y la organización de grandes sistemas, y por otro, la incorporación de nueva tecnología a la estrategia de las organizaciones, asesorándoles en su planificación y puesta en marcha.

Por otra parte, el IS'97 delimita perfectamente el ámbito de los sistemas de información frente a las ciencias de la computación. El contexto de los sistemas de información es una organización y sus sistemas. Esto conduce a diferencias con las ciencias de la computación en el contexto del trabajo a realizar, los tipos de problemas a resolver, los tipos de sistemas a diseñar y gestionar, y la manera en la que se emplea la tecnología. Los modelos curriculares en SI se centran en la formación de titulados para el entorno de las organizaciones. Por el contrario, los modelos curriculares en ciencias de la computación invierten la situación y se centran en la enseñanza de la tecnología y en los procesos algorítmicos asociados, dejando al margen, por lo general, las necesidades reales de la empresa, pues funciones organizativas o de organización de sistemas no suele ser un área de mucho interés.

Sin embargo, existen conocimientos y materias que son compartidas tanto en los Sistemas de Información como en las Ciencias de la Computación. Éste es el caso particular de los Algoritmos y estructuras de datos, materia que forma parte del bloque de conocimientos en los Sistemas de Información, estudiada básicamente en el área temática Tecnología de la Información del modelo curricular IS'97, el cual se describe a continuación.

El modelo curricular se organiza en 5 áreas temáticas. Cada una de estas áreas tiene asociado uno o más cursos, formando un total de 10 cursos, los cuales se organizan en tres niveles diferentes

El conjunto de los 10 cursos está formado por 127 unidades de aprendizaje, que describen el material que los estudiantes deben aprender. Las unidades de aprendizaje se forman de un conjunto de más de 1000 temas o elementos del cuerpo de conocimiento de los SI. Cada uno de estos temas se incluye en una o varias unidades de aprendizaje, siendo estudiado a un grado de profundidad (nivel de competencia) distinto en cada unidad, dependiendo del curso al que se asigne.

En lo que respecta a Programación y Estructuras de Datos, destacamos en concreto la descripción que se realiza de los temas o elementos incluidos dentro del epígrafe 1.0 Tecnología de la Información, en el punto 1.2 Algoritmos y estructuras de datos. Esta descripción se muestra a continuación:

- 1.2 Algoritmos y estructuras de datos.
 - 1.2.1 Estrategias para la resolución de problemas.
 - 1.2.1.1 Técnicas de resolución de problemas utilizando algoritmos voraces.
 - 1.2.1.2 Técnicas de resolución de problemas utilizando algoritmos divide—vencerás.
 - 1.2.1.3 Técnicas de resolución de problemas utilizando algoritmos de backtracking.
 - 1.2.1.4 Proceso de diseño de software; de la especificación a la implementación.
 - 1.2.1.5 Determinación algorítmica; abstracción procedural; parámetros.
 - 1.2.1.6 Estrategias de implementación.
 - 1.2.1.7 Conceptos de verificación formal.
 - 1.2.1.8 Modelos formales de computación.
 - 1.2.2 Estructuras de datos básicas: listas, vectores, cadenas, registros, con juntos, listas enlazadas, pilas, colas, árboles, grafos.
 - 1.2.3 Estructuras de datos complejas: texto, voz, imágenes, vídeos y objetos multimedia.
 - 1.2.4 Tipos abstractos de datos.
 - 1.2.4.1 Propósito e implementación de tipos abstractos de datos.
 - 1.2.4.2 Especificaciones informales.
 - 1.2.4.3 Especificaciones formales, precondiciones y postcondiciones, especificaciones algebraicas para los tipos abstractos de datos.
 - 1.2.4.4 Módulos, cohesión, acoplamiento; diagramas de flujo de datos y conversión a diagramas jerárquicos.
 - 1.2.4.5 Corrección, verificación y validación.
 - 1.2.4.6 Estructuras de control; selección, iteración, recursión; tipos de datos y sus usos en la resolución de problemas.
 - 1.2.5 Estructuras de ficheros: secuencial, acceso directo, hashing, indexado.
 - 1.2.5.1 Ficheros: conceptos fundamentales; ficheros secuenciales y no secuenciales.
 - 1.2.5.2 Ficheros: contenidos de un directorio y estructura, renombrar, buscar, acceso y copias de seguridad.
 - 1.2.5.3 Ficheros: métodos de seguridad, protección y acceso.
 - 1.2.6 Algoritmos y estructuras de datos para ordenación y búsqueda.
 - 1.2.6.1 Algoritmos de ordenación.
 - 1.2.6.2 Algoritmos de búsqueda.
 - 1.2.6.3 Búsqueda, hashing, resolución de colisiones.

- 1.2.7 Eficiencia algorítmica, complejidad y métricas.
 - 1.2.7.1 Análisis asintótico: cotas superior, inferior y media.
 - 1.2.7.2 Consideraciones de tiempo y espacio en los algoritmos.
 - 1.2.7.3 Clases de complejidad: P y NP.
 - 1.2.7.4 Análisis de la cota inferior (para ordenación).
 - 1.2.7.5 NP-completitud.
 - 1.2.7.6 Algoritmos de ordenación cuadráticos.
 - 1.2.7.7 Algoritmos de ordenación logarítmicos.
 - 1.2.7.8 Backtracking, análisis sintáctico, simulaciones discretas, etc.
 - 1.2.7.9 Fundamentos de análisis de algoritmos.
- 1.2.8 Algoritmos recursivos.
 - 1.2.8.1 Inducción matemática y recursión.
 - 1.2.8.2 Comparación de algoritmos iterativos y recursivos.
- 1.2.9 Redes neurales y algoritmos genéticos.
 - 1.2.10 Consideraciones avanzadas.
 - 1.2.10.1 Funciones calculables.
 - 1.2.10.2 Máquinas universales (Turing, etc.).
 - 1.2.10.3 Problemas de decisión.
 - 1.2.10.4 Modelos de arquitecturas paralelas.
 - 1.2.10.5 Algoritmos para arquitecturas paralelas.
 - 1.2.10.6 Problemas matemáticos: problemas condicionados.
 - 1.2.10.7 Problemas matemáticos: método de Newton; eliminación gaussiana.
 - 1.2.10.8 Problemas matemáticos: clasificación de los errores.
 - 1.2.10.9 Problemas matemáticos: aplicación de métodos iterativos en ciencias e ingeniería.
 - 1.2.10.10 Calculabilidad y decibilidad.

1.1.3 Discusión comparativa de los currículos internacionales

En este apartado se realizará una comparativa entre los programas propuestos de los currículos internacionales presentados anteriormente y el programa propuesto para PED.

La asignatura Programación y Estructuras de Datos (PED), como asignatura básica en la enseñanza de la materia de programación, ha sido contemplada en cada uno de los currículos internacionales tratados.

En la propuesta de ACM/IEEE'91, los cursos correspondientes a la materias o requisitos comunes AL: Algoritmos y estructuras de datos y PL: Lenguajes de programación están directamente relacionados con la asignatura PED. Concretamente, las unidades de conocimiento AL1: Estructuras de datos básicas, AL2: Tipos abstractos de datos, AL3: Algoritmos recursivos y PL3: Representación de tipos de datos coinciden prácticamente con la propuesta de la asignatura aquí

presentada. Hay que destacar que el Bloque referente al tipo árbol no ha sido abordado con tanto detalle como en nuestra propuesta y que el Bloque tipo conjunto no se trata en ninguna unidad.

Las prácticas propuestas en las distintas unidades están también relacionadas con las prácticas propuestas para PED.

En la propuesta de ACM/IEEE'2001 crece el número de cursos relacionados con la asignatura PED. Estos cursos son los siguientes: PF: Fundamentos de programación, AL: Algoritmos y complejidad, PL: Lenguajes de programación. Concretamente las unidades de conocimiento relacionadas con PED son las siguientes PF3: Estructuras de datos fundamentales, PF4: Recursión, AL3: Algoritmos fundamentales para computación, PL6: Programación orientada a objetos, PL9: Sistemas tipo y PL10: Semántica de los lenguajes de programación. Esta nueva propuesta de ACM/IEEE cubre en su totalidad la propuesta aquí presentada. Debemos destacar que se aborda con mucho más detalle las tablas hash y los algoritmos y aplicaciones de grafos. Por último, mencionar que existen 3 unidades dedicadas a la programación orientada a objetos, los tipos abstractos de datos y la semántica para la declaración de los tipos de datos.

Respecto a la propuesta del modelo curricular ACM/AIS/AITP en SI (informe IS'97), el epígrafe 1.2 Algoritmos y estructuras de datos cubre totalmente la propuesta de la asignatura PED presentada. Debemos resaltar el papel que juega la programación orientada a objetos como mecanismo de implementación de TADs. Este modelo curricular, junto con el ACM/IEEE'2001, son los únicos que aportan esta visión.

Como resumen podemos concluir que las propuestas internacionales mencionadas abordan esta asignatura de un modo similar a la propuesta aquí presentada. Los modelos curriculares ACM/IEEE'2001 y ACM/AIS/ AITP (informe IS'97) coinciden con nuestra propuesta en la utilización de la programación orientada a objetos como mecanismo para implementar los TADs.

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

A continuación se mostrará un estudio de las diferentes afinidades existentes entre la asignatura de Programación y Estructuras de Datos (PED) con otras asignaturas del plan de estudios de Ingeniería en Informática, tal y como se ha comentado anteriormente.

En la Figura se muestra un resumen de las diferentes relaciones entre cada una de las asignaturas del plan de estudios y la asignatura Programación y Estructuras de Datos.

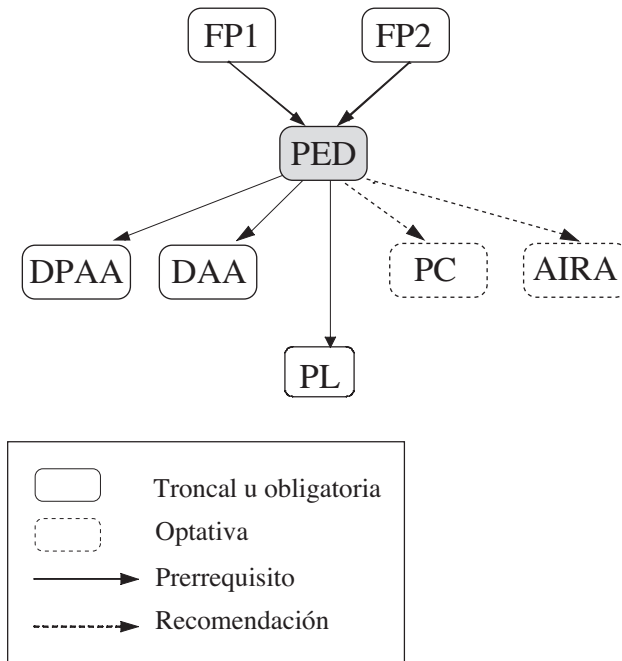


Ilustración 11: relaciones de Programación y Estructuras de Datos en el plan de estudios

Fundamentos de Programación I (FPI) y Fundamentos de Programación II (FPII)

- Carácter: Troncal.
- Curso: 1.
- Relación: Estas asignaturas son prerrequisitos de PED.
- Objetivos:
 1. Facilitar al alumno una visión inicial del campo de la Programación
 2. Que el alumno sea capaz de Especificar, Diseñar, Analizar y Elaborar (implementar) un Algoritmo a partir de un problema que se le plantee, habiendo estudiado previamente el concepto de algoritmo y sus distintas representaciones.
 3. Introducir al alumno en el proceso de Diseño y Elaboración de un Programa, independientemente del lenguaje de programación que se vaya a utilizar.
 4. Que el alumno aprenda a trabajar con los Elementos Básicos de un lenguaje de programación de 3ra. generación: variables, funciones, tipos de datos estructurados, punteros, estructuras de datos dinámicas y ficheros.

5. Que el estudiante sea capaz de escribir, en un plazo razonable, programas correctos, eficientes, bien organizados, bien documentados y legibles.
 6. Introducir al alumno en los distintos paradigmas de programación, como alternativa y complemento a los lenguajes de programación procedimentales clásicos.
 7. Que el alumno conozca los tipos generales de problemas susceptibles de resolución por computador y las herramientas necesarias para ello.
 8. Que el estudiante sea consciente de las implicaciones del trabajo de programador, individualmente y como miembro de un equipo.
 9. Conocer el lenguaje de programación C y las mejoras que aporta al mismo el lenguaje C++, diferenciando claramente aquellos aspectos dependientes de la sintaxis del lenguaje de los que no lo son.
- Contenidos:
 1. Introducción: computadores y programas.
 2. Tipos de datos elementales.
 3. Lenguaje algorítmico. Concepto de programa.
 4. Programación modular.
 5. Tipos de datos estructurados.
 6. Lenguajes de programación.
 7. Introducción al Lenguaje C/C++. Funciones Básicas de Entrada-Salida en C++. Estructuras de Control. Programación Modular. Estructuras de datos: Vectores y Matrices, Cadenas de Caracteres, Registros y Uniones. Tipos Enumerados. Ficheros. Estructuras Dinámicas.

Diseño y Análisis de Algoritmos (DAA).

- Carácter: Troncal.
- Curso: 3.
- Relación: PED es prerequisite de esta asignatura.
- Objetivos:
 1. Proporcionar al alumno la capacidad de analizar con rigor la eficiencia de los algoritmos.
 2. Proporcionar al estudiante un conocimiento detallado de los algoritmos de ordenación de tablas.
 3. Proporcionar al alumno la capacidad para diseñar algoritmos recursivos.
 4. Proporcionar al estudiante técnicas de diseño basadas en esquemas generales que se pueden instanciar para resolver problemas concretos.
- Contenidos:
 1. La eficiencia de los algoritmos. Introducción. Noción de complejidad. Obtención de las cotas superior e inferior de los algoritmos. Ejemplos de aplicación. Problemas y cuestiones. # Diseño recursivo. Introducción al

diseño recursivo. Fundamentos del diseño recursivo. Etapas del diseño recursivo. Recursión versus Iteración.

2. Programación con Esquemas. Divide y vencerás, Estrategia voraz, Backtracking ó vuelta atrás: ámbito de aplicación, método general, presentación del esquema y ejemplos de aplicación.

Diseño y Programación Avanzada de Aplicaciones (DPAA)

- Carácter: Obligatoria.
- Curso: 3.
- Relación: PED es prerequisite de esta asignatura.

Procesadores del Lenguaje (PL)

- Carácter: Troncal.
- Curso: 4.
- Relación: PED es recomendación de esta asignatura.
- Objetivos:
 1. Saber escribir una especificación léxica y sintáctica para un lenguaje determinado.
 2. Comprender el funcionamiento de un analizador léxico.
 3. Comprender el funcionamiento de los analizadores sintácticos descendentes recursivos, LL(1), y por desplazamiento-reducción.
 4. Saber transformar una gramática para construir un analizador sintáctico descendente recursivo, LL(1), o SLR.
 5. Saber programar completamente cada tipo de analizador sintáctico.
 6. Comprender los principios básicos de la utilización de la sintaxis de los programas fuente en el proceso de traducción a programas objeto.
 7. Saber especificar una traducción (y tareas parcialmente relacionadas con ella, como un proceso de comprobación de tipos) utilizando las herramientas formales de la teoría de compiladores.
 8. Saber implementar una traducción especificada formalmente sobre cada tipo básico de analizador sintáctico.
 9. Comprender la necesidad del tratamiento de los diferentes tipos de datos y su efecto sobre el diseño de los compiladores.
 10. Conocer los mecanismos básicos del tratamiento y de la comprobación de tipos simples y estructurados.
 11. Comprender la conveniencia para el diseño de un conjunto de compiladores para varios lenguajes fuente y varias máquinas de hacer una traducción a una representación intermedia antes de generar código final.
 12. Saber escribir un traductor utilizando yacc y lex.
 13. Conocer las técnicas básicas para la compilación de arrays, registros, funciones y procedimientos en lenguajes de alto nivel.

14. Conocer las técnicas básicas de optimización de código intermedio y de generación de código objeto optimizado.
- Contenidos:
 1. Análisis léxico.
 2. Análisis sintáctico.
 3. Análisis sintáctico descendente.
 4. Análisis sintáctico ascendente.
 5. Análisis semántico.
 6. Traducción dirigida por la sintaxis: implementación de ETDS con analizadores LR.
 7. Tratamiento de tipos. Compilación de construcciones complejas.
 8. Compilación de funciones/procedimientos en lenguajes de alto nivel.
 9. Optimización.

Aplicaciones Industriales del Reconocimiento Automático (AIRA)

- Carácter: Optativa.
- Curso:
- Relación: PED es recomendación de esta asignatura.
- Objetivos:
 1. Que el alumno conozca algunas técnicas de preproceso en problemas típicos en la industria.
 2. Que el alumno conozca algunas técnicas de selección de características (qué características escoger de las obtenidas en el preproceso).
 3. Que el alumno conozca algunas técnicas de extracción de características (cómo combinar las características para reducir su número sin perder información).
 4. Que el alumno conozca algunas técnicas para la construcción de clasificadores (técnicas paramétricas y basadas en distancias).
- Contenidos:
 1. Métodos geométricos. Teoría de la decisión de Bayes. Técnicas paramétricas. Técnicas no paramétricas. Funciones discriminantes lineales.
 2. Métodos no supervisados.
 3. Métodos sintácticos. Gramáticas de regulares. Gramáticas de incontextuales. Gramáticas de dimensión superior. Transductores.

Programación Concurrente (PC)

- Carácter: Optativa.
- Curso:
- Relación: PED es recomendación de esta asignatura.
- Objetivos:
 1. Introducir al alumno en el paradigma de la programación concurrente y

en el uso de las herramientas y técnicas utilizadas, en general, por los lenguajes concurrentes.

- Contenidos:
 1. Procesos.
 2. Semáforos.
 3. Regiones críticas.
 4. Monitores.
 5. Paso de mensajes.
 6. Modelo Unificado.
 7. Lenguajes de programación.

Programación Orientada a Objetos (POO)

- Carácter: Obligatoria.
- Curso: 2.
- Relación: Contenidos útiles para PED.
- Objetivos:
 1. Presentar al alumno la metodología de Programación Orientada a Objetos (POO).
 2. Tratar el tema de identificación de Clases y Objetos.
 3. Dar a conocer de manera sencilla una metodología de Diseño Orientada a Objetos estándar.
 4. Estudiar diversos Lenguajes de Programación relacionados con la programación orientada a objetos.
 5. Por último, se tratarán conceptos más recientemente relacionados con la programación orientada a objetos, tales como Persistencia, Objetos Difusos, etc.
- Contenidos:
 1. Introducción a la programación orientada a objetos.
 2. Diseño orientado a objetos.
 3. Programación orientada a objetos y lenguajes de programación.
 4. Temas actuales en la programación orientada a objetos: persistencia, objetos difusos, objetos distribuidos, herencia y genericidad.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos:

- OI1: Conocer los mecanismos de abstracción y su importancia para la resolución de problemas.

- OI2: Conocer los tipos de datos más usuales en programación, sus realizaciones más comunes y su utilidad.
- OI3: Comprender la necesidad de separación entre los niveles de especificación, instrumentación y uso.
- OI4: Comprender la necesidad de adaptar la representación interna del tipo abstracto de datos a los requerimientos de la aplicación a resolver.
- OI5: Ser capaz de distinguir entre las instrumentaciones alternativas de una abstracción de datos, y razonar sobre la solución escogida en cuanto a coste se refiere.
- OI6: SAPlicar los tipos abstractos de datos básicos aprendidos a problemas prácticos reales.
- OI7: Ser capaz de organizar un determinado volumen de datos de la forma más racional posible en función de los requerimientos del problema a resolver.
- OI8: Crear nuevos tipos abstractos de datos, o que pueda elegir otra implementación de los mismos para adaptarlos a una aplicación determinada.
- OI9: Ser capaz de evaluar cada representación de un tipo abstracto de datos en función de su consumo de recursos (eficiencia).

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, estos son los objetivos cOIP1, cOIP2 y cOIP3 detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, estos son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber):

Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2 desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes:

Bloque 1: Introducción a los tipos abstractos de datos

- CIC1: Comprender la importancia de la abstracción, y en particular, de la abstracción de datos, en el proceso de descomposición modular.
- CIC2: Entender la separación de los niveles de especificación, implementación y uso de un tipo abstracto de datos.
- CIC3: Conocer las especificaciones algebraicas como herramienta para la especificación de los tipos abstractos de datos.
- CIC4: Conocer los lenguajes de programación que soporten la definición y uso de los tipos abstractos de datos.
- CIC5: Saber los conceptos básicos relacionados con el cálculo de la eficiencia de los algoritmos.

Bloque 2: Tipos lineales

- CIC6: Conocer los tipos de datos lineales clásicos como tipos abstractos de datos.
- CIC7: Conocer la importancia y utilidad de los tipos estudiados.

Bloque 3: Tipo árbol

- CIC8: Conocer el concepto de árbol y la terminología asociada en su uso.
- CIC9: Reconocer el tipo abstracto de datos árbol binario y sus representaciones.
- CIC10: Conocer los algoritmos de recorrido de árboles.
- CIC11: Conocer los árboles n-arios y árboles generales.
- CIC12: Conocer los árboles binarios de búsqueda.
- CIC13: Conocer los árboles n-arios de búsqueda como extensión del árbol binario de búsqueda.

Bloque 4: Tipo conjunto

- CIC14: Conocer el concepto de conjunto mediante su especificación como tipo abstracto de datos.
- CIC15: Conocer las principales representaciones del conjunto.
- CIC16: Entender los diferentes tipos de conjuntos: diccionario, cola de prioridad, trie y conjunto unión-búsqueda.

Bloque 5: Tipo grafo

- CIC17: Conocer el concepto de grafo y su especificación.
- CIC18: Conocer los recorridos de grafo en profundidad y en anchura.
- CIC19: Conocer algoritmos clásicos de manipulación de grafos dirigidos y no dirigidos.

Capacidades metodológicas (saber hacer):

Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3 introducidas a nivel general en el artículo introductorio de este libro:

Bloque 1: Introducción a los tipos abstractos de datos

- CIM1: Ser capaz de calcular la eficiencia de un algoritmo.
- CIM2: Ser capaz de distinguir entre distintas implementaciones de un tipo abstracto de datos en función del análisis de su eficiencia.
- CIM3: Ser capaz de analizar y comprender el funcionamiento de los algoritmos recursivos.

Bloque 2: Tipos lineales

- CIM4: Distinguir y comparar las diferentes implementaciones de los tipos de datos lineales analizando su eficiencia.
- CIM5: Identificar y evaluar la representación idónea de cada tipo abstracto de datos en función de los requerimientos del problema.
- CIM6: Ser capaz de unir y combinar los tipos abstractos de datos básicos para formar estructuras más complejas.

Bloque 3: Tipo árbol

- CIM7: Identificar las diferencias entre el tipo árbol y los tipos lineales.
- CIM8: Implementar el tipo árbol y de analizar la eficiencia de diferentes implementaciones.
- CIM9: Aplicar los diferentes recorridos de árboles a la resolución de problemas.
- CIM10: Aplicar el árbol binario para la ordenación de un conjunto de elementos.

Bloque 4: Tipo conjunto

- CIM11: Ser capaz de emplear diferentes representaciones (árbol binario de búsqueda, árboles-B, etc.) para implementar el tipo conjunto.
- CIM12: Analizar la eficiencia de los diferentes tipos de conjuntos.

Bloque 5: Tipo grafo

- CIM13: Analizar la eficiencia de las diferentes representaciones de grafos.
- CIM14: Implementar grafos dirigidos y no dirigidos.

Destrezas tecnológicas (saber hacer):

Además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con fluidez el lenguaje de programación C++ que se empleará para la realización de las prácticas relacionadas con la asignatura.
- CIT2: Saber emplear las herramientas de compilación y depuración más conocidas.

Destrezas lingüísticas (saber hacer):

Además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico-técnico aplicado al manejo de las estructuras de datos, siendo riguroso en las explicaciones de cualquier proceso ya sea algorítmico o teórico.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas:

Las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo:

Las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

- Comprender e interpretar el concepto de algoritmo.
- Conocer los principales paradigmas de programación.
- Conocer los elementos básicos de un lenguaje de programación de tercera generación: variables, estructuras de control, funciones, etc.
- Trabajar con los tipos de datos básicos de un lenguaje moderno de programación.

- Entender las técnicas de la programación procedural.
- Entender y saber aplicar la recursividad.
- Conocer las estructuras de datos lineales básicas (lista, pila y cola).
- Conocer los algoritmos básicos de ordenación y búsqueda.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCIÓN DE LOS PRERREQUISITOS

Los prerrequisitos necesarios para el estudio y entendimiento de esta asignatura se cubren en las asignaturas Fundamentos de Programación I y Fundamentos de Programación II que se imparten en primero de las tres titulaciones de informática.

4. BLOQUES Y TEMAS DE CONTENIDOS

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

Bloque 1: Introducción a los tipos abstractos de datos

- Tema 0: Presentación y objetivos de la asignatura.
- Tema 1: Introducción a los tipos abstractos de datos.

Bloque 2: Tipos lineales

- Tema 2: Vectores.
- Tema 3: Pilas.
- Tema 4: Colas.
- Tema 5: Listas.

Bloque 3: Tipo árbol

- Tema 6: Definiciones generales de árboles.
- Tema 7: Árboles binarios.
- Tema 8: Árboles n-arios y árboles generales.
- Tema 9: Árboles binarios de búsqueda.
- Tema 10: Árboles n-arios de búsqueda.

Bloque 4: Tipo conjunto

- Tema 11: El conjunto.
- Tema 12: El diccionario.
- Tema 13: Colas de prioridad.
- Tema 14: Otros tipos de conjuntos.

Bloque 5: Tipo grafo

- Tema 15: Grafos.
- Tema 16: Aplicaciones con grafos.

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1: Introducción a los tipos abstractos de datos

- Tema 0: Presentación y objetivos de la asignatura.
- Tema 1: Introducción a los tipos abstractos de datos.
 - 1.1 Introducción.
 - 1.2 Especificación algebraica de los tipos abstractos de datos.
 - 1.3 Implementación de tipos abstractos de datos.
 - 1.4 Diseño recursivo.

Bloque 2: Tipos lineales

- Tema 2: Vectores.
 - 2.1 Definición y especificación.
 - 2.2 Representación secuencial.
 - 2.3 Aplicaciones de vectores: matrices y polinomios.
- Tema 3: Pilas.
 - 3.1 Definición y especificación.
 - 3.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
 - 3.3 Aplicaciones de pilas: evaluación de expresiones y gestión de la llamada a procedimientos recursivos.
- Tema 4: Colas.
 - 4.1 Definición y especificación.
 - 4.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
 - 4.3 Aplicaciones de colas: simulación y gestión de colas de espera.
- Tema 5: Listas.
 - 5.1 Listas de acceso por posición: definición y especificación.
 - 5.2 Representación secuencial.
 - 5.3 Representación enlazada: listas enlazadas circulares, listas doblemente enlazadas y listas multienlazadas.
 - 5.4 Análisis de la eficiencia de las representaciones.
 - 5.5 Aplicaciones de listas: matrices dispersas y representación de polinomios.

Bloque 3: Tipo árbol

- Tema 6: Definiciones generales de árboles.
- Tema 7: Árboles binarios.
 - 7.1 Definición y especificación.
 - 7.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
 - 7.3 Recorridos de un árbol binario: preorden, inorden, postorden y niveles.

- 7.4 Operaciones complementarias: profundidad, número de elementos y altura de un nodo.
- 7.5 Árboles binarios enhebrados.
- 7.6 Aplicaciones de árboles binarios: árboles para la evaluación de expresiones aritméticas y códigos de Huffman.
- Tema 8: Árboles n-arios y árboles generales.
 - 8.1 Definición y especificación.
 - 8.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
 - 8.3 Recorridos.
 - 8.4 Aplicaciones: árboles genealógicos.
- Tema 9: Árboles binarios de búsqueda.
 - 9.1 Definición y especificación.
 - 9.2 Operaciones básicas: inserción, búsqueda y eliminación. Análisis de la eficiencia de las operaciones.
 - 9.3 Árbol binario de búsqueda equilibrado en altura (AVL). Algoritmos de inserción y eliminación. Análisis de su eficiencia.
 - 9.4 Aplicaciones.
- Tema 10: Árboles n-arios de búsqueda.
 - 10.1 Definición y especificación.
 - 10.2 Árboles n-arios de búsqueda: árboles 2-3, árboles 2-3-4, árboles rojos-negros y árboles-B.
 - 10.3 Aplicación: utilización en la organización interna de bases de datos.

Bloque 4: Tipo conjunto

- Tema 11: El conjunto.
 - 11.1 Definición y especificación.
 - 11.2 Representación: secuencial y enlazada basada en listas. Análisis de la eficiencia de las representaciones.
 - 11.3 Clasificación de los conjuntos según las operaciones.
- Tema 12: El diccionario.
 - 12.1 Definición y especificación.
 - 12.2 Representación mediante tablas de dispersión (hashing). Dispersión cerrada y abierta; análisis comparativo entre ambas. Operaciones del diccionario con métodos de dispersión.
 - 12.3 Otras representaciones: árbol 2-3, árbol 2-3-4, árbol rojo-negro y árbol-B.
- Tema 13: Colas de prioridad.
 - 13.1 Definición y especificación.
 - 13.2 Definición de árbol parcialmente ordenado. Representaciones: listas ordenadas y árboles parcialmente ordenados.

- 13.3 El montículo (heap) para representar árboles parcialmente ordenados. Tipos de montículos.
- 13.4 Algoritmos de inserción y borrado del elemento de prioridad mínima (máxima). Análisis de su eficiencia.
- Tema 14: Otros tipos de conjuntos.
 - 14.1 El trie. Definición e implementación. Algoritmos de búsqueda, inserción y borrado. Análisis de su eficiencia.
 - 14.2 Conjunto unión-búsqueda. Definición e implementación. Análisis de la eficiencia de las representaciones.

Bloque 5: Tipo grafo

- Tema 15: Grafos.
 - 15.1 Definición de grafo y terminología. Especificación.
 - 15.2 Algunas representaciones: matriz de adyacencia y lista de adyacencia. Análisis de la eficiencia de las representaciones.
 - 15.3 Recorridos en un grafo: profundidad y anchura. Comparación de los métodos.
- Tema 16: Aplicaciones con grafos.
 - 16.1 Grafos dirigidos. Cálculo de caminos mínimos: algoritmos de Dijkstra y Floyd. Grafos acíclicos dirigidos.
 - 16.2 Grafos no dirigidos. Árbol de extensión de coste mínimo: algoritmos de Prim y Kruskal. Comparación de los dos métodos. Coloreado de grafos y ordenación topológica.

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

Las propuestas metodológicas que se plantean se basarán en el establecimiento de los objetivos de la asignatura, así como una serie de actividades didácticas dirigidas a la adquisición de conocimientos mediante las clases teóricas y de problemas, las clases de prácticas en el laboratorio, el proceso tutorial y una serie de actividades adicionales.

5.1 EL ESTABLECIMIENTO DE OBJETIVOS

En una adecuada metodología didáctica intervienen fundamentalmente tres factores: los objetivos propuestos, los medios disponibles y las posibilidades de los sujetos que participan en el acto educativo.

El primer factor vendrá definido por los planes de estudio, y matizado por la labor docente del profesor. Así, la tarea del docente será la de establecer los medios en función de los objetivos y los objetivos en función de los medios, y ambos conceptos con las posibilidades reales que la situación genera.

De este conjunto de relaciones se extraerá la propuesta de objetivos que el docente planifica para la asignatura. Un objetivo debe indicar, sin posibles

ambigüedades, cuáles son las tareas que debe realizar el alumno y con qué contenidos.

A medida que estas tareas se concretan, los objetivos se clarifican pasando de objetivos generales a objetivos específicos. Es muy importante que en este proceso de concreción de los objetivos se conserve la coherencia, de forma que no se pierdan los objetivos generales de la asignatura al concentrarse en los detalles de la misma. Una presentación de los objetivos bien realizada puede ayudar a conseguir una motivación en el alumno que le facilite superar muchos obstáculos posteriores.

5.2 LA ADQUISICIÓN DE LOS CONOCIMIENTOS

De acuerdo con los objetivos generales y con la estructuración de los contenidos, se indicarán los métodos, procedimientos e instrumentos que, de forma genérica, se van a utilizar:

Las clases de teoría. En estas clases se presentarán los contenidos básicos de un cierto tema. Comienzan por una introducción donde se comentarán los contenidos del tema y sus objetivos específicos, así como un breve comentario sobre los conceptos ya vistos en clases anteriores y que tengan incidencia en los puntos a tratar.

A continuación se realizará la exposición de los contenidos, haciendo uso de aquellos medios audiovisuales, textos y documentos que permitan un adecuado nivel de motivación e interés en los alumnos.

En la metodología propuesta se compaginarán clases expositivas-frontales (de tipo magistral) con clases expositivas-interactivas (solución de problemas, estudio de casos prácticos, trabajos por grupos, etc.), intentando hacer uso de estas segundas siempre que sea posible. Las características de las asignaturas propuestas facilitan dicha aproximación ya que se pueden introducir los conceptos básicos para después pasar a proponer y solucionar problemas.

En esa línea, se debe motivar a los alumnos a intervenir en cualquier momento en las clases de teoría, exponiendo sus dudas, realizando comentarios que se intentarán aprovechar para realizar de forma más dinámica la clase y facilitar el aprendizaje deseado por parte de los alumnos.

Por último, es interesante dedicar siempre un periodo de la clase para mostrar las conclusiones más importantes a modo de resumen del tema.

Las clases de prácticas. Las prácticas son también un medio excelente para que el alumno potencie su iniciativa y capacidad crítica. Además, son imprescindibles para conseguir un buen aprendizaje en materias científicas y tecnológicas, permitiendo proyectar los conocimientos acumulados sobre problemas reales. Por todos estos motivos es necesario prestar una gran atención y dedicación a las prácticas, para conseguir hacerlas útiles y sugestivas. Así, la planificación y supervisión deben ser completas y directas.

Por último, las clases prácticas permiten al profesor estar en mayor contacto con los alumnos debido al número reducido de alumnos por cada grupo, permitiéndole realizar un mejor seguimiento de cada estudiante.

El enfoque utilizado en las prácticas es el siguiente:

- Se plantean como una serie de problemas de complejidad creciente.
- Se realizará un seguimiento continuado a las tareas que realiza el alumno. Este seguimiento permite observar los progresos que realiza el alumno a lo largo del curso. Además, se pretende evitar que el alumno inicie sus prácticas con planteamientos incorrectos. Por último, se establece la comunicación con el alumno que permite que se le puedan indicar los errores que está cometiendo y el modo de solucionarlos.
- Se deberán realizar en parejas. Con ello se pretende fomentar el trabajo en equipo y favorecer el intercambio de conocimientos entre los integrantes de la pareja (desarrollo de habilidades de comunicación tanto oral como escrita). Sin embargo, se especificará que la evaluación de sus conocimientos se realizará de forma individual.
- Los alumnos dispondrán con suficiente antelación de los enunciados de las prácticas. Los enunciados deberán contener: objetivos generales, materias de trabajo, contenido mínimo de las memorias que se exigirán en cada práctica y fecha de entrega.
- Deberán estar sincronizadas con las clases de teoría. Así se consigue que el alumno conozca los aspectos teóricos, previamente al planteamiento del trabajo práctico.

El proceso tutorial. Las horas de asistencia al alumnado o tutorías son esenciales para la calidad de la enseñanza. Estas horas son la continuación de las clases de exposición, de problemas y prácticas de laboratorio.

La atención personalizada en las horas de asistencia al alumnado mejora el aprendizaje y el conocimiento entre profesor y alumno. Los alumnos podrán acudir al profesor dentro de un horario estipulado.

Desgraciadamente, un elevado número de alumnos no acude a tutorías o hace un escaso uso de ellas. Probablemente, ello se debe a que los alumnos no estudian de forma continuada y sólo acuden al profesor en las vísperas del examen o de la entrega de trabajos. Por ello, es muy importante recordarles que disponen de ellas durante todo el curso y animarlos a que acudan.

Los objetivos de las tutorías son:

- Motivar al alumno en la asignatura.
- Informar al alumno sobre contenidos teórico-prácticos de la misma.
- Ayudar al alumno en la resolución de los ejercicios propuestos en teoría y prácticas.

Las actividades adicionales. Es conveniente que el alumno pueda realizar algunas actividades adicionales que le estimulen e incrementen su interés por la

asignatura. Estas actividades deben ser optativas, ya que su principal objetivo es permitir al alumno profundizar en algunos temas y no marcar un requisito adicional para superar la asignatura. Las actividades pueden consistir en asistencia a seminarios y conferencias de interés sobre temas puntuales o bien realizar trabajos de curso que potencien el trabajo en equipo.

Los recursos de Internet. Las asignaturas ofrecen páginas web a través de las cuales los alumnos pueden adquirir el material necesario para la docencia tanto teórica como práctica, consultar los objetivos, temario y planteamiento de las asignaturas, así como estar al tanto de sus noticias y aspectos dinámicos.

Además, el correo electrónico proporciona una excepcional vía de comunicación entre profesores y alumnos al permitir resolver ciertas dudas puntuales sin necesidad de producirse un contacto presencial.

Por último, destacar que la Universidad de Alicante proporciona un Campus Virtual a través del cual se proporciona un servicio de complemento a la docencia y a la gestión académica dirigido tanto al profesorado como al alumnado y al personal de administración de la Universidad de Alicante.

Entre otras funcionalidades se encuentran:

- Publicación de la bibliografía.
- Publicación enlaces de interés.
- Tutorías electrónicas.
- Publicación de un repertorio de preguntas frecuentes.
- Propuesta y moderación de debates.
- Publicación de controles y sus notas.
- Suministro de materiales docentes.
- Listas de clase.
- Fichas de alumnos.
- Información sobre planes de estudio.
- Gestión de anuncios de interés.
- Publicación de horarios de tutorías presenciales.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se muestra el plan de trabajo de la asignatura. Se distingue entre horas presenciales (realización de actividades en las aulas) y horas no presenciales (trabajo y esfuerzo personal realizado en la asignatura por los estudiantes).

NÚMERO DE HORAS PRESENCIALES ¹				
ACTIVIDAD		Clases de teoría	Clases de práctica	Actividades en grupos pequeños/ tutorías docentes
BLOQUE 1	Tema 0	1	1	0
	Tema 1	2	1	1
BLOQUE 2	Tema 2	1	3	1
	Tema 3	1	3	1
	Tema 4	1	3	0
BLOQUE 3	Tema 5	2	4	1
	Tema 6	1	1	0
	Tema 7	2	2	2
	Tema 8	1	2	0
BLOQUE 4	Tema 9	3	2	2
	Tema 10	3	2	4
	Tema 11	1	1	0
	Tema 12	2	1	1
BLOQUE 5	Tema 13	2	1	2
	Tema 14	2	1	0
	Tema 15	2	1	1
	Tema 16	3	1	1
TOTAL: 77		30	30	17

Cuadro 1. Plan de trabajo de la asignatura, número de horas presenciales

NÚMERO DE HORAS NO PRESENCIALES		
ACTIVIDAD	Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura
Tema 0	0	0
Tema 1	2	1
Tema 2	2	1
Tema 3	1	2
Tema 4	1	2
Tema 5	2	2
Tema 6	1	2
Tema 7	2	2
Tema 8	1	2
Tema 9	2	2
Tema 10	4	2
Tema 11	2	1
Tema 12	1	1
Tema 13	2	1
Tema 14	2	1
Tema 15	2	1
Tema 16	1	1
TOTAL: 52	28	24

Cuadro 2. Plan de trabajo de la asignatura, número de horas no presenciales

¹ Las sesiones tanto de teoría como de prácticas que aparecen en la Tabla corresponden a 1,5 horas de docencia.

7. BIBLIOGRAFÍA

A continuación se presenta la bibliografía básica comentada, que se recomienda al alumno en la asignatura. Posteriormente, se ofrece una relación de otros textos como bibliografía complementaria que, en opinión del autor, pueden ser interesantes para la preparación de la asignatura por parte del profesor o como textos de consulta del alumno. En ambos casos, la bibliografía se presenta ordenada en función del apellido del autor.

7.1 BIBLIOGRAFÍA BÁSICA

Aho et al. (1988). *Estructuras de datos y algoritmos.* Addison-Wesley Iberoamericana.

Este libro de casi 440 páginas examina las estructuras de datos y los algoritmos que conforman la base de la programación de ordenadores. En los capítulos del 4 al 7, se examinan las estructuras de datos y los algoritmos que fundamentan gran parte de la programación actual. En la descripción y realización de los algoritmos, se usan de manera informal los tipos abstractos de datos, ya que se consideran un instrumento útil para el diseño de programas con independencia del lenguaje utilizado. También, en el artículo introductorio de este libro se introduce el concepto de complejidad temporal, debido a la importancia que los autores otorgan al análisis de la complejidad de los algoritmos. Sin embargo, se echa en falta un capítulo dedicado a los árboles 2-3-4 y los rojos-negros.

Carrano (1998). *Data abstraction and problem solving with C++.* Walls and mirrors. Addison-Wesley.

Entre los libros que utilizan C++ como lenguaje de instrumentación destacamos éste de más de 700 páginas. Se ajusta globalmente al enfoque docente propuesto en la programación de la asignatura. Como la mayoría de los libros clásicos relacionados con los tipos de datos, utiliza el lenguaje natural para realizar la especificación de los tipos de datos. Sin embargo, a diferencia de éstos, en el capítulo 12, se describen varios árboles de búsqueda avanzados: árboles 2-3, árboles 2-3-4 y árboles rojos-negros. El libro contiene además un gráfico que indica las dependencias entre los distintos capítulos. Por último, el libro incorpora un apéndice donde aparece una revisión de los conceptos fundamentales del lenguaje de programación. También posee un apéndice dedicado a la inducción matemática y otro que contiene la solución de los ejercicios del libro.

Ceballos (1997). *Programación orientada a objetos con C++.* RA-MA.

En este libro de casi 700 páginas, el alumno puede encontrar una gran ayuda para fijar los conocimientos sobre el lenguaje C++, puesto que apoyado en gran cantidad de ejemplos, repasa el camino que debe recorrer el programador para

llegar a aprovechar las innovaciones de la programación orientada al objeto. También se puede emplear como obra de referencia.

Deitel & Deitel (1995). *Cómo programar en C/C++*. Prentice Hall Hispanoamericana.

Este libro de casi 1000 páginas presenta la ventaja de partir de la programación en C para introducir C++. El libro está muy bien estructurado con resúmenes al final de cada capítulo, así como errores habituales en el aprendizaje del lenguaje. Además, como ventaja, este libro es recomendado como bibliografía básica en la asignatura de Fundamentos de Programación II, por lo que ya le resultará conocido al alumno.

Franch (1994). *Estructuras de datos. Especificación, diseño e implementación*. Edicions de la Universitat Politecnica de Catalunya.

Este libro de 420 páginas se ajusta globalmente al enfoque docente propuesto en la programación de la asignatura, excepto en el estudio de algunos tipos de datos: árboles 2-3, árboles 2-3-4, árboles rojos-negros, árboles-B, diccionario, trie y conjunto unión-búsqueda. Conviene destacar el artículo introductorio de este libro dedicado a la especificación algebraica de tipos de datos. También resaltar que utiliza pseudocódigo para la instrumentación, concretamente notación Merlí.

Horowitz & Sahni (1990). *Fundamentals of Data Structures in Pascal*. Computer Science Press.

Es un libro clásico en el estudio de las estructuras de datos. En sus 720 páginas se recogen contenidos de estructuras de datos y de algoritmos. Utiliza una especificación informal para la introducción de los tipos abstractos de datos. En general, se ajusta perfectamente al temario aquí propuesto. Hay que destacar el capítulo 10 en el que se detallan estructuras de datos avanzadas de árboles que suelen faltar en muchos de los libros habituales de tipos de datos, como por ejemplo los árboles 2-3, 2-3-4, rojos-negros, árboles-B, tries, árboles de búsqueda digitales, heap, leftist, etc.

Joyanes et al. (1999). *Estructura de Datos. Libro de Problemas*. McGraw-Hill.

Este libro de 550 páginas está dirigido a estudiantes que quieran profundizar en la disciplina de estructuras de datos. Los capítulos 4 al 12 del libro se ajustan globalmente al enfoque docente propuesto en la programación de la asignatura, excepto en el estudio de algunos tipos de árboles (árboles 2-3, árboles 2-3-4 y árboles rojos-negros) y de los tipos de datos del Bloque 4 (tipo conjunto). El lenguaje utilizado para la implementación de los tipos es Pascal. Utiliza tanto espe-

cificación formal (algebraica) como informal (en lenguaje natural). Uno de los principales intereses de este libro en comparación con otra bibliografía sobre tipos de datos es que aparecen una gran cantidad de ejercicios resueltos y propuestos al final de cada capítulo que son muy útiles para la correcta comprensión y aplicación de los conceptos explicados.

Kernighan & Richie (1985). *El lenguaje de programación C*. Addison-Wesley.

Con casi 300 páginas, es el manual original del lenguaje C, y como tal debe seguir siendo consultado, sobre todo por el principiante, aunque hay que admitir que el lenguaje C utilizado hoy en día es bastante diferente al descrito aquí. Todos los compiladores actuales soportan el estándar ANSI C, el cual es una evolución sobre el lenguaje descrito en este libro. Por ello, se recomienda al alumno que lea el libro de Gotfried (1991), que presenta y utiliza el ANSI C.

Peña (1998). *Diseño de programas. Formalismo y abstracción*. Prentice Hall.

Este libro de 330 páginas es la segunda edición del libro con el mismo título y autor del año 1992. La principal diferencia consiste en el desdoblamiento en dos capítulos del capítulo original de especificación e implementación de estructuras de datos y en la inclusión de soluciones para algunos de los ejercicios propuestos. Este libro se puede considerar interesante para la asignatura de Programación y Estructuras de Datos (capítulos 2, 5, 6 y 7). Se ajusta globalmente al enfoque docente propuesto en la programación de la asignatura, sin embargo, hay algunos detalles del libro que hay que tener en cuenta:

- En los capítulos 2 y 5 se describe un formalismo asociado a las funciones y procedimientos basados en el uso de la lógica de predicados, también conocida como técnica pre/post. Para los tipos de datos utiliza otra técnica formal que considera más apropiada: la especificación algebraica.
- No se describen los temas relacionados con los árboles de búsqueda del Bloque 3 del programa aquí propuesto: árboles 2-3, árboles 2-3-4, árboles rojos-negros y árboles B.
- No se describen los temas relacionados con los tipos especiales de conjuntos del Bloque 4: diccionario, trie y conjunto unión-búsqueda.
- No se describe ninguna aplicación de grafos de las indicadas en el Bloque 5.

Stroustrup (2002). *El lenguaje de programación C++*. Edición especial. Pearson Educación.

Un libro básico para cualquier estudiante que trabaje con C++. Escrito por el creador y diseñador original de C++, en sus más de 1000 páginas se presenta el lenguaje C++ al completo: desde una visión global de sus conceptos y características básicas hasta todos los aspectos del diseño y desarrollo de técnicas orientadas a objetos con C++. Contiene la implementación de algunos tipos abstractos

de datos sencillos (pila, cola, etc.), pero no profundiza en la implementación de tipos más complejos, ya que posee varios capítulos dedicados a la biblioteca estándar de C++ (listas, mapas, iteradores, funciones de ordenación, etc.).

Weiss (1995). *Estructuras de datos y algoritmos*. Addison-Wesley Iberoamericana.

Este libro de casi 500 páginas es una traducción del libro *Data Structures and Algorithm Analysis* de 1992 del mismo autor. Cubre el estudio de estructuras de datos clásicas y otras más avanzadas de una manera informal, utilizando como lenguaje de instrumentación el Pascal. Un libro del mismo autor que sigue el mismo contenido pero que utiliza el lenguaje C++ sería Weiss (2000). En cuanto a contenidos, no se tratan varios tipos de árboles avanzados: sólo trata los árboles AVL y árboles-B.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

A continuación se incluye una serie de libros como bibliografía complementaria. No se han incluido en la bibliografía básica por varias razones:

- Algunos sólo tratan una parte pequeña de los contenidos de Programación y Estructuras de Datos.
- Algunos son versiones en inglés de versiones que ya existen en español.
- Existen ediciones más modernas de algunos de ellos.

Sin embargo, los incluimos porque algunos libros son más fáciles de encontrar que sus correspondientes ediciones más modernas o en español.

Ammeral (1992). *Programs and Data Structures in C*. John Wiley & Sons.

Bishop (1986). *Data Abstraction in Programming Languages*. International Computer Science Series. Addison-Wesley.

Cairó & Guardati (1993). *Estructuras de datos*. McGraw-Hill.

Decker (1989). *Data Structures*. Prentice Hall.

Goguen & Tardo (1986). *An Introduction to OBJ: A Language for Writing and Testing Formal Algebraic Program Specifications*. Addison-Wesley.

Gotfried, B.S. 1991. *Programación en C*. McGraw-Hill.

Gutttag (1976). *The Design of Data Specifications*. Proceedings of the Second International Conference Software Engineering.

Horebeek (Horebeek, 1985). *Algebraic Specifications. From many-sorted algebras to a practical specification language*. Department of Computer Science. K.U. Leuven.

Kingston (1990). *Algorithms and Data Structures. Design, Correctness, Analysis*. Addison-Wesley.

Lippman (1991). *C++ Primer*. Addison-Wesley.

Liskov & Gutttag (1986). *Abstraction and Specification in Program Development*.

- Martin (1986). *Data types and data structures*. MIT Press.
- Meyers (1991). *Effective C++*. 50 Ways to Improve Your Programs and Designs. Addison-Wesley.
- Sedgewick (1992). *Algorithms in C++*. Addison-Wesley.
- Stroustrup (1993). *El lenguaje de programación C++*. Addison-Wesley.
- Weiss (2000). *Data structures and problem solving with C++*. Addison-Wesley.
- Wirth (1980). *Algoritmos + Estructuras de Datos = Programas*. Ediciones del Castillo.

7.3 RECURSOS DOCENTES

Al alumno se le proporcionan las normas de la asignatura, las transparencias de clase y los enunciados de las prácticas que tienen que realizar durante el curso. Pero además de estos recursos básicos, incluimos aquí otros recursos de especial interés, tanto para el profesorado como el alumnado, que proponemos como herramienta de apoyo a la docencia el Campus Virtual de la Universidad de Alicante. Las principales funcionalidades que ofrece son: publicación de la bibliografía, publicación de enlaces de interés, publicación de materiales, publicación de anuncios, publicación de calificaciones, fichas de alumnos, tutorías y

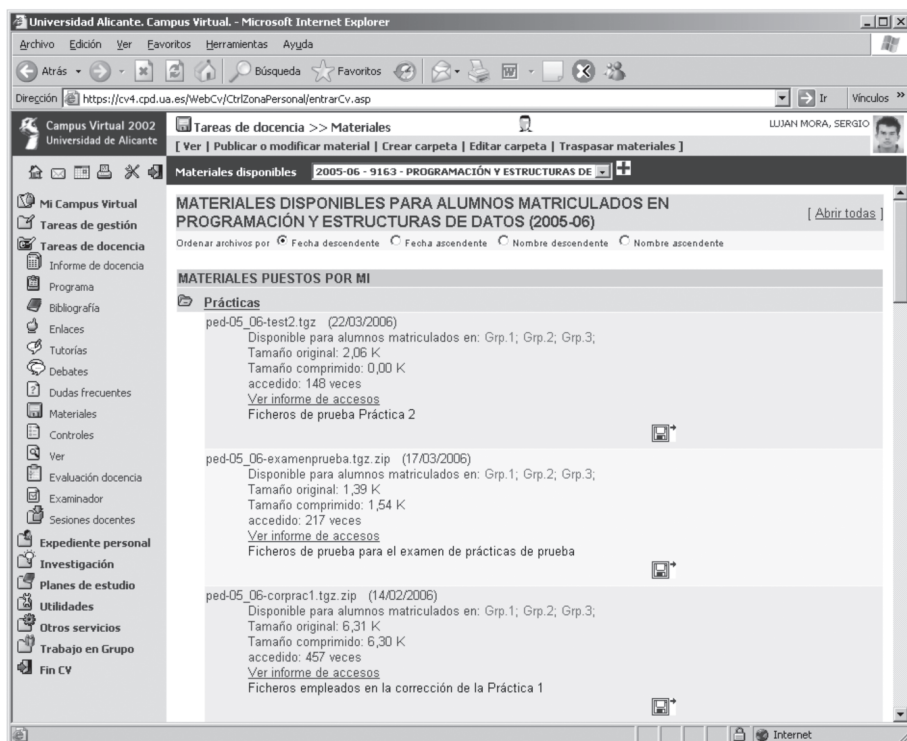


Ilustración 12. Sección de materiales disponibles en el Campus Virtual

debates. De las diversas opciones que proporciona el Campus Virtual, nos gustaría destacar tres: la publicación de materiales, la publicación de anuncios y las tutorías.

La publicación de materiales permite al profesor poner a disposición de los alumnos todos los materiales empleados en la docencia. Este sistema ofrece múltiples ventajas, como son la inmediatez, la seguridad y la posibilidad de acceder a los materiales desde cualquier ordenador con conexión a Internet. En la siguiente figura mostramos la página web de consulta de materiales publicados del Campus Virtual para nuestra asignatura. A través de esta página el profesor puede obtener una información valiosa: número de descargas de cada material publicado.

La publicación de anuncios permite hacer llegar al alumnado cualquier información que el docente considere necesaria.

Por otro lado, las tutorías a través del Campus Virtual proporcionan una excepcional vía de comunicación entre profesores y alumnos al permitir resolver ciertas dudas puntuales sin necesidad de producirse un contacto presencial. En la siguiente figura, se muestra una página web del Campus Virtual que visualiza una tutoría realizada por un alumno. Si así lo desea, el profesor puede acceder a la ficha del alumno.

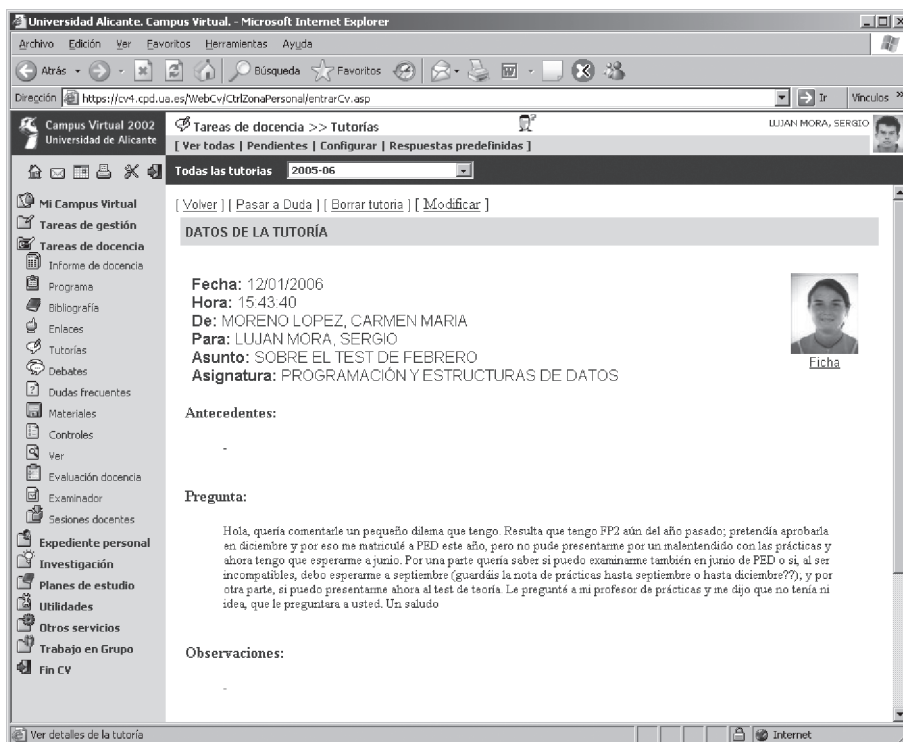


Ilustración 13. Tutoría a través del Campus Virtual

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces es fácil la resolución de dudas de esta forma. No obstante, cuando la duda es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.

Además, hemos desarrollado una aplicación web a la que hemos denominado Aprendizaje en la Web AutoMotivado (AWAM), que nos permite poner a disposición de los alumnos una gran cantidad de ejercicios que pueden resolver cuando ellos quieran (la aplicación está disponible las 24 horas del día) y desde donde ellos quieran (se puede acceder a la aplicación desde cualquier ordenador con acceso a Internet). Con el fin de motivar al alumno, hemos dotado a la aplicación de una apariencia de “competición” o juego que les anime a superarse. El funcionamiento de la aplicación es sencillo: se plantea al alumno una serie de ejercicios que tienen que resolver en un tiempo determinado y el juego finaliza cuando el alumno comete un número determinado de errores o contesta todos los ejercicios. Según el número de ejercicios contestados correctamente, el alumno obtiene una puntuación, que se traduce en una posición con respecto a los resultados de sus compañeros.

Los ejercicios planteados son de tipo test, agrupados por los diferentes temas que conforman el programa de la asignatura. Estos ejercicios inciden en los erro-

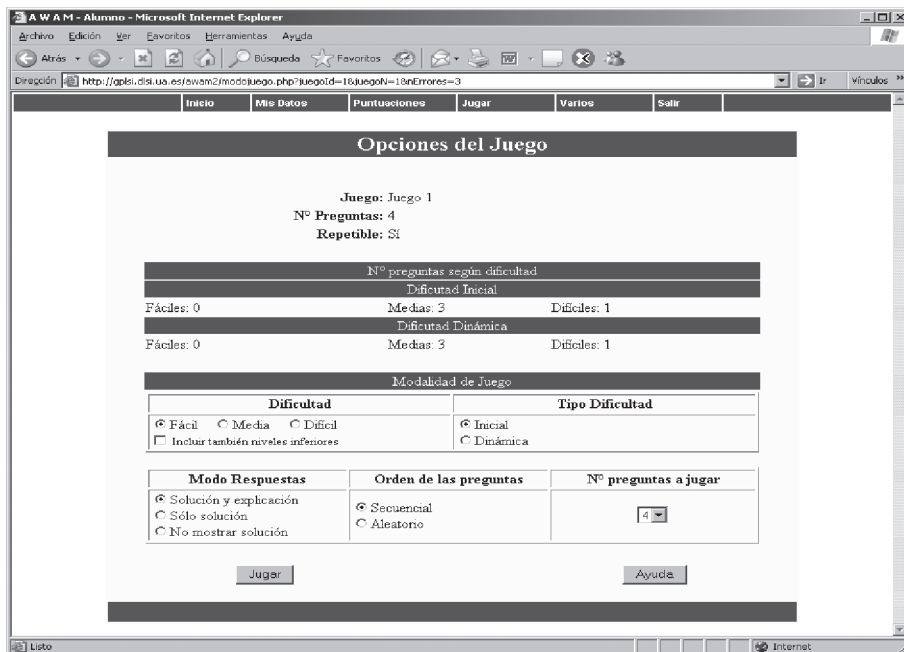


Ilustración 14. Juego de autoevaluación

res más comunes que cometen los alumnos a la hora de programar. Cada ejercicio tiene asignada una dificultad; si el profesor lo desea, se puede emplear este valor para presentar al alumno los ejercicios de menor a mayor dificultad. Además, también existe la opción de que esta dificultad se actualice en base a las contestaciones de los alumnos (cuantas más veces sea mal contestada una pregunta, más aumentará su dificultad). Por último, también existe la posibilidad de mostrar al alumno un comentario que explique la respuesta correcta de una pregunta cuando falle.

Finalmente, también empleamos una herramienta visual de simulación que implementa las diferentes estructuras de datos que se explican en la asignatura. Esta herramienta permite al alumno corregir sus ejercicios, verificar el comportamiento de las estructuras de datos en diversos escenarios, etc.

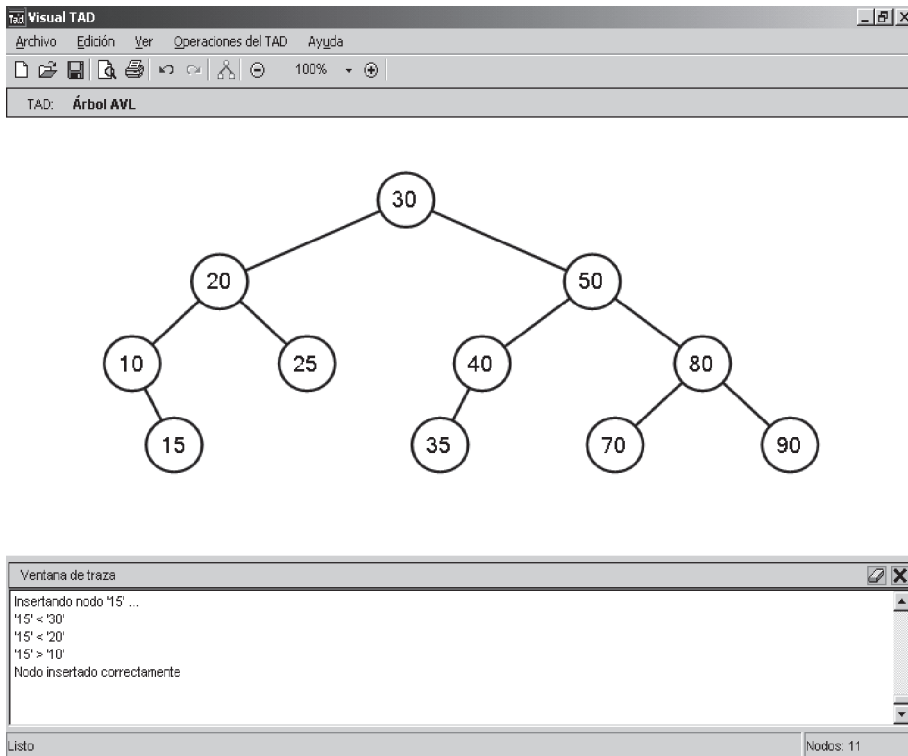


Ilustración 15. *Herramienta visual de visualización*

7.4 OTROS RECURSOS

7.4.1 Páginas web de interés:

Algunas páginas Web muy interesantes, que proporcionan enlaces o recopilaciones de enlaces a recursos relacionados con la asignatura son las siguientes:

- Estructuras dinámicas de datos:
<http://www.conclase.net/c/edd/index.php>
- Algoritmos y estructuras de datos:
<http://www.algoritmia.net/articles.php?folder=Estructuras%20de%20Datos>
- Estructura de datos:
http://es.wikipedia.org/wiki/Estructura_de_datos
- Thinking in C++:
<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>
- Curso de C++:
<http://decsai.ugr.es/~jfv/ed1/c++/cursoCpp.pdf>
- C++ Annotations:
<http://www.icce.rug.nl/documents/cplusplus/>
- Aprenda lenguaje ANSI C como si estuviera en primero:
http://proton.ucting.udg.mx/tutorial/c/leng_c.pdf

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

Como consecuencia del método docente planteado, la evaluación tendrá como misión fundamental cuantificar el grado de cumplimiento de los objetivos, además debe ser una contribución esencial al proceso formativo de los alumnos en forma de seguimiento y realimentación que valore sus esfuerzos y progresos y les ayude a aceptar el error como un elemento más del proceso de aprendizaje.

El método docente, considerado como realimentación, nos permitirá analizar el grado de cumplimiento de los objetivos, valorar el progreso de cada alumno y realizar un estudio de la eficacia del método docente.

Toda evaluación debe partir de los siguientes supuestos:

- Debe responder a normas predefinidas y conocidas de antemano por todos los profesores y alumnos.
- Debe ser coherente con los objetivos fijados.
- Debe ser objetiva y fiable.
- Debe abarcar todos los niveles de conocimiento y toda actividad del alumno con relación a la materia estudiada.
- Debe realizarse de forma continuada para facilitar las correcciones necesarias, tanto por parte del alumno, como del profesor.

El último supuesto no es viable en la asignatura debido al elevado número de alumnos de la asignatura tanto en las clases prácticas como en las teóricas. Por ello, la evaluación se basará en dos calificaciones: examen final escrito y prácticas.

La evaluación no deberá consistir solamente en valorar un documento escrito o unas prácticas, sino que deberá incluir la valoración de los demás aspectos del proceso educativo, es decir, pequeños ejercicios escritos, tutorías, actividades voluntarias, etc.

Los exámenes serán pruebas escritas que, aunque han sido criticadas por diversos motivos, inducen al alumno a estudiar con el único fin de superar el examen y siguen siendo la forma de evaluación más viable. Para intentar conseguir que estos exámenes sirvan mejor al proceso de aprendizaje debería cumplirse que:

- El examen sea una revisión global de la materia considerada.
- El examen sea corregido lo antes posible, discutiéndose las posibles respuestas, los errores que han aparecido, etc. Para ello se puede dejar una solución por escrito a disposición del alumno, donde se realicen los comentarios sobre el examen que el profesor considere oportuno.
- Las revisiones de los exámenes no se deben reducir a discutir para obtener la calificación necesaria para aprobar, sino que además de subsanar posibles errores cometidos en la corrección, se debe comentar al alumno los motivos de la calificación del examen.

Además de los exámenes escritos, el otro elemento fundamental en la evaluación son las prácticas de laboratorio. Estas prácticas se valorarán atendiendo a las memorias que los alumnos entregarán y a la defensa que realicen de las mismas.

La evaluación de la asignatura se basará en exámenes escritos y las prácticas de laboratorio. Sus características son las siguientes:

- Examen final escrito a la terminación del curso, que constará de dos partes:
- Una prueba de tipo test, con la que se pueda evaluar los conceptos básicos de la asignatura (40% de la nota del examen).
- Una prueba de preguntas no excesivamente largas, con la que se pueda evaluar la capacidad desarrollada por el alumno para aplicar los conceptos teóricos a la resolución de problemas (60% de la nota del examen).
- El peso relativo de este examen en la nota final será del 50%.
- Las prácticas de laboratorio se evaluarán atendiendo a las memorias que los alumnos entregan. Concretamente, se les convoca a los alumnos a la corrección de las prácticas. En la corrección, se evalúa el correcto funcionamiento de la práctica, así como el conocimiento individual que tengan de ella.
- El peso relativo de las prácticas en la nota final será del 50%.

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla presentamos el análisis de coherencia de la guía docente de Programación y estructuras de datos. En dicha tabla se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo propuesto para el alumnado y el sistema y criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
OI1	De CIC1 a CIC4 cCIC1 cCIC2	0 1					Enseñanza presencial (lección magistral / Trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a los mecanismos de abstracción.
OI2	De CIC6 a CIC19 cCIC1 cCIC2	0 1					Enseñanza presencial (lección magistral / Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los tipos de datos más usuales

Cuadro 3. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
O13	De CIC1 a CIC4	1					Enseñanza presencial (lección magistral /Trabajo de aula en grupos).Enseñanza no presencial (aprendizaje on-line/ biblioteca realización de ejercicios y prácticas). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
O14	De CIC1 a CIC19 cCIC2	1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio).Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
O15	De CIC1 a CIC19	1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio).Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas). Tutorías individualizadas.	<u>Procedimientos:</u> Prácticas. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual.
O16	De CIC6 a CIC19 cCIC2		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje online/ biblioteca/realización de ejercicios y prácticas propuestas).Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
O17	De CIC6 a CIC19		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 4. *Análisis de coherencia: competencias instrumentales (saber)*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
OI8	De CIC6 a CIC19		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
OI9	De CIC5 a CIC19	1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.

Cuadro 5. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber y hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
OI5	CIM1 CIM2 CIM3 CIM4 CIM8 CIM12 cCIM1, cCIM2	0 1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.

Cuadro 6. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
OI6	CIM5 CIM6 CIM7 CIM9 CIM10 CIM11 CIM14		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de destreza en la aplicación de los conocimientos a problemas planteados.
OI9	CIM1 CIM2 CIM3 CIM4 CIM8 CIM12	0 1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOI1	De CIM1 a CIM14 CIT1 CIT2 cCIT1		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (trabajo de aula en grupos/prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental, eficiencia y precisión en la resolución de problemas.
cOI2	CIT1 CIT2 cCIT1		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ realización de prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de destreza experimental.
cOI3	CIL1 cCIL1	0 1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestos).Tutorías individualizadas y organizadas.Memorias de los trabajos realizados.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. <u>Criterios:</u> Grado de rigurosidad en las explicaciones de los procedimientos aplicados.

Cuadro 7. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
		Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
cOI4	cCIL2	0 1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de conocimiento de los distintos términos de la asignatura, en castellano y/o valenciano y en inglés.
cOI5	De CIM1 a CIM14 cCIM3	0 1	2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza no presencial (aprendizaje on-line/biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos.
cOI6	CIC1 CIC2 cCIC1 cCIC2	0 1					Enseñanza presencial (lección magistral). Enseñanza no presencial (Aprendizaje on-line/ biblioteca).	<u>Procedimientos:</u> Trabajo en grupos. Trabajo complementario. <u>Criterios:</u> Grado de conocimiento del ámbito y aplicación de la asignatura.

Cuadro 8. *Análisis de coherencia: competencias instrumentales (saber hacer)*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
cOIP1	cCIPTC1 cCIPTC2		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y prácticas propuestos). Tutorías organizadas. Memorias de los trabajos realizados.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.

Cuadro 9. Análisis de coherencia: competencias interpersonales (saber y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS					PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)	Bloque 4 (temas)	Bloque 5 (temas)		
cOS1	cCS1 cCS4 cCS5		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los conocimientos aplicados.
cOS3	cCS1 cCS2 cCS4 cCS5		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/biblioteca/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2		2 3 4 5	6 7 8 9 10	11 12 13 14	15 16	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas	<u>Procedimientos:</u> Prácticas. Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación de los conocimientos aplicados.

Cuadro 10. Análisis de coherencia: competencias sistémicas

GUÍA DOCENTE DE SISTEMAS OPERATIVOS

David Gil Méndez; Pedro Fuentes Gómez

*Departamento de Tecnología Informática y Computación
Universidad de Alicante
(dgil;pfuentes)@dtic.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

A continuación se detallan las consideraciones específicas para la asignatura Sistemas Operativos 1I, en las que se define el marco conceptual concreto de la materia. Se tienen en cuenta los siguientes aspectos:

- Revisar el contenido de sistemas operativos en las recomendaciones de organismos internacionales de reconocido prestigio.
- Examinar los planteamientos de la asignatura en planes de estudio de otras universidades españolas.

La propuesta conjunta de ACM e IEEE, el *Computing Curricula 2001* [CC2001] o el *Computing Curriculum-Computer Engineering* [CE2004] dedica un área de conocimiento completa al estudio de los Sistemas Operativos [OS] distribuyendo la materia en un conjunto de contenidos fundamentales y optativos. Los contenidos que se incluyen son los siguientes:

Sistemas operativos (18 horas fundamentales)

- SO1: Introducción a los sistemas operativos
- SO2: Principios de los sistemas operativos
- SO3: Concurrencia de procesos
- SO4: Planificación de procesos
- SO5: Gestión de memoria
- SO6: Gestión de dispositivos

- SO7: Seguridad y protección
- SO8: Sistemas de archivos
- SO9: Sistemas embebidos y de tiempo-real y
- SO10: Sistemas tolerantes a fallos
- SO11: Evaluación del rendimiento del sistema
- SO12: Encriptación

Un sistema operativo representa una abstracción del comportamiento del hardware y un gestor de los recursos del sistema. En esta área se contemplan los temas que influyen en el diseño de los sistemas operativos actuales. Muchas de las ideas implicadas en uso del sistema operativo tienen aplicabilidad más amplia a través del campo de la informática, tal como programación concurrente. Estudiar el diseño de los sistemas operativos tiene influencia en muchas otras áreas de la informática tales como la programación, diseño y puesta en práctica confiable, desarrollo moderno del dispositivo, ambientes virtuales, internet, sistemas seguro y muchos otras.

Se propone su estudio específico en cursos de nivel medio, ya sea de contenido exclusivo de sistemas operativos *Operating Systems* (CS225) o relacionados con otras materias: *Architecture and Operating Systems* (CS221), *Architectures for Networking and Communication* (CS222) y *Operationg Systems and Networking* (CS226). Estos conocimientos pueden ser ampliados mediante un conjunto de cursos avanzados.

Con relación a la docencia de la asignatura en la universidad española, destaca el carácter troncal de la materia en el primer ciclo con 9 créditos asignados, lo que pone de manifiesto su relevancia en el currículo del ingeniero en informática. El encaje de la materia en los planes de estudio de las universidades españolas se realiza en el primer ciclo y en cursos medios. De su análisis se extraen las siguientes conclusiones:

- La mayoría de las universidades ofrecen asignaturas básicas, avanzadas y un conjunto de asignaturas optativas afines a sistemas operativos.
- Respecto a los contenidos teóricos, no suele haber grandes diferencias, si acaso destacar el hecho de que profundizan más en determinados aspectos de la materia, motivados principalmente por un mayor número de créditos.

En la Universidad de Alicante se han distribuido los contenidos que se imparten en la asignatura de sistemas operativos para la Ingeniería Técnica en Informática de Sistemas. Dichos contenidos constituyen uno de los pilares básicos sobre los que se sustenta la formación de los ingenieros en informática. Las aportaciones de los sistemas operativos a los diversos perfiles profesionales se relacionan en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de computadores. • Conocimientos y habilidades en el diseño de sistemas operativos. • Conocimientos básicos del funcionamiento de microprocesadores. • Conocimientos de computación concurrente.
Arquitectura y diseño de software	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de computadores. • Conocimientos y habilidades en el diseño de sistemas operativos.
Diseño multimedia	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de sistemas operativos. • Conocimientos básicos de microprocesadores y computación paralela. • Conocimientos y habilidades en el ámbito del multimedia, a nivel de procesamiento concurrente y administración de recursos.
Ingeniería de comunicación de datos	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos y habilidades en el diseño de sistemas operativos. • Nociones fundamentales sobre comunicación digital y sistemas distribuidos.
Diseño de redes de comunicación	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Nociones fundamentales sobre comunicación digital y sistemas distribuidos.
Asistencia técnica	<ul style="list-style-type: none"> • Conocimientos y habilidades en el mantenimiento de sistemas operativos. • Conocimientos sobre especificaciones del hardware de computadores y sus periféricos.
Ingeniería de integración y pruebas e implantación y pruebas	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Habilidades de utilización de herramientas de simulación para validación de diseños. Realización de prototipos para pruebas.
Consultoría en empresas de TI	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Formalización de especificaciones y requerimientos del usuario. • Conocimientos básicos sobre tecnología de computadores.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento de la metodología científica para la resolución de problemas. • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos básicos de Sistemas de tiempo real en la industria, robótica y fabricación, automatización de procesos industriales, tratamiento digital de la imagen. Enfoque industrial de los sistemas de control informático.
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento de la metodología científica para la resolución de problemas. • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos básicos de Sistemas de tiempo real en la industria, robótica y fabricación, automatización de procesos industriales, tratamiento digital de la imagen. Enfoque industrial de los sistemas de control informático. • Conocimientos y habilidades en el diseño de proyectos informáticos y aplicación de herramientas informáticas en la empresa.

Cuadro 1. Adecuación de los tópicos relacionados con la asignatura *Sistemas Operativos*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

A continuación, se analiza el contexto local de la asignatura describiendo su relación con el resto de materias de la titulación según el plan de estudios. La figura siguiente ilustra dicha relación:

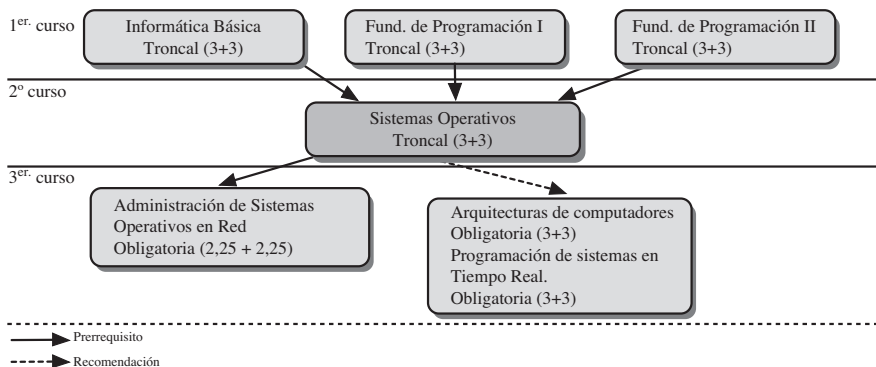


Ilustración 16. Relaciones de *Sistemas Operativos* en el plan de estudios

Como se aprecia en la figura, el estudio de los sistemas operativos comienza en segundo curso, una vez que el alumno conoce parte de los conceptos previos necesarios, y se extiende hasta el tercer curso.

Sistemas Operativos abarca el estudio de la totalidad los aspectos de un sistema operativo completo, tanto desde un punto de vista conceptual como ingenieril.

Se cursa también en el segundo curso, Estructuras de Computadores lo que posibilita que los alumnos tengan una base más sólida del hardware del computador y puedan comprender mejor los sistemas operativos y su relación con el hardware.

La asignatura se sustenta en los conocimientos adquiridos en Estructuras de Computadores sobre la organización interna y funcionamiento de los computadores, y en los conocimientos adquiridos en Fundamentos de la Programación I y II, en cuanto a conceptos generales de programación en lenguajes de medio y alto nivel.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos para la asignatura de Sistemas Operativos:

- OI1: Comprender nociones básicas del funcionamiento de los sistemas operativos así como las relaciones fundamentales entre los diferentes módulos que conforman la arquitectura de los sistemas operativos.
- OI2: Conocer los conceptos y problemas básicos de la gestión de recursos de un sistema informático.
- OI3: Conocer la estructura lógica de un planificador de procesos y las diferentes alternativas o algoritmos de asignación de elementos de proceso.
- OI4: Comprender los conceptos y problemas básicos de la gestión de memoria y saber diseñar soluciones y en particular los aspectos relacionados con la memoria virtual.
- OI5: Conocer la estructura lógica de un sistema de archivos y la gestión del almacenamiento.
- OI6: Analizar la problemática y técnicas de gestión de los dispositivos de entrada/salida y saber diseñar manejadores.
- OI7: Aplicar los conceptos teóricos y prácticos del diseño de sistema operativos y así captar su relación con ramas específicas como las Arquitecturas de Computadores, las Redes de Computadores y el resto de asignaturas de estudio de los sistemas operativos.

2.1.2 Objetivos interpersonales generales

Estos objetivos se corresponden con los objetivos interpersonales generales, comunes a todas las asignaturas de primer curso. Concretamente, estos son los objetivos cOIP1, cOIP2 y cOIP3 detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Además de los objetivos cOS1, cOS2, cOS3 y cOS4 considerados en el artículo introductorio de este libro, incluimos aquí los siguientes:

- OS1: Asumir los esquemas de teoría, abstracción y diseño como método que permite impulsar la capacidad para abstraer y generalizar los problemas, así como asimilar los rápidos avances en la disciplina y situarlos en su contexto de innovación científica y tecnológica.
- OS2: Desarrollar el espíritu crítico tanto para enfrentarse a un problema como para evaluar las ventajas e inconvenientes de un diseño concreto.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales

Dentro de las competencias instrumentales, distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

La asignatura Sistemas Operativos contiene los siguientes temas:

- Tema 1: Introducción. Definiciones y conceptos
- Tema 2: Gestión de procesos
- Tema 3: Comunicación y sincronización de procesos. Interbloqueos
- Tema 4: Gestión de memoria
- Tema 5: Sistema de archivos
- Tema 6: Gestión de Entrada/Salida

Desarrollamos los apartados anteriores para cada uno de ellos.

2.2.2 Habilidades cognitivas

Además de las habilidades comunes cCIC1 y cCIC2 consideradas en el artículo introductorio de este libro, incluimos las siguientes, agrupadas por temas de la asignatura.

Tema 1: Introducción. Definiciones y conceptos

- CIC1: Conocer los conceptos fundamentales de los sistemas operativos y adquirir una visión básica, al tiempo que global y homogénea entre todos los alumnos.
- CIC2: Concebir el estudio de los sistemas operativos como una rama de la informática y comprender y conocer sus interrelaciones con el resto de la disciplina.

- CIC3: Desarrollar la capacidad de abstracción de los alumnos mediante la exposición de los paradigmas característicos de teoría, abstracción y diseño, junto a los conceptos recurrentes de la disciplina informática.

Tema 2: Gestión de procesos

- CIC4: Comprender la problemática de la ejecución de procesos en sistemas multiprogramados y conocer los métodos de planificación de procesos más comunes.
- CIC5: Aprender las técnicas de simulación de estrategias de planificación y adquirir la capacidad de resolver las situaciones de conflicto.

Tema 3: Comunicación y sincronización de procesos. Interbloqueos

- CIC6: Comprender la problemática de la gestión de recursos en un sistema informático y en especial el problema de la exclusión mutua.
- CIC7: Conocer los principales modelos y herramientas de resolución de problemas de comunicación y sincronización de procesos.
- CIC8: Comprender las causas y condiciones que provocan situaciones de interbloqueo en el acceso de a los recursos del sistema.
- CIC9: Comprender las técnicas y mecanismos de prevención y evitación de situaciones de interbloqueo.

Tema 4: Gestión de memoria

- CIC10: Conocer los conceptos básicos de gestión de memoria.
- CIC11: Describir métodos básicos de asignación de memoria real.
- CIC12: Analizar los problemas de los esquemas de gestión de memoria real, identificando la necesidad de un esquema de memoria más sofisticado: la memoria virtual.
- CIC13: Describir y comparar los distintos esquemas de asignación de memoria virtual y analizar diferentes casos de estudio.

Tema 5: Sistema de archivos

- CIC14: Conocer la estructura lógica de un sistema de archivos: archivos y directorios.
- CIC15: Describir la gestión de archivos y directorios.
- CIC16: Describir métodos de implementación física del sistema de archivos.

Tema 6: Gestión de Entrada/Salida

- CIC17: Clasificar los dispositivos de entrada/salida atendiendo al flujo de información con el que trabaja.
- CIC18: Enumerar los objetivos del software de entrada/salida, justificando la necesidad de estructurarlo en capas para conseguir dichos objetivos y describir las funciones de cada una de dichas capas.

2.2.3 Capacidades metodológicas

Además de las habilidades comunes cCIM1, cCIM2 y cCIM3 consideradas en el artículo introductorio de este libro, incluimos las siguientes agrupadas por temas de la asignatura.

Tema 1: Introducción. Definiciones y conceptos

- CIM1: Ser capaz de identificar los aspectos de teoría, abstracción y diseño característicos en el contexto concreto de los sistemas operativos.
- CIM2: Reconocer la arquitectura de un sistema informático y el tipo de sistema operativo que contiene.

Tema 2: Gestión de procesos

- CIM3: Ser capaz de identificar el modelo de planificación más adecuado para un sistema operativo y una determinada carga de trabajo.
- CIM4: Realizar simulaciones y diagramas de la ejecución de los procesos de un sistema de acuerdo a una planificación.

Tema 3: Sincronización y comunicación de procesos. Interbloqueos

- CIM5: Familiarización con los modelos y herramientas de gestión de la concurrencia.
- CIM6: Adquirir destreza en la resolución de situaciones de sincronización y comunicación entre procesos.
- CIM7: Aprender a identificar una situación de interbloqueo.
- CIM8: Ser capaz de aplicar las estrategias de prevención, evitación y recuperación para soslayar situaciones de interbloqueo en un sistema informático.

Tema 4: Gestión de memoria

- CIM9: Ser capaz de realizar una gestión de memoria real eficiente, identificando la necesidad de un esquema de memoria más sofisticado: la memoria virtual.
- CIM10: Trabajar con los distintos esquemas de asignación de memoria virtual.

Tema 5: Sistema de archivos

- CIM11: Estudiar la gestión del espacio en disco para que funcione de forma fiable y eficiente, analizando diferentes casos de estudio.
- CIM12: Construir una estructura que contenga los archivos y directorios de un sistema de archivos

Tema 6: Gestión de Entrada/Salida

- CIM13: Conocer los dispositivos de entrada/salida más comunes en un computador y cómo se gestionan poniendo especial atención en los dispositivos de bloques. Analizar diferentes casos de estudio.

Destrezas tecnológicas

Además de la destreza cCIT1 considerada en el artículo introductorio de este libro, incluimos aquí las siguientes destrezas tecnológicas desarrolladas para todos los bloques.

- CIT1: Ser capaz de manejar aplicaciones y programas. Familiarizarse con el manejo de ventanas, menús y componentes.
- CIT2: Usar con habilidad y destreza un sistema operativo real. Caso concreto de Linux.
- CIT3: Usar con destreza librerías de funciones y llamadas al sistema para programar sencillos programas de aplicación.
- CIT4: Adquirir destrezas en el diseño de módulos de un sistema operativo de acuerdo a unos determinados requerimientos.

Destrezas lingüísticas

Las destrezas lingüísticas abarcan las destrezas comunes cCIL1 y cCIL2, y las siguientes desarrolladas para todos los bloques.

- CIL1: Saber desarrollar cualquier tema, tanto de forma oral como escrita, en el que se tengan que aplicar conceptos en el ámbito de los sistemas operativos.
- CIL2: Saber desarrollar cualquier tema, tanto de forma oral como escrita, en el ámbito de la programación de rutinas en linux usando llamadas al sistema e interrupciones del sistema operativo.

2.2.4 Competencias interpersonales

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas: quedan establecidas en las competencias comunes cCIPTC1 y cCIPTC2, enunciadas en el capítulo introductorio de este libro.

Compromiso con el trabajo: las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el capítulo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.5 Competencias sistémicas

Integración de capacidades cognitivas, destrezas prácticas y disposiciones

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el capítulo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCs3, cCs4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

Las competencias y contenidos mínimos adecuados para superar esta asignatura quedan definidas por los prerrequisitos oficiales que tiene en el plan de estudios de la titulación: Informática Básica, Fundamentos de Programación I y II.

Además de estos requisitos, es conveniente saber lo siguiente:

- Manejar un sistema operativo a nivel de usuario.
- Entender las diferencias entre teoría abstracción y diseño.
- Poseer una capacidad de abstracción adecuada.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los profesores de la asignatura recomendamos a todos los alumnos diversas lecturas para entender algunos aspectos fundamentales de los sistemas operativos, en especial aquellas que relacionan los conceptos adquiridos en primer curso con los sistemas operativos. Esta información la tienen disponible en el campus virtual y en la página web de la asignatura.

Además publicamos, al comienzo de cada curso, en el campus virtual, un test de autoevaluación “test cero” para que cada alumno pueda comprobar cuál es su nivel sobre la materia que va a estudiar. Según los resultados obtenidos, se le recomendará una determinada trayectoria que le facilitará el aprendizaje de la asignatura para su puesta al día.

4. BLOQUES Y TEMAS DE CONTENIDO

4.1 DESARROLLO DE LOS TEMAS O UNIDADES DE CONTENIDO

La asignatura se compone de los siguientes temas de contenido, como ya se ha mencionado anteriormente:

- Tema 1: Introducción. Definiciones y conceptos
- Tema 2: Gestión de procesos
- Tema 3: Comunicación y sincronización de procesos. Interbloques
- Tema 4: Gestión de memoria
- Tema 5: Sistema de archivos
- Tema 6: Gestión de Entrada/Salida

Desarrollo del Tema 1: Introducción. Definiciones y Conceptos

Contenidos de teoría:

- 1.1 Concepto y funciones del sistema operativo
- 1.2 Evolución histórica de los sistemas operativos
 - 1.2.1 Evolución histórica
 - 1.2.2 Sistemas operativos actuales
- 1.3 Estructura de los sistemas operativos

1.4 Principios de diseño de sistemas operativos

1.4.1 Metodologías

1.4.2 Modelos

Contenidos de práctica:

1.5 Introducción

1.6 Conexión y desconexión del sistema

1.7 Edición, compilación y ejecución de programas

1.8 Órdenes básicas de UNIX

Desarrollo del Tema 2: Gestión de procesos

Contenidos de teoría:

2.1 Conceptos de procesos

2.1.1 Modelo de procesos

2.1.2 Multiprogramación

2.1.3 Estados de un proceso

2.1.4 Operaciones sobre procesos

2.1.5 Implementación de procesos

2.2 El núcleo del sistema operativo

2.2.1 Cambio de contexto

2.2.2 Manejador de interrupciones, despachador y planificador

2.3 Hilos de ejecución

2.3.1 Concepto de hilo de ejecución

2.3.2 Implementación de hilos de ejecución

2.4 Planificación de procesos

2.4.1 Planificación de procesos

2.4.2 Algoritmos de planificación

2.4.3 Evaluación de algoritmos de planificación

2.5 Diseño de estructuras

2.5.1 Bloque de control de proceso

2.5.2 Tabla de procesos

Contenidos de práctica:

2.1 Introducción

2.2 Descripción. Algoritmos de planificación

2.3 Restricciones y consideraciones

2.4 Implementación

Desarrollo del Tema 3: Comunicación y sincronización de procesos.

Interbloqueos

Contenidos de teoría:

3.1 Especificación de la concurrencia

3.1.1 Necesidad de coordinación y cooperación entre procesos

- 3.1.2 Condiciones de carrera
- 3.1.3 El problema de la sección crítica
- 3.2 Primitivas de coordinación y sincronización
 - 3.2.1 Sincronización mediante hardware
 - 3.2.2 Sincronización mediante software. Semáforos. Monitores. Mensajes
- 3.3 Problemas clásicos y ejemplos de resolución
 - 3.3.1 Interbloqueo. Especificación del problema
- 3.4 Ejemplos de interbloqueos
- 3.5 Tipos de recursos y caracterización de interbloqueos
 - 3.5.1 Grafos de recursos reutilizables y consumibles
 - 3.5.2 Sistemas con recursos reutilizables y consumibles
- 3.6 Prevención y evitación
- 3.7 Detección de interbloqueos y recuperación del sistema

Contenidos de práctica:

- 3.1 Introducción
- 3.2 Descripción. El problema del productor/consumidor
- 3.3 El problema de lectores/escritores FIFO
- 3.4 Restricciones y consideraciones
- 3.5 Implementación
- 3.6 Interbloqueo. El problema de la comida de filósofos
- 3.7 El algoritmo del banquero
- 3.8 Implementación

Desarrollo del Tema 4: Gestión de memoria

Contenidos de teoría:

- 4.1 Conceptos de gestión de memoria
 - 4.1.2 Requisitos
 - 4.1.3 Organización
 - 4.1.4. Memoria real y virtual
 - 4.1.5 Intercambio
- 4.2 Memoria real. Asignación contigua
 - 4.2.1 Particiones fijas
 - 4.2.2 Particiones dinámicas
- 4.3 Memoria real. Asignación dispersa
 - 4.3.1 Paginación
 - 4.3.2 Segmentación
 - 4.3.3. Técnicas combinadas
- 4.4 Memoria virtual
 - 4.4.1. Paginación por demanda

- 4.4.2 Segmentación por demanda
- 4.4.3. Sistemas combinados
- 4.4.4. Otras consideraciones
- 4.5 Casos de estudio: Unix/Linux

Contenidos de práctica:

- 4.1 Simulación de un gestor de memoria de particiones dinámicas
 - 4.1.1 Identificación de la fragmentación externa
 - 4.1.2 Compactación
 - 4.1.3 Comparar los algoritmos de asignación de huecos (primer hueco, mejor hueco, pero hueco y siguiente hueco)
- 4.2 Simulación de un gestor de memoria virtual
 - 4.2.1 Comparar los algoritmos de sustitución de páginas

Desarrollo del Tema 5: Sistema de archivos

Contenidos de teoría:

- 5.1 Archivos y directorios
 - 5.1.1. Tipos
 - 5.1.2 Métodos de acceso
 - 5.1.3 Operaciones
 - 5.1.4 Estructuras
- 5.2 Implantación del sistema de archivos
 - 5.2.1 Gestión del almacenamiento secundario
 - 5.2.2 Gestión del espacio libre
- 5.3 Casos de estudio: unix/linux

Contenidos de práctica:

- 5.4 Sistema de archivos de Linux basado en estructuras de nodos-i
- 5.5 Llamadas al sistema de Linux para la gestión de archivos (stat,fstat, mknod, etc.)

Desarrollo del Tema 6: Gestión de Entrada/Salida

Contenidos de teoría:

- 6.1 Introducción. El problema de E/S
- 6.2 Dispositivos de E/S
 - 6.2.1 Dispositivos de bloques
 - 6.2.2. Dispositivos de caracteres
- 6.3 Software de E/S
- 6.4 Dispositivos de bloques. Discos
- 6.5 Dispositivos de caracteres. Terminales
- 6.6 Otros dispositivos. Relojes
- 6.7 Casos de estudio: Unix/Linux

Contenidos de práctica:

- 6.1 Manejadores de dispositivos
- 6.2 Conocer el funcionamiento de la entrada/salida en Linux, así como la filosofía de funcionamiento de los manejadores de dispositivos
- 6.3 Relacionar el dispositivo de E/S y el manejador para constituir el periférico

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente que se ha desarrollado en el capítulo introductorio de este libro para las asignaturas de primer curso de Ingeniería Técnica en Informática de Sistemas, junto con la que aportamos en este apartado, conforma la metodología de la asignatura de *Sistemas Operativos*. Se ha tenido en cuenta el alto contenido tecnológico de la asignatura para dotar los mecanismos necesarios que mejoren el aprendizaje. Por lo tanto, presentamos un modelo en el que además de los recursos o métodos tradicionales de enseñanza, como la lección magistral, introducimos otros para el mejor seguimiento de la misma, pensado que de esta manera el alumno se sentirá más motivado.

Las actividades que se proponen son:

- Clases de teoría y ejercicios, en aulas.
- Clases de prácticas en laboratorios informáticos.
- Actividades en grupos pequeños.
- Trabajos complementarios.
- Tutorías organizadas.

Clases de teoría y ejercicios

Objetivos generales

- Preparar al alumno para que sea un buen conocedor de la materia teórica propia de la asignatura que va a aprender y conocer sus aplicaciones en el campo que le interese.
- Despertar la motivación y el interés por la asignatura así como desarrollar su curiosidad intelectual y el interés por el rigor científico.
- Facilitar las orientaciones oportunas para el aprendizaje de cada materia.
- Favorecer la realización de ejercicios individuales que puedan llevar a una discusión colectiva. Se fomenta una reflexión cualitativa previa, deteniéndose en la clarificación de los conceptos.
- Facilitar la retroalimentación de conocimiento que fomente el aprendizaje constructivo.
- Desarrollar el espíritu crítico y una actitud abierta ante cambios de todo tipo y en especial a los científico-técnicos de su especialidad.

- Inculcar los principios deontológicos y adquirir conciencia del sentido humanístico de la ciencia y la técnica.

Metodología

- **Clase cero:** preparación que el alumno recibe para la siguiente clase. El profesor usará recursos web en donde publicará ejercicios y test de autoevaluación para “preevaluar” el tema teórico que se comenzará en la siguiente clase de teoría. Recomendación de libros, artículos, links de interés y apuntes o notas de relacionadas.
- **Lección magistral:** guión de la clase, objetivos y exposición de contenidos teóricos.
- **Incentivos para participar en clase:** preguntar si tienen conocimientos previos del tema que se va a impartir, dónde los han adquirido y lo que recuerden de ese aprendizaje. Hay que motivar al alumno para que participe en clase haciendo preguntas al profesor.
- **Recursos y medios audiovisuales:** uso de pizarra y/o pizarra táctil, transparencias y retroproyector, diapositivas, modelos, cañón de proyección u ordenador (según las características del aula).
- **Comentar bibliografía recomendada:** indicar los libros, apuntes o enlaces web que son apropiados para el seguimiento del tema.
- **Material de apoyo:** proponer ejercicios para realizarlos en grupo y cuestionarios de autoevaluación para hacerlos individualmente.
- **Opcionales:** motivar la realización de algún pequeño trabajo de investigación para reforzar el tema que acaban de aprender. Buscar aplicaciones. Pedir resumen del tema.

Clases de prácticas en laboratorios de informática

Objetivos generales

- Proporcionar experiencia y madurez en la aplicación de los principios desarrollados en teoría al diseño, prueba de software y hardware práctico, facilitando su comprensión y desarrollando un saber hacer en computación.
- Introducir los métodos experimentales y presentar correctamente los descubrimientos mediante la elaboración de informes.
- Integrar la actividad práctica con las lecciones teóricas, definiendo proyectos de laboratorio con una secuenciación adecuada —introducción, resolución de problemas y diseño creativo—, una planificación cuidadosa y una buena sincronización con el desarrollo de la teoría.

Metodología

- **Clase cero:** a través del campus virtual se informará tanto del contenido de las clases que se impartirán durante el curso, como de lo que deben prepa-

rar para llevar a cabo cada sesión (matriculación, material, asignación de profesores y horarios).

- **Sesión práctica:** presentación de los contenidos que se desarrollarán en esa clase. Se organiza de manera gradual la realización del proyecto de laboratorio a tres escalas:

5. *Introducción.* En primer lugar se estudia y comprende un sistema o programa ya diseñado, familiarizando al alumno con herramientas básicas y consolidando conceptos. A continuación se observa el comportamiento derivado de modificaciones o ampliaciones del sistema

6. *Resolución de problemas.* Se amplía la experiencia inicial y se desarrolla la autoconfianza en la habilidad de resolver problemas mediante la resolución de tareas de complejidad creciente y la propuesta de soluciones en entornos diversos.

7. *Diseño creativo.* Refuerza la habilidad de resolución de problemas en contextos más complejos mediante la propuesta y evaluación crítica de soluciones, la toma de decisiones y el trabajo en equipo.

- **Recursos y medios audiovisuales:** se usará pizarra y cañón de proyección u ordenador.
- **Comentar bibliografía recomendada:** indicar los libros o apuntes que son apropiados para el seguimiento de las clases.
- **Material de apoyo:** al final de cada bloque de prácticas, se repartirá un cuestionario de autoevaluación individual.
- **Opcionales:** proponer realizar algún trabajo de mayor envergadura que los propuestos en el enunciado y que se podrán hacer en grupos pequeños.

Actividades en grupos pequeños

Objetivos generales

- Potenciar la actividad de trabajar en grupo (grupos pequeños), para desarrollar la relación del alumno con sus compañeros.
- Preparar al alumno para desarrollar temas de interés y motivar aprendizaje de investigación.

Actividades

- Hacer ejercicios y cuestionarios teóricos y prácticos.
- Hacer la lectura de libros, artículos o revistas recomendadas en la asignatura entre los miembros del grupo y después hacer comparaciones entre ellos.
- Realizar un trabajo de prácticas de mayor envergadura que la diseñada para las clases de prácticas, usando el ordenador.

Trabajos complementarios

Objetivos generales

Aprender a trabajar de forma individual, haciendo una investigación.

Actividades

Hacer de forma voluntaria trabajos de índole particular que podrán tener tanto carácter teórico como aplicado.

Tutorías organizadas

Objetivos generales

- Fomentar que el alumno acuda a tutorías para conseguir una relación profesor/alumno más personal.
- Realizar un seguimiento controlado del proceso de aprendizaje del alumno a lo largo del curso.

Actividades

- Hacer consultas relacionadas con la materia de teoría para dirigir el aprendizaje.
- Hacer consultas acerca de los trabajos propuestos en clase.
- Coordinar la realización de los ejercicios y trabajos en grupo.
- Hacer consultas dirigidas al seguimiento de sus estudios en Informática.

5.2 ESTRATEGIAS DE APRENDIZAJE

Es la forma en que el alumno debe prepararse para seguir cada uno de los puntos de la metodología empleada en el aprendizaje de la asignatura.

Estrategia para las clases teóricas

- El alumno debe leer las directrices marcadas en la clase cero.
- Obtener y leer el material que el profesor ha puesto a disposición del alumno para el desarrollo de cada clase.
- Una vez en clase, debe hacer un seguimiento de la misma preguntado todo lo que sea necesario, en las pautas establecidas por el profesor.
- Después de clase se deben consultar apuntes, libros o enlaces web recomendados
- Para cualquier duda, acudir a tutorías y /o consultar con los compañeros.

Estrategia para las clases de prácticas en laboratorio

- El alumno debe leer las directrices marcadas en la clase cero.
- Llevar material para la elaboración de las prácticas: dispositivos de almacenamiento, enunciado del trabajo a realizar y apuntes del tema.

- Una vez en clase, debe hacer un seguimiento de la misma preguntado todo lo que sea necesario, en las pautas establecidas por el profesor.
- El profesor dirigirá en todo momento al alumno para la realización de las prácticas dándole consejos y estrategias.
- Cada alumno debe analizar el grado de conocimiento acerca de la práctica que está realizando para superar la evaluación individual de la misma.

Estrategia para las actividades de los grupos pequeños

- El profesor repartirá trabajos y ejercicios para realizarlos en grupos pequeños.
- Recomendará al grupo la realización de trabajos relacionados con algún tema de la materia.
- El profesor establecerá unas determinadas pautas de trabajo.
- En horas de tutorías, se hará el seguimiento del trabajo.

Estrategia para hacer trabajos complementarios

- El profesor repartirá trabajos y ejercicios para realizarlos de forma individual o colectiva y los recomendará de acuerdo a las pretensiones de cada alumno.
- El profesor establecerá unas determinadas pautas de trabajo.
- En horas de tutorías se hará el seguimiento y la coordinación de los trabajos complementarios.

Estrategia para hacer una autoevaluación

La autoevaluación va dirigida a que el alumno sepa en todo momento su grado de aprendizaje de la asignatura, lo primero porque le interesa saber si va a ir bien preparado al examen (lo que más le interesa), y lo segundo porque aprenderá conceptos que le servirán para el aprendizaje de otras materias. Por ello proponemos:

- Realización de ejercicios básicos al finalizar cada tema para afianzar los conocimientos adquiridos.
- Realización de ejercicios más complejos con la intención de preparar al alumno para que vaya tomando conciencia de las dificultades que se puede encontrar en los exámenes.
- Realización de pruebas de autoevaluación.
- Si lo cree conveniente, puede volver a repetir las pruebas tantas veces como quiera.

Estrategia para hacer una evaluación final

Constituye el instrumento docente de validación de los demás mecanismos que intervienen en el proceso de enseñanza-aprendizaje. Es el proceso, sin duda complejo, que comprende la obtención de información útil acerca del grado de

asimilación de los conceptos, el nivel de adiestramiento conseguido en lo aplicado y la adecuación de la actitud hacia los objetivos. Este proceso permitirá obtener valoraciones, a partir de las cuales tomar decisiones sobre la conveniencia de reorientación del proceso educativo.

Si el alumno ha llevado un seguimiento de la asignatura marcado por las pautas indicadas por los profesores, se supone que debe ir bien preparado para la evaluación final de la asignatura que consistirá en un examen teórico. Por otra parte, la evaluación de las clases prácticas de forma continuada hace que el alumno siga una estrategia para trabajar en cada clase de prácticas y así al finalizar las mismas tiene su evaluación final respectiva, casi siempre satisfactoria.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

Para obtener el número de créditos necesarios para completar la asignatura, cada alumno debe realizar un plan de trabajo que está dividido en actividades presenciales y no presenciales. Las horas que debe dedicar a las actividades presenciales se reparten entre las clases de teoría, de prácticas en laboratorio y las actividades en grupos pequeños. Para las no presenciales, el tiempo dedicado a las clases teóricas está repartido entre la clase cero y el test de autoevaluación que se hace después de cada tema; para las actividades en grupos pequeños se hace un reparto de tiempo dedicado a la lectura y comprensión de los ejercicios propuestos y corregidos por el profesor y entregados a cada grupo. Las horas dedicadas a la preparación del examen de teoría y las de tutorías organizadas completan las que debe realizar el alumno para sacar adelante la asignatura.

En las tablas siguientes se propone el número de horas que debe dedicar cada alumno para el conocimiento de la asignatura.

HORAS DE ACTIVIDADES PRESENCIALES				
Actividad		Clases de Teoría	Clases de Prácticas	Actividades en Grupos Pequeños
Presentación		2		2
fundamentos de los sistemas operativos	Tema 1	2	2	
gestión de procesos. concurrencia e interbloqueos	Tema 2	3	3	2
	Tema 3	3	5	4
gestión de memoria	Tema 4	4	6	4
gestión de archivos	Tema 5	3	2	2
gestión de entrada/salida	Tema 6	2	2	2
Examen Final	Preparación	4		
	Duración	3		
Total:	62	26	20	16

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

HORAS DE ACTIVIDADES NO PRESENCIALES				
Actividad		Clases de Teoría	Clases de Prácticas	Actividades en Grupos Pequeños
Presentación		3,5		0,5
fundamentos de los sistemas operativos	Tema 1	5	2	
gestión de procesos. concurrencia e interbloqueos	Tema 2	10	4	0,5
	Tema 3	9	5	1
gestión de memoria	Tema 4	21,5	6	2
gestión de archivos	Tema 5	10	5	1
gestión de entrada/salida	Tema 6	9		
examen final	Preparación	3		
Tutorías		1	1	1
Total:	102	72	24	6

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

7. BIBLIOGRAFÍA Y MATERIALES

7.1 BIBLIOGRAFÍA BÁSICA

Existe una gran variedad de libros que tratan los contenidos básicos de la asignatura. Los referentes son aquéllos que, ofreciendo un nivel de rigor y de detalle adecuados, realizan una exposición clara y didáctica de los contenidos:

Sistemas Operativos. A. Silberschatz y P.B Galván, 1999. Addison - Wesley, 5ª edición.

Los contenidos de este libro abarcan los siguientes temas: Introducción. Estructuras de los sistemas de computación. Estructuras de los sistemas operativos. Procesos. Planificación de la CPU. Sincronización de procesos. Bloqueos mutuos. Administración de memoria. Memoria virtual. Interfaz del sistema de archivos. Implementación del sistema de archivos. Sistemas de E/S. Estructura del almacenamiento secundario. Estructura del almacenamiento ternario. Estructuras de redes. Estructuras de los sistemas distribuidos. Sistemas de archivos distribuidos. Coordinación distribuida. Protección. Seguridad. El sistema UNIX. El sistema LINUX. Windows NT. Perspectiva histórica.

La primera parte del libro se corresponde con la asignatura y, más en concreto, se puede establecer las siguientes correspondencias entre los temas de Sistemas Operativos y los capítulos del libro. El tema I de Introducción corresponde a los capítulos 1 y 3. El tema 2 Gestión de Procesos a los capítulos 4 y 5. El tema

3 Interfaz del sistema al capítulo 3. El tema 4 de Concurrencia entre procesos al capítulo 5. El tema 5 de Interbloqueos al capítulo 7.

Se puede observar que constituye un libro clásico de sistemas operativos que ha tenido mucha aceptación a lo largo de su dilatada historia, como lo demuestra el hecho de que vaya por la quinta edición. La exposición es clara y concisa, las explicaciones son adecuadas y los ejemplos suficientes, lo que hace que sea uno de los libros recomendados tradicionalmente para la asignatura de iniciación de sistemas operativos en muchas universidades.

Con una organización tradicional, presenta los conceptos básicos sobre los que se construye un sistema operativo, aunque sin presentar uno concreto. Cada capítulo incluye ejercicios y bibliografía comentada en la que buscar información adicional.

En los capítulos avanzados, presenta los sistemas distribuidos y protección y seguridad.

En la parte dedicada a casos de estudio, expone con detalle los sistemas operativos más extendidos (UNIX, Linux y Windows NT), un capítulo para perspectiva histórica (Atlas, CTSS, MULTICS, OS/360, etc.).

Está publicada la sexta edición del libro, pero aún no ha sido traducida al castellano; no obstante, es muy recomendable para los alumnos que trabajen con sistemas operativos y no les cueste leer en inglés.

Sistemas Operativos. W. Stallings, 2005. Prentice Hall, 5ª edición.

Este libro contiene gran parte del temario de la asignatura. En concreto: Introducción a los sistemas informáticos. Introducción a los sistemas operativos. Descripción y control de procesos. Hilos, SMP y micronúcleos. Concurrencia: exclusión mutua y sincronización. Concurrencia: interbloqueo e inanición. Gestión de memoria. Memoria virtual. Planificación de monoprocesadores. Planificación de multiprocesadores y en tiempo real. Gestión de la E/S y planificación de discos. Gestión de archivos. Procesamiento distribuido, Cliente/Servidor y Agrupamientos. Gestión distribuida de procesos. Seguridad Informática. Apéndices: TCP/IP; Diseño orientado a objetos; Proyectos de sistemas operativos y programación; OSP: Un entorno para proyectos de sistemas operativos; BACI: El sistema de programación concurrente Ben-Ari.

Concretando las correspondencias entre la asignatura y los capítulos del libro, se puede establecer que el tema I de Introducción se corresponde al capítulo 2. El tema 2, Gestión de Procesos, a los capítulos 3 y 4. El tema 3, Interfaz del sistema, al artículo introductorio de este libro. El tema 4, de Concurrencia entre procesos, al capítulo 5. Y, finalmente, el tema 5 de Interbloqueos, al capítulo 6.

Es un libro extenso que trata ampliamente toda la parte básica y temas avanzados como procesamiento distribuido, arquitecturas cliente/servidor o seguridad, e incluye, en cada tema, apéndices con información ampliada relativa al

capítulo, lecturas recomendadas y problemas. Es de destacar el empeño que se pone en todos los capítulos de relacionar lo que se está viendo, con lo visto en los capítulos anteriores.

A pesar de que no presenta capítulos dedicados a sistemas operativos concretos, ni desarrolla un sistema operativo educativo en el que apoyarse para presentar los conceptos teóricos, a lo largo de todos los temas se incluyen secciones que ilustran la aplicación de los conceptos vistos a algunos sistemas operativos como Windows NT y diversas implementaciones de UNIX.

Al final del libro se incluyen varios apéndices en los que presenta distintos entornos y herramientas para el desarrollo de prácticas de laboratorio y un apéndice dedicado al diseño de sistemas operativos orientados a objeto.

Sistemas Operativos Modernos. A. S. Tanenbaum, 1993. Prentice Hall.

Se estudian los siguientes apartados: Introducción. Procesos. Administración de la memoria. Sistemas de archivo. Entrada/Salida. Bloqueos. Estudio 1: UNIX. Estudio 2: MS-DOS. Introducción a los sistemas distribuidos. Comunicaciones en los sistemas distribuidos. Sincronización en los sistemas distribuidos. Procesos y procesadores en sistemas distribuidos. Sistemas distribuidos de archivos. Estudio 3: Amoeba. Estudio 4: Mach.

El capítulo de la asignatura más tratado en el libro es el de Concurrencia entre procesos, puesto que dedica tres capítulos, en concreto el 2, 11 y 12, a la explicación de dicho tema.

El libro está estructurado en dos partes: la primera, titulada sistemas operativos tradicionales y la segunda con título sistemas operativos distribuidos. El contenido está planteado con un enfoque muy equilibrado entre pragmatismo y teoría, que junto con lo riguroso de la exposición, hacen al libro muy adecuado para la materia que trata. Sin embargo, esta traducción al castellano de la edición original inglesa de 1992 [Tanenbaum92] presenta una cantidad importante de erratas que exigen del lector un esfuerzo notable para evitar la adquisición de conceptos erróneos o de falsas interpretaciones. A pesar de todo, el libro se recomienda en clase, pero se facilita a los estudiantes una relación de las erratas. Recientemente, ha sido publicado el libro distributed operating systems [Tanenbaum95], que recoge fundamentalmente la segunda parte del texto del 93, al que se incorpora la revisión de dos sistemas no contemplados en dicha versión: Chorus y DCE.

Al final del libro se añade una completa bibliografía, ordenada por temas y alfabéticamente, y una breve introducción al lenguaje C.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Las referencias que se presentan a continuación están encaminadas a completar algunos apartados proporcionando un enfoque alternativo.

Introducción a UNIX. Un enfoque práctico. A. Afzal, 1997. Prentice Hall.

El libro está dedicado al sistema operativo UNIX con un enfoque práctico. Está escrito de una forma tutorial y, por tanto, es un libro enfocado a la docencia del entorno del sistema operativo UNIX en estados tempranos de aprendizaje, simple y fácil de leer.

Además de presentar una introducción general a los sistemas operativos y, más concretamente, al sistema operativo UNIX. También añade un capítulo introductorio del sistema de archivos de UNIX. El resto del libro lo dedica a presentar las utilidades más significativas del entorno de programación UNIX.

El libro entero cuenta con multitud de ejemplos sencillos, lo que lo convierte en un libro adecuado para una toma de contacto con el entorno y que no necesita tutorización, útil por tanto, en las primeras sesiones prácticas de toma de contacto con el entorno. Sin embargo, no presenta ningún capítulo dedicado a la programación de sistemas UNIX, lo que hace que pierda parte de su utilidad.

UNIX. Programación avanzada. F. M. Márquez, 1996. Rama.

Amplia y completa obra de referencia del entorno de programación UNIX, siendo de obligada lectura para todo aquel que no solo desee conocer el entorno de programación UNIX, sino que le interese descubrir la implementación interna de un sistema operativo real de amplia difusión, todo ello ilustrado profusamente con multitud de ejemplos.

Describe la interfaz de programación del sistema UNIX, no solo la interfaz de llamadas al sistema sino también muchas funciones proporcionadas por la librería C estándar. Cabe destacar la parte que dedica a los mecanismos IPC (semáforos, memoria compartida, etc.) por la clara y concisa explicación con multitud de ejemplos.

Linux. J. Tackett y D. Gunter, 2000. Prentice Hall, 4ª Edición.

Libro muy completo dedicado al sistema operativo Linux. Presenta su estructura, las particularidades de ser un sistema operativo de libre distribución y una guía de las distribuciones.

El grueso fundamental del libro detalla la instalación y administración del sistema y la red. Es un libro avanzado y eminentemente práctico, útil para aquellos que quieran tocar un sistema operativo.

Cada capítulo presenta al final una sección *Para más información...*, en la que establece referencias cruzadas con otros capítulos. Es un libro muy útil para tener una visión general del funcionamiento del sistema.

Sistemas Operativos: Diseño e Implementación. A. S. Tanenbaum, 1998. Prentice Hall.

Es un libro básico en lo que a la enseñanza de sistemas operativos se refiere.

La primera edición, publicada en 1987, revolucionó la docencia de esta materia al acompañar la presentación de los conceptos teóricos básicos de sistema operativo, con un sistema operativo educativo, MINIX, desarrollado por A. S. Tanenbaum, y del que se presenta el código fuente, ampliamente comentado, en la segunda mitad del libro.

A lo largo de la primera parte del libro, se hacen continuas referencias al código de MINIX, lo que permite al lector hacerse una idea precisa de los detalles de implementación de cada parte de un sistema operativo a través de una implementación concreta. Todos los capítulos finalizan con una serie de problemas de distinto grado de dificultad, que incluyen modificaciones en el código fuente de MINIX para llevar a la práctica los conceptos teóricos estudiados. Con el libro, viene la versión 2.0 de MINIX en CD-ROM que soporta los últimos microprocesadores Intel y también el protocolo de red TCP/IP.

7.3 OTROS RECURSOS

El alumno tiene a su disposición varios recursos para el aprendizaje de la asignatura que mostramos en los siguientes puntos:

- **Página web de la asignatura:** información completa y actualizada de la asignatura durante el curso académico. Informa sobre:
 8. Administración: profesores, periodización y criterios de evaluación.
 9. Clases teóricas: transparencias y enlaces de los temas.
 10. Clases prácticas: enunciados, software y seminarios.
 11. Trabajos: trabajos complementarios, enlaces interesantes.
 12. Estudiantes: ejercicios y exámenes.
 13. Novedades y anuncios.
- **Campus Virtual:** servicio Internet de complemento a la docencia y a la gestión académica y administrativa. El alumno puede encontrar material de trabajo, test de autoevaluación, calificaciones de exámenes y realizar consultas tutoriales con el profesor.
- **Direcciones de correo electrónico:** los profesores de la asignatura disponen de direcciones de correo electrónico donde se pueden mandar sugerencias y hacer consultas como mecanismo flexible e inmediato.
- **Transparencias y presentaciones multimedia:** elaboradas por los profesores de la asignatura con los contenidos que se desarrollan durante el curso.
- **Problemas y ejercicios:** conjunto de ejercicios y problemas resueltos y propuestos por los profesores de la asignatura.
- **Enlaces de interés:**
 14. <http://tunes.org/Review/OSes.html>: amplia colección de enlaces y direcciones de información relacionada con los sistemas operativos, tanto de sistemas de libre distribución como de sistemas comerciales.

15. <http://www.osdata.com/>: en esta página se realiza una extensa comparación de los sistemas operativos, mostrando abundante información acerca de sus características.
 16. <http://www.cs.arizona.edu/people/bridges/oses.html>: página con enlaces a proyectos de diseño de sistemas operativos específicos y generales. Permite contemplar parte de la investigación y avances que se producen en este campo así como diversa información de los centros que la realizan.
- **Artículos en revistas especializadas:**
 17. A fast file system for UNIX, M.K. McKusick, W. N. Joy, S. J. Leffler, R. S. Fabry, 1984. ACM Transactions on Computer Systems (TOCS).
 18. Analyzing the Multimedia Operating System, R. Steinmetz, 1995. IEEE Multimedia.
 19. Virtual Memory: Issues of Implementation, B. Jacob, T. Mudge, 1998. IEEE Computer.
 - **Grupos de Noticias (Netnews):**
 20. comp.arch
 21. comp.arch.arithmetic
 22. comp.parallel
 23. comp.sys.intel
 24. sci.electronics.components
 - **Redes Internacionales e Institutos**
 25. CESCO: Centro de Supercomputación de Cataluña
 26. CESGA: Centro de Supercomputación de Galicia
 27. CNM: Centro Nacional de Microelectrónica
 28. ITI: Instituto de Tecnología Informática (Universidad Politécnica de Valencia)
 29. IUTEPI: Instituto Universitario de Tecnología para la Informática
 30. I2IT: International Institute of Information Technology
 - **Asociaciones y Grupos de Interés**
 31. American Computer Science Association
 32. Asociación de Empresas de Electrónica, Tecnologías de la Información y Telecomunicaciones de España
 33. Asociación de Ingenieros en Informática
 34. International Association of Computer Information Systems
 35. International Association of Computer Investigative Specialists.
 36. European Association for Theoretical Computer Science
 - **Publicaciones Electrónicas**
 37. IR: Information research
 38. VJ: Virtual Journal of Nanoscale Science and Technology

39. The Technology Source
40. Issues in science and technology
- **Publicaciones Periódicas**
 41. Computer System Architecture
 42. IEEE Transactions on Circuits and Systems
 43. IEEE Transactions on Computers
 44. IEEE Transactions on VLSI
 45. IEEE Journal of Solid-State Circuits
 46. Information and Computation
 47. Integration, the VLSI journal
 48. Journal of the ACM
 49. Journal of Circuit systems and Computers
 50. Proceedings of the IEEE

GUÍA DOCENTE DE SISTEMAS OPERATIVOS I

Higinio Mora Mora; Francisco José Mora Gimeno

*Departamento de Tecnología Informática y Computación
Universidad de Alicante
(hmora; fjmora)@dtic.ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

A continuación se detallan las consideraciones específicas para la asignatura Sistemas Operativos 1I, en las que se define el marco conceptual concreto de la asignatura. Se tienen en cuenta los siguientes aspectos:

- Revisar el contenido de sistemas operativos en las recomendaciones de organismos internacionales de reconocido prestigio.
- Examinar los planteamientos de la asignatura en planes de estudio de otras universidades españolas.

La propuesta conjunta de ACM e IEEE, el *Computing Curricula* 2001 [CC2001] o el *Computing Curriculum-Computer Engineering* [CE2004] dedica un área de conocimiento completa al estudio de los Sistemas Operativos [OS], distribuyendo la materia en un conjunto de contenidos fundamentales y optativos. Los contenidos que se incluyen son los siguientes:

Sistemas operativos (18 horas fundamentales)

- SO1: Introducción a los sistemas operativos
- SO2: Principios de los sistemas operativos
- SO3: Concurrencia de procesos
- SO4: Planificación de procesos
- SO5: Gestión de memoria
- SO6: Gestión de dispositivos

- SO7: Seguridad y protección
- SO8: Sistemas de archivos
- SO9: Sistemas embebidos y de tiempo-real y
- SO10: Sistemas tolerantes a fallos
- SO11: Evaluación del rendimiento del sistema
- SO12: Encriptación

Un sistema operativo representa una abstracción del comportamiento del hardware y un gestor de los recursos del sistema. En esta área se contemplan los temas que influyen en el diseño de los sistemas operativos actuales. Muchas de las ideas implicadas en uso del sistema operativo tienen aplicabilidad más amplia a través del campo de la informática, tal como programación concurrente. Estudiar el diseño de los sistemas operativos tiene influencia en muchas otras áreas de la informática tales como la programación, diseño y puesta en práctica confiable, desarrollo moderno del dispositivo, ambientes virtuales, internet, sistemas seguro y muchos otras.

Se propone su estudio específico en cursos de nivel medio, ya sea de contenido exclusivo de sistemas operativos *Operating Systems* (CS225) o relacionados con otras materias: *Architecture and Operating Systems* (CS221), *Architectures for Networking and Communication* (CS222) y *Operationg Systems and Networking* (CS226). Estos conocimientos pueden ser ampliados mediante un conjunto de cursos avanzados.

Con relación a la docencia de la asignatura en la universidad española, destaca el carácter troncal de la materia en el primer ciclo con 9 créditos asignados, lo que pone de manifiesto su relevancia en el currículo del ingeniero en informática. El encaje de la materia en los planes de estudio de las universidades españolas se realiza en el primer ciclo y en cursos medios. De su análisis se extraen las siguientes conclusiones:

- La mayoría de las universidades ofrecen asignaturas básicas, avanzadas y un conjunto de asignaturas optativas afines a sistemas operativos.
- Respecto a los contenidos teóricos, no suele haber grandes diferencias, si acaso destacar el hecho de que profundizan más en determinados aspectos de la materia, motivados principalmente por un mayor número de créditos.

En la Universidad de Alicante, se han distribuido los contenidos en dos asignaturas troncales ubicadas en segundo y tercer curso: sistemas operativos I y sistemas operativos II. Los contenidos que se imparten en estas materias sirven constituyen uno de los pilares básicos sobre los que se sustenta la formación de los ingenieros en informática. Las aportaciones de los sistemas operativos a los diversos perfiles profesionales se relacionan en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de computadores. • Conocimientos y habilidades en el diseño de sistemas operativos. • Conocimientos básicos del funcionamiento de microprocesadores. • Conocimientos de computación concurrente.
Arquitectura y diseño de software	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de computadores. • Conocimientos y habilidades en el diseño de sistemas operativos. • Conocimientos básicos del funcionamiento de microprocesadores. • Conocimientos de computación concurrente.
Diseño multimedia	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de sistemas operativos. • Conocimientos básicos de microprocesadores y computación paralela. • Conocimientos y habilidades en el ámbito del multimedia, a nivel de procesamiento concurrente y administración de recursos.
Ingeniería de comunicación de datos	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Nociones fundamentales sobre comunicación digital y sistemas distribuidos.
Diseño de redes de comunicación	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Nociones fundamentales sobre comunicación digital y sistemas distribuidos
Asistencia técnica	<ul style="list-style-type: none"> • Conocimientos y habilidades en el mantenimiento de sistemas operativos. • Conocimientos sobre especificaciones del hardware de computadores y sus periféricos
Ingeniería de integración y pruebas e implantación y pruebas	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Habilidades de utilización de herramientas de simulación para validación de diseños. Realización de prototipos para pruebas.
Consultoría en empresas de TI	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos. • Formalización de especificaciones y requerimientos del usuario. • Conocimientos básicos sobre tecnología de computadores

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos de computación concurrente. • Conocimientos y habilidades en el diseño de sistemas operativos.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento de la metodología científica para la resolución de problemas. • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos básicos de Sistemas de tiempo real en la industria, robótica y fabricación, automatización de procesos industriales, tratamiento digital de la imagen. Enfoque industrial de los sistemas de control informático
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento de la metodología científica para la resolución de problemas. • Conocimientos y habilidades en el diseño de aplicaciones, arquitecturas y estructuras de dispositivos. • Conocimientos básicos de Sistemas de tiempo real en la industria, robótica y fabricación, automatización de procesos industriales, tratamiento digital de la imagen. Enfoque industrial de los sistemas de control informático. • Conocimientos y habilidades en el diseño de proyectos informáticos y aplicación de herramientas informáticas en la empresa.

Cuadro 1. *Adecuación de los tópicos relacionados con la asignatura Sistemas Operativos I*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

A continuación, se analiza el contexto local de la asignatura describiendo su relación con el resto de materias de la titulación, según el plan de estudios. La figura siguiente ilustra dicha relación:

IB: Informática Básica
SOI: Sistemas Operativos I
SOII: Sistemas Operativos II
SI: Sistemas Informáticos
SIND: Sistemas Industriales
STR: Sistemas de Tiempo Real
ARC: Administración e instalación de Redes de Computadores

FPI: Fundamentos de Programación I
FPII: Fundamentos de Programación II
SOR: Sistemas Operativos en Red
ESO: Estudio de un Sistema Operativo
PC: Programación Concurrente
ASI: Administración de Servicios de Internet
ASOTR: Arquitecturas y Sistemas Operativos para Tiempo Real

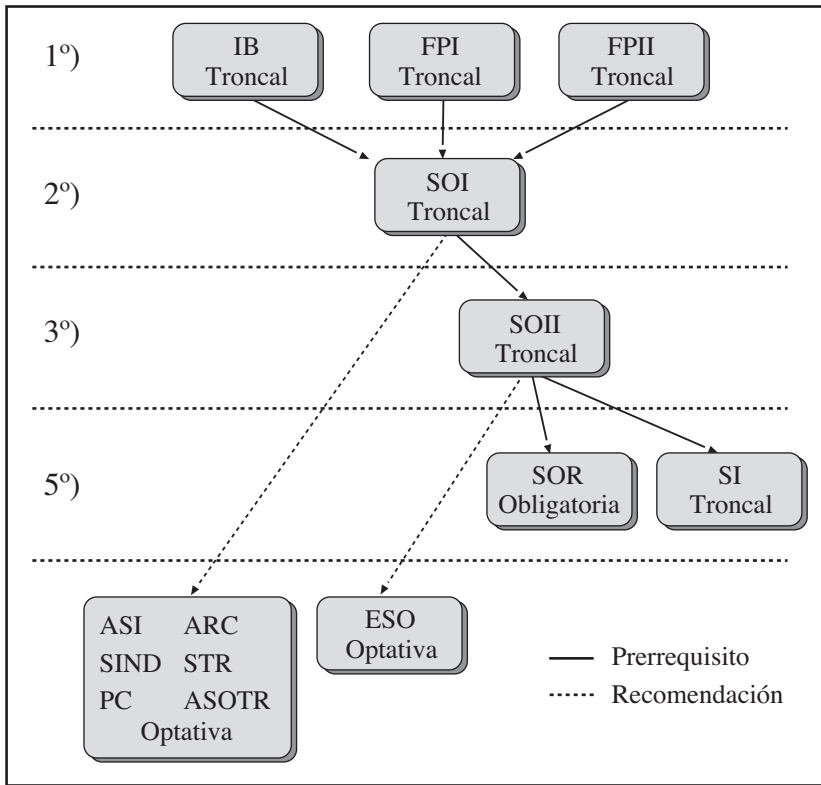


Ilustración 17. Relaciones de Sistemas Operativos I en el plan de estudios

Como se aprecia en la figura, el estudio de los sistemas operativos comienza en segundo curso, una vez que el alumno conoce parte de los conceptos previos necesarios, y se extiende hasta el quinto curso.

Sistemas Operativos I representa la introducción al estudio de los sistemas operativos y junto con Sistemas Operativos II, abarcan entre ambas el estudio de la totalidad los aspectos de un sistema operativo completo desde tanto un punto de vista conceptual como ingenieril. Respecto a su relación con el plan de estudio de 1993, se corresponde, junto con sistemas operativos II, con la asignatura de Fundamentos de los Sistemas Operativos (6 créditos) y Diseño de Sistemas Operativos (6 créditos).

Sistemas Operativos II se impartirá en el primer cuatrimestre del tercer curso. El hecho de impartir la asignatura en el primer cuatrimestre la acerca en el tiempo a Sistemas Operativos I, que se impartirá en el segundo cuatrimestre del segundo curso. También se habrá cursado, en el segundo curso, Estructuras de Computadores lo que posibilita que los alumnos tengan una base más sólida del

hardware del computador y puedan comprender mejor los sistemas operativos y su relación con el hardware.

La asignatura de Sistemas Operativos II se sustenta en los conocimientos adquiridos en Estructuras de Computadores sobre la organización interna y funcionamiento de los computadores y en los conocimientos adquiridos en Fundamentos de la Programación I y II, en cuanto a conceptos generales de programación en lenguajes de medio y alto nivel y por supuesto en la de Sistemas Operativos I de la que es continuación.

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro como objetivos comunes a todas las asignaturas, planteamos los siguientes objetivos para la asignatura de Sistemas Operativos I:

- OI1: Comprender nociones básicas del funcionamiento de los sistemas operativos así como las relaciones fundamentales entre los diferentes módulos que conforman la arquitectura de los sistemas operativos
- OI2: Conocer los conceptos y problemas básicos de la gestión de recursos de un sistema informático.
- OI3: Conocer la estructura lógica de un planificador de procesos y las diferentes alternativas o algoritmos de asignación de elementos de proceso.
- OI4: Comprender las características del interfaz del sistema operativo hacia el usuario y hacia el hardware del sistema.
- OI5: Aprender métodos de gestión de la concurrencia en la ejecución de procesos y de resolución de situaciones de conflicto en la asignación de recursos.
- OI6: Aplicar los conceptos teóricos y prácticos del diseño de sistema operativos y así captar su relación con ramas específicas como las Arquitecturas de Computadores, las Redes de Computadores y el resto de asignaturas de estudio de los sistemas operativos.

2.1.2 Objetivos interpersonales generales

Estos objetivos se corresponden con los objetivos interpersonales generales, comunes a todas las asignaturas de primer curso. Concretamente, estos son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

2.1.3 Objetivos sistémicos generales

Además de los objetivos cOS1, cOS2, cOS3 y cOS4 considerados en el artículo introductorio de este libro, incluimos los siguientes:

- OS1: Asumir los esquemas de teoría, abstracción y diseño como método que permite impulsar la capacidad para abstraer y generalizar los problemas, así como asimilar los rápidos avances en la disciplina y situarlos en su contexto de innovación científica y tecnológica.
- OS2: Desarrollar el espíritu crítico tanto para enfrentarse a un problema como para evaluar las ventajas e inconvenientes de un diseño concreto.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales

Dentro de las competencias instrumentales, distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Considerando que la asignatura *Sistemas Operativos I* contiene los siguientes temas:

- Tema 1: Introducción. Definiciones y conceptos
- Tema 2: Planificación de procesos
- Tema 3: Interfaz del sistema
- Tema 4: Sincronización y comunicación de procesos
- Tema 5: Interbloqueos

Desarrollamos los apartados anteriores para cada uno de ellos.

2.2.2 Habilidades cognitivas

Además de las habilidades comunes cCIC1 y cCIC2, consideradas en el artículo introductorio de este libro, incluimos las siguientes, agrupadas por temas de la asignatura.

Tema 1: Introducción. Definiciones y conceptos

- CIC1: Conocer los conceptos fundamentales de los sistemas operativos y adquirir una visión básica, al tiempo que global y homogénea entre todos los alumnos.
- CIC2: Concebir el estudio de los sistemas operativos como una rama de la informática y comprender y conocer sus interrelaciones con el resto de la disciplina.
- CIC3: Desarrollar la capacidad de abstracción de los alumnos mediante la exposición de los paradigmas característicos de teoría, abstracción y diseño, junto a los conceptos recurrentes de la disciplina informática.

Tema 2: Planificación de procesos

- CIC4: Comprender la problemática de la ejecución de procesos en sistemas multiprogramados y conocer los métodos de planificación de procesos más comunes.

- CIC5: Aprender las técnicas de simulación de estrategias de planificación y adquirir la capacidad de resolver las situaciones de conflicto.

Tema 3: Interfaz del sistema

- CIC6: Comprender el principio de separación entre especificación e implementación de funciones en un sistema informático y aplicarlo al diseño de sistemas operativos.
- CIC7: Asimilar los métodos de organización de las funciones de un sistema operativo y conocer como ejemplo el caso concreto de un sistema operativo real.
- CIC8: Aprender los métodos y la implementación de interrupciones y llamadas al sistema.

Tema 4: Sincronización y comunicación de procesos

- CIC9: Comprender la problemática de la gestión de recursos en un sistema informático y en especial el problema de la exclusión mutua.
- CIC10: Conocer los principales modelos y herramientas de resolución de problemas de comunicación y sincronización de procesos.

Tema 5: Interbloqueos

- CIC11: Comprender las causas y condiciones que provocan situaciones de interbloqueo en el acceso de a los recursos del sistema.
- CIC12: Comprender las técnicas y mecanismos de prevención y evitación de situaciones de interbloqueo.

2.2.3 Capacidades metodológicas

Además de las habilidades comunes cCIM1, cCIM2 y cCIM3, consideradas en el artículo introductorio de este libro, incluimos las siguientes agrupadas por temas de la asignatura.

Tema 1: Introducción. Definiciones y conceptos

- CIM1: Ser capaz de identificar los aspectos de teoría, abstracción y diseño característicos en el contexto concreto de los sistemas operativos.
- CIM2: Reconocer la arquitectura de un sistema informático y el tipo de sistema operativo que contiene.

Tema 2: Planificación de procesos

- CIM3: Ser capaz de identificar el modelo de planificación más adecuado para un sistema operativo y una determinada carga de trabajo.
- CIM4: Realizar simulaciones y diagramas de la ejecución de los procesos de un sistema, de acuerdo a una planificación.

Tema 3: Interfaz del sistema

- CIM5: Identificar los elementos de comunicación entre las capas de un sistema operativo.
- CIM6: Ser capaz de programar rutinas sencillas de aplicación que hagan uso de la interfaz del sistema.

Tema 4: Sincronización y comunicación de procesos

- CIM7: Familiarización con los modelos y herramientas de gestión de la concurrencia.
- CIM8: Adquirir destreza en la resolución de situaciones de sincronización y comunicación entre procesos.

Tema 5: Interbloques

- CIM9: Aprender a identificar una situación de interbloqueo.
- CIM10: Ser capaz de aplicar las estrategias de prevención, evitación y recuperación para soslayar situaciones de interbloqueo en un sistema informático.

Destrezas tecnológicas

Además de la destreza cCIT1, considerada en el artículo introductorio de este libro, incluimos aquí las siguientes destrezas tecnológicas desarrolladas para todos los bloques.

- CIT1: Ser capaz de manejar aplicaciones y programas. Familiarizarse con el manejo de ventanas, menús y componentes.
- CIT2: Usar con habilidad y destreza un sistema operativo real. Caso concreto de Linux.
- CIT3: Usar con destreza librerías de funciones y llamadas al sistema para programar sencillos programas de aplicación.
- CIT4: Adquirir destrezas en el diseño de módulos de un sistema operativo de acuerdo a unos determinados requerimientos.

Destrezas lingüísticas

Las destrezas lingüísticas abarcan las destrezas comunes cCIL1 y cCIL2, y las siguientes desarrolladas para todos los bloques.

- CIL1: Saber desarrollar cualquier tema, tanto de forma oral como escrita, en el que se tengan que aplicar conceptos en el ámbito de los sistemas operativos.
- CIL2: Saber desarrollar cualquier tema, tanto de forma oral como escrita, en el ámbito de la programación de rutinas en Linux, usando llamadas al sistema e interrupciones del sistema operativo.

2.2.4 Competencias interpersonales

Las competencias interpersonales se han dividido en: competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas

Quedan establecidas en las competencias comunes cCIPTC1 y cCIPTC2, enunciadas en el capítulo introductorio de este libro.

Compromiso con el trabajo

Las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.5 Competencias sistémicas

Integración de capacidades cognitivas, destrezas prácticas y disposiciones

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCs3, cCs4 y cCS5.

3. PRERREQUISITOS

3.1 COMPETENCIAS Y CONTENIDOS MÍNIMOS

Las competencias y contenidos mínimos adecuados para superar esta asignatura quedan definidas por los prerrequisitos oficiales que tiene en el plan de estudios de la titulación: Informática Básica, Fundamentos de Programación I y II.

Además de estos requisitos, es conveniente saber lo siguiente:

- Saber manejar un sistema operativo a nivel de usuario.
- Entender las diferencias entre teoría abstracción y diseño.
- Poseer una capacidad de abstracción adecuada.

3.2 PLAN DE TRABAJO Y ACTIVIDADES PARA LA CONSECUCCIÓN DE LOS PRERREQUISITOS

Los profesores de la asignatura recomendamos a todos los alumnos diversas lecturas para entender algunos aspectos fundamentales de los sistemas operativos, en especial aquellas que relacionan los conceptos adquiridos en primeros cursos con los sistemas operativos. Esta información la tienen disponible en el campus virtual y en la página web de la asignatura.

Además publicamos, al comienzo de cada curso, en el campus virtual, un test de autoevaluación (“test cero”) para que cada alumno pueda comprobar cuál es su nivel sobre la materia que va a estudiar. Según los resultados obtenidos, se le

recomendará una determinada trayectoria que le facilitará el aprendizaje de la asignatura para su puesta al día.

4. TEMAS DE CONTENIDO

4.1 TEMAS DE CONTENIDOS DE APRENDIZAJE

La asignatura se compone de los siguientes temas de contenido, como ya se ha mencionado anteriormente:

- Tema 1: Introducción. Definiciones y conceptos
- Tema 2: Procesos
- Tema 3: Interfaz del sistema
- Tema 4: Comunicación y sincronización de procesos
- Tema 5: Interbloqueos

4.2 DESARROLLO DE LOS TEMAS O UNIDADES DE CONTENIDO

Desarrollo del Tema 1: Introducción. Definiciones y Conceptos

Contenidos de teoría:

- 1.1 Concepto y objetivos del sistema operativo
- 1.2 Funciones del sistema operativo
- 1.3 Servicios del sistema operativo
- 1.4 Principio de diseño de los sistemas operativos
- 1.5 Evolución histórica de los sistemas operativos

Contenidos de práctica:

- 1.1 Introducción al entorno de prácticas
- 1.2 Introducción a UNIX

Desarrollo del Tema 2: Procesos

Contenidos de teoría:

- 2.1 Conceptos de procesos
- 2.2. Implementación de procesos
- 2.3. Hilos de ejecución
- 2.4. Planificación de procesos

Contenidos de práctica:

- 2.1. Creación de árboles de procesos en Linux
- 2.2. Manipulación de comandos fork, exit y wait.

Desarrollo del Tema 3: Interfaz del sistema

Contenidos de teoría:

- 3.1. Llamadas al sistema
- 3.2. Diseño de llamadas al sistema en UNIX

Contenidos de práctica:

- 3.1. Creación de rutinas y funciones con llamadas al sistema

Desarrollo del Tema 4: Comunicación y sincronización de procesos

Contenidos de teoría:

- 4.1. Conceptos de concurrencia
- 4.2. Coordinación y sincronización
- 4.3. Problemas clásicos y ejemplos de resolución
- 4.4. Comunicación
- 4.5. Caso de estudio

Contenidos de práctica:

- 4.1. Creación de ejercicios y rutinas de comunicación entre procesos
- 4.2. Tuberías, intercambio de señales y memoria compartida.

Desarrollo del Tema 5: Interbloqueos

Contenidos de teoría:

- 5.1. Especificación del problema
- 5.2. Prevención y evitación
- 5.3. Detección de interbloqueos y recuperación del sistema
- 5.4. Caso de estudio

Contenidos de práctica:

- 5.1. Análisis de situaciones de interbloqueos
- 5.2. Ejercicios de resolución de interbloqueos

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente que se ha desarrollado en el capítulo introductorio de este libro para las asignaturas de segundo curso de Ingeniería Informática, junto con la que aportamos en este apartado, conforman la de la asignatura de *Sistemas Operativos I*. Se presenta un modelo en el que además de los recursos o métodos tradicionales de enseñanza, como la lección magistral, introducimos otros para el mejor seguimiento de la misma, pensado que de esta manera el alumno se sentirá más motivado. Las actividades que se proponen son:

- Clases de teoría y ejercicios, en aulas.
- Clases de prácticas en laboratorios informáticos.
- Actividades en grupos pequeños/tutorías docentes.
- Trabajos complementarios.
- Tutorías de atención al alumno.

Clases de teoría y ejercicios

Objetivos generales

- Preparar al alumno para que sea un buen conocedor de la materia teórica propia de la asignatura que va a aprender y conocer sus aplicaciones en el campo que le interese.

- Despertar la motivación y el interés por la asignatura y desarrollar su curiosidad intelectual e interés por el rigor científico.
- Facilitar las orientaciones oportunas para el aprendizaje de cada materia.
- Favorecer la realización de ejercicios individuales que puedan llevar a una discusión colectiva. Se fomenta una reflexión cualitativa previa, deteniéndose en la clarificación de los conceptos.
- Facilitar la retroalimentación de conocimiento que fomente el aprendizaje constructivo.
- Desarrollar el espíritu crítico y una actitud abierta ante cambios de todo tipo y, en especial, a los científico-técnicos de su especialidad.
- Inculcar los principios deontológicos y adquirir conciencia del sentido humanístico de la ciencia y la técnica.

Metodología

- **Clase cero:** preparación que el alumno recibe para la siguiente clase. El profesor usará recursos web en donde publicará ejercicios y test de autoevaluación para “preevaluar” el tema teórico que se comenzará en la siguiente clase de teoría. Recomendación de libros, artículos, links de interés y apuntes o notas de relacionadas.
- **Lección magistral:** guión de la clase, objetivos y exposición de contenidos teóricos.
- **Incentivos para participar en clase:** preguntar si tienen conocimientos previos del tema que se va a impartir; dónde los han adquirido y lo que recuerden de ese aprendizaje. Hay que motivar al alumno para que participe en clase haciendo preguntas al profesor. Al ser una asignatura de segundo, la participación en clase por parte de los alumnos rompe con los esquemas de timidez de los alumnos de primer curso y recién llegados a la universidad.
- **Recursos y medios audiovisuales:** uso de pizarra y/o pizarra táctil, transparencias y retroproyector, diapositivas, modelos, cañón de proyección u ordenador (según las características del aula).
- **Comentar bibliografía recomendada:** indicar los libros, apuntes o enlaces web que son apropiados para el seguimiento del tema.
- **Material de apoyo:** proponer ejercicios para realizarlos en grupo, y cuestionarios de autoevaluación para hacerlos individualmente.
- **Opcionales:** motivar la realización de algún pequeño trabajo de investigación para reforzar el tema que acaban de aprender. Buscar aplicaciones. Pedir resumen del tema.

Clases de prácticas en laboratorios de informática

Objetivos generales

- Proporcionar experiencia y madurez en la aplicación de los principios des-

arrarrollados en teoría al diseño, prueba de software y hardware práctico, facilitando su comprensión y logrando un saber hacer en computación.

- Introducir los métodos experimentales y presentar correctamente los descubrimientos mediante la elaboración de informes.
- Integrar la actividad práctica con las lecciones teóricas definiendo proyectos de laboratorio con una secuenciación adecuada —introducción, resolución de problemas y diseño creativo—, una planificación cuidadosa y una buena sincronización con el desarrollo de la teoría.

Metodología

- **Clase cero:** a través del campus virtual, se informará tanto del contenido de las clases que se impartirán durante el curso, como de lo que deben preparar para llevar a cabo cada sesión (matriculación, material, asignación de profesores y horarios).
- **Sesión práctica:** presentación de los contenidos que se desarrollarán en esa clase. Se organiza de manera gradual la realización del proyecto de laboratorio a tres escalas:
- **Introducción.** En primer lugar se estudia y comprende un sistema o programa ya diseñado, familiarizando al alumno con herramientas básicas y consolidando conceptos. A continuación se observa el comportamiento derivado de modificaciones o ampliaciones del sistema
- **Resolución de problemas.** Se amplía la experiencia inicial y se desarrolla la autoconfianza en la habilidad de resolver problemas mediante la resolución de tareas de complejidad creciente y la propuesta de soluciones en entornos diversos.
- **Diseño creativo.** Refuerza la habilidad de resolución de problemas en contextos más complejos mediante la propuesta y evaluación crítica de soluciones, la toma de decisiones y el trabajo en equipo.
- **Recursos y medios audiovisuales:** se usará pizarra, proyector de transparencias y cañón de proyección u ordenador.
- **Comentar bibliografía recomendada:** indicar los libros o apuntes que son apropiados para el seguimiento de las clases.
- **Material de apoyo:** al final de cada bloque de prácticas, se repartirá un cuestionario de autoevaluación individual.
- **Opcionales:** proponer realizar algún trabajo de mayor envergadura que los propuestos en el enunciado. Se podrá hacer en grupos pequeños.

Actividades en grupos pequeños/Tutorías docentes

Objetivos generales

- Potenciar la actividad de trabajar en grupo (grupos pequeños), para desarrollar la relación del alumno con sus compañeros.

- Preparar al alumno para desarrollar temas de interés y motivar aprendizaje de investigación.

Actividades

- Hacer ejercicios y cuestionarios teóricos y prácticos.
- Hacer la lectura de libros, artículos o revistas recomendadas en la asignatura entre los miembros del grupo y después hacer comparaciones entre ellos.
- Realizar un trabajo de prácticas de mayor envergadura que la diseñada para las clases de prácticas, usando el ordenador.

Trabajos complementarios

Objetivos generales

- Aprender a trabajar de forma individual, haciendo un trabajo de investigación.

Actividades

- Hacer de forma voluntaria trabajos de índole particular que podrán tener tanto carácter teórico como aplicado.

Tutorías de atención al alumno

Objetivos generales

- Fomentar que el alumno acuda a tutorías para conseguir una relación profesor/alumno más personal.
- Realizar un seguimiento controlado del alumno por el profesor de su proceso de aprendizaje a lo largo del curso.

Actividades

- Hacer consultas relacionadas con la materia de teoría para dirigir el aprendizaje.
- Hacer consultas acerca de los trabajos propuestos en clase.
- Coordinar la realización de los ejercicios y trabajos en grupo.
- Hacer consultas dirigidas al seguimiento de sus estudios en Informática.

5.2 ESTRATEGIAS DE APRENDIZAJE

Es la forma en que el alumno debe prepararse para seguir cada uno de los puntos de la metodología empleada en el aprendizaje de la asignatura.

Estrategia para las clases teóricas

- El alumno debe leer las directrices marcadas en la clase cero.
- Obtener y leer el material que el profesor ha puesto a disposición del alumno para el desarrollo de cada clase.

- Una vez en clase, debe hacer un seguimiento de la misma, preguntado todo lo que sea necesario, en las pautas establecidas por el profesor.
- Después de clase se deben consultar apuntes, libros o enlaces web recomendados
- Para cualquier duda, acudir a tutorías y /o consultar con los compañeros.

Estrategia para las clases de prácticas en laboratorio

- El alumno debe leer las directrices marcadas en la clase cero.
- Llevar material para la elaboración de las prácticas: dispositivos de almacenamiento, enunciado del trabajo a realizar y apuntes del tema.
- Una vez en clase, debe hacer un seguimiento de la misma preguntado todo lo que sea necesario, en las pautas establecidas por el profesor.
- El profesor dirigirá en todo momento al alumno para la realización de las prácticas dándole consejos y estrategias.
- Cada alumno debe analizar el grado de conocimiento acerca de la práctica que está realizando para superar la evaluación individual de la misma.

Estrategia para las actividades de los grupos pequeños/tutorías docentes

- El profesor repartirá trabajos y ejercicios para realizarlos en grupos pequeños.
- Recomendará al grupo la realización de trabajos relacionados con algún tema de la materia.
- El profesor establecerá unas determinadas pautas de trabajo.
- Se hace un seguimiento del trabajo.

Estrategia para hacer trabajos complementarios

- El profesor repartirá trabajos y ejercicios para realizarlos de forma individual o colectiva y los recomendará de acuerdo a las pretensiones de cada alumno.
- El profesor establecerá unas determinadas pautas de trabajo.
- En horas de tutorías, se hará el seguimiento y la coordinación de los trabajos complementarios.

Estrategia para hacer una autoevaluación

La autoevaluación va dirigida a que el alumno sepa en todo momento su grado de aprendizaje de la asignatura, lo primero porque le interesa saber si va a ir bien preparado al examen (lo que más le interesa), y lo segundo porque aprenderá conceptos que le servirán para el aprendizaje de otras materias. Por ello proponemos:

- Realización de ejercicios básicos al finalizar cada tema para afianzar los conocimientos adquiridos.

- Realización de ejercicios más complejos con la intención de preparar al alumno para que vaya tomando conciencia de las dificultades que se puede encontrar en los exámenes.
- Realización de pruebas de autoevaluación.

Si lo cree conveniente, puede volver a repetir las pruebas tantas veces como quiera.

Estrategia para hacer una evaluación final

Constituye el instrumento docente de validación de los demás mecanismos que intervienen en el proceso de enseñanza-aprendizaje. Es el proceso, sin duda complejo, que comprende la obtención de información útil acerca del grado de asimilación de los conceptos, el nivel de adiestramiento conseguido en lo aplicado y la adecuación de la actitud hacia los objetivos. Este proceso permitirá obtener valoraciones, a partir de las cuales tomar decisiones sobre la conveniencia de reorientación del proceso educativo.

Si el alumno ha llevado un seguimiento de la asignatura marcado por las pautas indicadas por los profesores, se supone que debe ir bien preparado para la evaluación final, que consistirá en un examen teórico. Por otra parte, la evaluación de las clases prácticas de forma continuada hace que el alumno siga una estrategia para trabajar en cada clase de prácticas y así, al finalizar las mismas, tiene su evaluación final, casi siempre satisfactoria.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

Para obtener el número de créditos necesarios para completar la asignatura, cada alumno debe realizar un plan de trabajo que está dividido en actividades presenciales y no presenciales. Las horas que debe dedicar a las actividades presenciales se reparten entre las clases de teoría, de prácticas en laboratorio y las actividades en grupos pequeños/tutorías docentes. Para las no presenciales, el tiempo dedicado a las clases teóricas está repartido entre la clase cero y el test de autoevaluación que se hace después de cada tema; para las actividades en grupos pequeños se hace un reparto de tiempo dedicado a la lectura y comprensión de los ejercicios propuestos y corregidos por el profesor y entregados a cada grupo. Las horas dedicadas a la preparación del examen de teoría y las de tutorías completan las que debe realizar el alumno para sacar adelante la asignatura.

En las tablas siguientes se propone el número de horas que debe dedicar cada alumno para el conocimiento de la asignatura.

HORAS DE ACTIVIDADES PRESENCIALES			
Actividad	Clases de Teoría	Clases de prácticas	Actividades en Grupos Pequeños
Presentación	1,5		1,5
Tema 1	1,5	1,5	3
Tema 2	6	6	6
Tema 3	3	3	1,5
Tema 4	6	3	3
Tema 5	3	3	3
Preparación examen final	1,5		3
Duración examen final	3		
Total: 63	16,5	21	16

Cuadro 2. Plan de trabajo de la asignatura, número de horas presenciales

HORAS DE ACTIVIDADES NO PRESENCIALES			
Actividad	Estudio de la asignatura	Realización de las prácticas fuera del horario de la asignatura	Desarrollo de las actividades en grupos pequeños / tutorías docentes
Presentación	3		0
Tema 1	3	3	0,5
Tema 2	6	9	1
Tema 3	3	3	0,5
Tema 4	6	6	1,5
Tema 5	3	3	0,5
Preparación examen final	6		
Total: 59	30	24	5

Cuadro 3. Plan de trabajo de la asignatura, número de horas no presenciales

7. BIBLIOGRAFÍA Y MATERIALES

7.1 BIBLIOGRAFÍA BÁSICA

Existe una gran variedad de libros que tratan los contenidos básicos de la asignatura. Los referentes son aquéllos que, ofreciendo un nivel de rigor y de detalle adecuados, realizan una exposición clara y didáctica de los contenidos. :

Sistemas Operativos. A. Silberschatz y P.B Galván, 1999. Addison - Wesley, 5ª edición.

Los contenidos de este libro abarcan los siguientes temas: Introducción. Estructuras de los sistemas de computación. Estructuras de los sistemas operati-

vos. Procesos. Planificación de la CPU. Sincronización de procesos. Bloqueos mutuos. Administración de memoria. Memoria virtual. Interfaz del sistema de archivos. Implementación del sistema de archivos. Sistemas de E/S. Estructura del almacenamiento secundario. Estructura del almacenamiento ternario. Estructuras de redes. Estructuras de los sistemas distribuidos. Sistemas de archivos distribuidos. Coordinación distribuida. Protección. Seguridad. El sistema UNIX. El sistema LINUX. Windows NT. Perspectiva histórica.

Este libro constituye un clásico de sistemas operativos y ha tenido mucha aceptación a lo largo de su dilatada historia, como lo demuestra el hecho de que vaya por la quinta edición. La exposición es clara y concisa, las explicaciones son adecuadas y los ejemplos suficientes, lo que hace que sea uno de los libros recomendados tradicionalmente para la asignatura de iniciación de sistemas operativos en muchas universidades.

Sistemas Operativos. W. Stallings, 2005. Prentice Hall, 5ª edición.

Este libro contiene gran parte del temario de la asignatura; en concreto: Introducción a los sistemas informáticos. Introducción a los sistemas operativos. Descripción y control de procesos. Hilos, SMP y micronúcleos. Concurrencia: exclusión mutua y sincronización. Concurrencia: interbloqueo e inanición. Gestión de memoria. Memoria virtual. Planificación de monoprocesadores. Planificación de multiprocesadores y en tiempo real. Gestión de la E/S y planificación de discos. Gestión de archivos. Procesamiento distribuido, Cliente/Servidor y Agrupamientos. Gestión distribuida de procesos. Seguridad Informática. Apéndices: TCP/IP; Diseño orientado a objetos; Proyectos de sistemas operativos y programación; OSP: Un entorno para proyectos de sistemas operativos; BACI: El sistema de programación concurrente Ben-Ari.

Es un libro extenso que trata ampliamente toda la parte básica y temas avanzados como procesamiento distribuido, arquitecturas cliente/servidor o seguridad, e incluye, en cada tema, apéndices con información ampliada relativa al capítulo, lecturas recomendadas y problemas. Es de destacar el empeño que se pone en todos los capítulos de relacionar lo que se está viendo con lo visto en los capítulos anteriores.

Sistemas Operativos Modernos. A. S. Tanenbaum, 1993. Prentice Hall.

El libro está estructurado en dos partes: la primera titulada *sistemas operativos tradicionales* y la segunda *sistemas operativos distribuidos*. El contenido está planteado con un enfoque muy equilibrado entre pragmatismo y teoría, que junto con lo riguroso de la exposición, hacen al libro muy adecuado para la materia que trata. Sin embargo, esta traducción al castellano de la edición original inglesa de 1992 presenta una cantidad importante de erratas que exigen del lector un esfuerzo notable para evitar la adquisición de conceptos erróneos o de

falsas interpretaciones. A pesar de todo, el libro se recomienda en clase pero se facilita a los estudiantes una relación de las erratas. Recientemente, ha sido publicado el libro *distributed operating systems*, que recoge fundamentalmente la segunda parte del texto del 93, al que se incorpora la revisión de dos sistemas no contemplados en dicha versión: Chorus y DCE.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

Las referencias que se presentan a continuación están encaminadas a completar algunos apartados proporcionando un enfoque alternativo.

Introducción a UNIX. Un enfoque práctico. A. Afzal, 1997. Prentice Hall.

El libro está dedicado al sistema operativo UNIX con un enfoque práctico. Está escrito de una forma tutorial y, por tanto, es un libro enfocado a la docencia del entorno del sistema operativo UNIX en estados tempranos de aprendizaje, simple y fácil de leer. Cuenta con multitud de ejemplos sencillos, lo que lo convierte en un libro adecuado para una toma de contacto con el entorno y que no necesita tutorización, útil por tanto, en las primeras sesiones prácticas de toma de contacto con el entorno. Sin embargo, no presenta ningún capítulo dedicado a la programación de sistemas UNIX, lo que hace que pierda parte de su utilidad.

UNIX. Programación avanzada. F. M. Márquez, 1996. Rama.

Amplia y completa obra de referencia del entorno de programación UNIX, siendo de obligada lectura para todo aquel que no solo desee conocer el entorno de programación UNIX, sino que le interese descubrir la implementación interna de un sistema operativo real de amplia difusión, todo ello ilustrado profusamente con multitud de ejemplos.

Describe la interfaz de programación del sistema UNIX, no solo la interfaz de llamadas al sistema, sino también muchas funciones proporcionadas por la librería C estándar. Cabe destacar la parte que dedica a los mecanismos IPC (semáforos, memoria compartida, etc.) por la clara y concisa explicación con multitud de ejemplos.

Linux. J. Tackett y D. Gunter, 2000. Prentice Hall, 4ª Edición.

Libro muy completo dedicado al sistema operativo Linux. Presenta su estructura, las particularidades de ser un sistema operativo de libre distribución y una guía de las distribuciones.

Cada capítulo presenta al final una sección *Para más información ...* en la que establece referencias cruzadas con otros capítulos. Es un libro muy útil para tener una visión general del funcionamiento del sistema.

7.3 OTROS RECURSOS

El alumno tiene a su disposición varios recursos para el aprendizaje de la asignatura que mostramos en los siguientes puntos:

- **Página web de la asignatura:** información completa y actualizada de la asignatura durante el curso académico. Informa sobre:
 1. Administración: profesores, periodización y criterios de evaluación.
 2. Clases teóricas: transparencias y enlaces de los temas.
 3. Clases prácticas: enunciados, software y seminarios.
 4. Trabajos: trabajos complementarios, enlaces interesantes.
 5. Estudiantes: ejercicios y exámenes.
 6. Novedades y anuncios.
- **Campus Virtual:** servicio Internet de complemento a la docencia y a la gestión académica y administrativa. El alumno puede encontrar material de trabajo, test de autoevaluación, calificaciones de exámenes y realizar consultas tutoriales con el profesor.
- **Direcciones de correo electrónico:** los profesores de la asignatura disponen de direcciones de correo electrónico donde se pueden mandar sugerencias y hacer consultas como mecanismo flexible e inmediato.
- **Transparencias y presentaciones multimedia:** elaboradas por los profesores de la asignatura con los contenidos que se desarrollan durante el curso.
- **Problemas y ejercicios:** conjunto de ejercicios y problemas resueltos y propuestos por los profesores de la asignatura.
- **Enlaces de interés:**
 - <http://tunes.org/Review/OSes.html>: amplia colección de enlaces y direcciones de información relacionada con los sistemas operativos, tanto de sistemas de libre distribución como de sistemas comerciales.
 - <http://www.osdata.com/>: en esta página se realiza una extensa comparación de los sistemas operativos, mostrando abundante información acerca de sus características.
 - <http://www.cs.arizona.edu/people/bridges/oses.html>: página con enlaces a proyectos de diseño de sistemas operativos específicos y generales. Permite contemplar parte de la investigación y avances que se producen en este campo así como diversa información de los centros que la realizan.
- **Artículos en revistas especializadas:**
 - A fast file system for UNIX, M.K. McKusick, W. N. Joy, S. J. Leffler, R. S. Fabry, 1984. ACM Transactions on Computer Systems (TOCS).
 - Analyzing the Multimedia Operating System, R. Steinmetz, 1995. IEEE Multimedia.

- Virtual Memory: Issues of Implementation, B. Jacob, T. Mudge, 1998. IEEE Computer.
- **Grupos de Noticias (Netnews):**
 - comp.arch
 - comp.arch.arithmetic
 - comp.parallel
 - comp.sys.intel
 - sci.electronics.components
- **Redes Internacionales e Institutos**
 - CESCA: Centro de Supercomputación de Cataluña
 - CESGA: Centro de Supercomputación de Galicia
 - CNM: Centro Nacional de Microelectrónica
 - ITI: Instituto de Tecnología Informática (Universidad Politécnica de Valencia)
 - IUTEPI: Instituto Universitario de Tecnología para la Informática
 - I2IT: International Institute of Information Technology
- **Asociaciones y Grupos de Interés**
 - American Computer Science Association
 - Asociación de Empresas de Electrónica, Tecnologías de la Información y Telecomunicaciones de España
 - Asociación de Ingenieros en Informática
 - International Association of Computer Information Systems
 - International Association of Computer Investigative Specialists.
 - European Association for Theoretical Computer Science
- **Publicaciones Electrónicas**
 - IR: Information research
 - VJ: Virtual Journal of Nanoscale Science and Technology
 - The Technology Source
 - Issues in science and technology
- **Publicaciones Periódicas**
 - Computer System Architecture
 - IEEE Transactions on Circuits and Systems
 - IEEE Transactions on Computers
 - IEEE Transactions on VLSI
 - IEEE Journal of Solid-State Circuits
 - Information and Computation
 - Integration, the VLSI journal
 - Journal of the ACM
 - Journal of Circuit systems and Computers
 - Proceedings of the IEEE

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DEL APRENDIZAJE

8.1 SISTEMAS DE EVALUACIÓN

La evaluación se plantea como un proceso de comprobación del grado de consecución de los objetivos planteados. En este sentido, se propone una evaluación discretizada a lo largo de todo el cuatrimestre, que ofrezca al alumno una mayor flexibilidad y al mismo tiempo muestre su evolución y madurez.

La distinta naturaleza de los conceptos a evaluar necesita de diversos métodos de valoración. Por consiguiente, se proponen dos tipos de pruebas:

- Evaluación de las clases de teoría: prueba escrita compuesta de cuestiones cortas teóricas y de problemas de especificación y desarrollo donde se apliquen los conceptos básicos.
- Claridad en la descripción de los conceptos teóricos exigidos.
- Forma en que se plantea el ejercicio que se debe desarrollar.
- Forma en que se realiza el ejercicio.
- Explicación de la conclusión a las que se llega al finalizarlo.
- La nota obtenida tiene un peso en la asignatura del 50% de la nota final.
- Evaluación de las prácticas en laboratorio: la calificación de los trabajos prácticos se realiza en función del grado de satisfacción de los requerimientos planteados en cada ejercicio y de la capacidad de abstracción y diseño demostrada en la resolución de los mismos. La valoración de este apartado se calcula mediante la media ponderada de las calificaciones de las prácticas de laboratorio realizadas durante todo el curso. El índice de ponderación (w_i) es directamente proporcional a la complejidad estimada para cada trabajo, establecido al comienzo del cuatrimestre:

$$P = \sum_i w_i \cdot P_i$$

w_i Ponderación de la práctica i

P_i Resultado práctica i

Ilustración 18. *cálculo de la nota de prácticas*

- Se propone, igualmente, una prueba global de prácticas al final del curso para aquellos alumnos que deseen mejorar su calificación y/o no superen la evaluación progresiva de las prácticas.

La nota obtenida tiene un peso en la asignatura del 50% de la nota final.

- Evaluación de trabajos en grupos pequeños/tutorías docentes: presentación al profesor y a los compañeros de clase del trabajo realizado por un grupo (pequeño) de alumnos. Se valora:

- Tema elegido y desarrollo del mismo.
- Plan de trabajo para cada componente.
- Grado de relación con la asignatura.
- Presentación, claridad y exposición.
- Bibliografía y recursos utilizados en su elaboración.
- Campos de aplicación.

La nota obtenida se compensará con la nota del examen o bien sustituirá a ésta (a criterio del profesor).

- Evaluación de trabajos complementarios: presentación al profesor de trabajos voluntarios individuales. Se valora:
 - Tema elegido y desarrollo del mismo.
 - Grado de relación con la materia.
 - Presentación y claridad.
 - Comentarios, bibliografía y recursos utilizados en su elaboración.

La nota obtenida se suma a la nota final.

- Evaluación de tutorías de atención al alumno: asistencia al despacho del profesor para consultar dudas y revisión de trabajos. Las pautas de asistencia a tutorías estarán, para cada alumno, en función de lo que se haya implicado en el trabajo de aprendizaje de la asignatura. Se valora:
 - Interés por preguntar acerca de lo que no se entiende en clase.
 - Interés por seguir la asignatura mediante la relación profesor/alumno.
 - En la nota final se tendrá en cuenta la asistencia a tutorías.

8.2 CRITERIOS DE EVALUACIÓN

Dependiendo de los puntos obtenidos en los apartados evaluables, la calificación se obtiene de acuerdo con las siguientes pautas:

Calificación	Rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • Si alcanza todos los puntos de cada apartado. Debe dominar en profundidad la parte teórica y práctica y haber realizado trabajos en grupo e individuales. Además debe haber participado en las clases y asistido a tutorías (presenciales o virtuales).
Notable	7,0-8,9	<ul style="list-style-type: none"> • Si alcanza la mitad de los puntos de cada apartado. Debe dominar satisfactoriamente la parte teórica y práctica y haber realizado trabajos en grupo e individuales. Además debe haber participado en las clases y asistido a tutorías (presenciales o virtuales).
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • Si alcanza algún punto de cada apartado. Debe dominar de forma básica la parte teórica y práctica. Debe haber participado en clase, aunque no satisfactoriamente, y debe haber realizado alguna consulta en tutorías.
Suspenso	0,0-4,9	<ul style="list-style-type: none"> • Si no alcanza ningún punto de cada apartado. Su actitud es pasiva tanto en la parte teórica como práctica. Sin haber realizado ninguna consulta en tutorías.

Cuadro 4. *Criterios de evaluación*

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En las siguientes tablas, presentamos el análisis de coherencia de la guía docente de Sistemas Operativos I. Como se puede comprobar, se han relacionado los objetivos y competencias con los temas, el plan de trabajo propuesto para el alumnado y los criterios de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales			
OI1	CIC1-12 CIM 1-10 CIT 1-4	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/revisión de libros/realización de ejercicios y trabajos propuestos). Tutorías individualizadas y organizadas.	Procedimientos: Examen. Prácticas. Actividades en grupo. Ejercicios de autoevaluación. Criterios: Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al lenguaje de la lógica de primer orden.
OI2	CIC 1-12 CIM 1-10 CIT 1-4	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y trabajos propuestos). Tutorías individualizadas y organizadas.	Procedimientos: Examen. Prácticas. Actividades en grupo. Ejercicios de autoevaluación. Criterios: Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos al lenguaje de la lógica de primer orden.
OI3	CIC 4-5 CIM 3-4 CIT 1-4	2	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/revisión de libros/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas	Procedimientos: Examen. Prácticas. Actividades en grupo. Ejercicios de autoevaluación. Criterios: Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a la teoría semántica.
OI4	CIC 9-10 CIM 5-6 CIT 1-4	3	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/revisión de libros/realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	Procedimientos: Examen. Actividades en grupo. Ejercicios de autoevaluación. Criterios: Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.

Cuadro 5. *Análisis de coherencia: competencias instrumentales*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales			
OI5	CIC 4-5; 9-12 CIM 3-4; 7-10 CIT 1-4	2,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.
OI6	CIC 4-5 CIM 3-4 CIT 1-4	1,2,3,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.
cOI1	CIC 2-12 CIM 3-10 CIT 1-4	1,2,3,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.

Cuadro 6. Análisis de coherencia: competencias instrumentales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales			
cOI2	CIC 2-12 CIM 3-10 CIT 1-4	1,2,3,4,5	Enseñanza presencial (trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.
cOI3	CIL1, CIL2 cCIL1	1,2,3,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.
cOI4	CIL1, CIL2 cCIL1	1,2,3,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.
cOI5	CIM 1-10 CIL1, CIL2	1,2,3,4,5	Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.

Cuadro 7. *Análisis de coherencia: competencias instrumentales*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales			
cOI6	CIC 1-12 CIM 1-2	1,2,3,4,5	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Actividades en grupo. Ejercicios de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos y métodos relativos a los sistemas de deducción.

Cuadro 8. *Análisis de coherencia: competencias instrumentales*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales			
cOIP1	CIPTC1 CIPTC2 De cCIPTR1 a cCIPTR4	2,3,4,5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y trabajos propuestos). Tutorías organizadas. Memoria de los trabajos propuestos.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3	2,3,4,5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y trabajos propuestos). Tutorías organizadas. Memoria de los trabajos propuestos.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.

Cuadro 9. *Análisis de coherencia: competencias interpersonales*

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales			
cOIP3	cCIPTR4	2,3,4,5	Enseñanza presencial (trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (realización de ejercicios y trabajos propuestos). Tutorías organizadas. Memoria de los trabajos propuestos.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos en grupo.

Cuadro 10. Análisis de coherencia: competencias interpersonales

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas			
OS1	cCS1 cCS2 cCS4 cCS5	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Examen. Prácticas. Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Nivel de actuación en destrezas transferibles
OS2	cCS1 cCS2 cCS3 cCS4 cCS5	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos relacionados con los sistemas de demostración.
cOS1	cCS1 cCS4 cCS5	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos relacionados con los sistemas de demostración.

Cuadro 11. Análisis de coherencia: competencias sistémicas

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	TEMAS	PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	ISistémicas			
cOS2	cCS2 cCS3	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos relacionados con los sistemas de demostración.
cOS3	cCS1 cCS2 cCS4 cCS5	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos relacionados con los sistemas de demostración.
cOS4	cCS2	1,2,3,4,5	Enseñanza presencial (lección magistral /Trabajo de aula en grupos/ prácticas de laboratorio). Enseñanza no presencial (aprendizaje on-line/ revisión de libros/realización de ejercicios y prácticas propuestas). Tutorías individualizadas y organizadas.	<u>Procedimientos:</u> Actividades en grupo. Ejercicios de autoevaluación. Trabajos complementarios. <u>Criterios:</u> Grado de análisis y evaluación de los procedimientos relacionados con los sistemas de demostración.

Cuadro 12. *Análisis de coherencia: competencias sistémicas*

GUÍA DOCENTE DE RELACIONES JURÍDICAS BÁSICAS

M.ª Teresa Cantó López

*Departamento de Estudios Jurídicos del Estado
Universidad de Alicante
teresa.canto@ua.es*

1. CONTEXTUALIZACIÓN

1.1 PERFIL DE LOS CRÉDITOS DE LA MATERIA. ADECUACIÓN AL PERFIL PROFESIONAL Y ACADÉMICO DE LA TITULACIÓN

En el marco del EEES, la titulación en Informática debe formar ingenieros que estén preparados para integrarse en las empresas del sector con conocimientos amplios en la implantación de Nuevas Tecnologías. El Libro Blanco del Título de Grado en Ingeniería Informática establece entre varias funciones propias de un técnico en esta materia la de consultoría informática, dirección de proyectos informáticos y centros de programación de datos; al tiempo que distingue como campos profesionalizados, la seguridad y las consultoras informáticas.

En la elaboración de la presente guía docente hay que partir de las distintas recomendaciones curriculares de informática recogidas en el “Computing curricula 2001” realizado conjuntamente por la ACM (Association for Computing Machinery) [ACM01b], y el IEEE (Institute for Electrical and Electronic Engineers). Conforme a este análisis, las actuales titulaciones de Informática están relacionadas principalmente con tres perfiles: perfil profesional de desarrollo del software, perfil profesional de sistemas, y perfil profesional de gestión y tecnologías de la información. En todos ellos se considera imprescindible la verificación de cualquier sistema informático formado por diversos productos como: software, redes, aplicaciones..., así como su puesta en el mercado para lograr la satisfacción de los potenciales clientes.

En este contexto, la asignatura de Relaciones Jurídicas Básicas cumple una función horizontal o transversal a dichos perfiles, y en particular aporta los ins-

trumentos necesarios para desarrollar las competencias de auditoria y peritaje. Proporciona al profesional como Ingeniero informático los elementos de conocimiento para llevar a cabo las competencias establecidas en el citado Libro Blanco, pues sus contenidos se dedican al estudio de la normativa vigente en materia de nuevas tecnologías. En consecuencia, esta materia cumple un papel fundamental en el desarrollo de las competencias específicas del auditor o perito tendentes a asegurar la conformidad del sistema informático a la legislación en vigor y a facilitar la tarea de redactar informes válidos ante las Administraciones públicas, Tribunales de Justicia y corporaciones oficiales en los asuntos relacionados con la materia informática.

En otro orden de cosas, y siguiendo necesariamente las directrices generales propias de las titulaciones de Informática (Reales decretos 1459/1990 1460/1990, 1461/1990 de 26 de octubre, BOE 1990), en la Universidad de Alicante, y en los planes de estudio de Informática, la asignatura Relaciones Jurídicas Básicas aparece como materia obligatoria en la titulación de Ingeniería Técnica en Informática de Gestión y como optativa en las titulaciones de ingeniería en informática e Ingeniería Técnica en Informática de Sistemas. En las tres titulaciones tiene asociados 4,5 créditos teóricos, cuyos contenidos son: conceptos jurídicos elementales, nociones de derecho privado, y aspectos jurídicos civiles y mercantiles (propiedad intelectual y propiedad industrial).

El desarrollo de la técnica informática y la incorporación de las tecnologías de la información y las comunicaciones (TIC) en las relaciones sociales y económicas suscitan determinados conflictos a resolver por el derecho. Como consecuencia de ello, el derecho de la informática es una disciplina en evolución constante, que permite regular jurídicamente la protección del software, bases de datos y productos multimedia, además de la contratación de bienes y servicios informáticos, la protección de los datos de carácter personal, la seguridad informática, internet, o el comercio electrónico, entre otros aspectos.

Por ello, el análisis de la adecuación del perfil de la asignatura a los perfiles de la titulación debe orientarse hacia el conocimiento y habilidades de los instrumentos jurídicos necesarios para adecuar el sistema informático de que se trate a la legislación relacionada con la informática y las TIC. Además, los descriptores de la asignatura reflejan la variedad de normas penales, civiles, mercantiles, administrativas que tienen por objeto proteger jurídicamente los aspectos relacionados con la misma, de manera que garantiza la formación básica de un informático para resolver con éxito las distintas situaciones en su ámbito profesional de acuerdo con su propio perfil, por ejemplo: desarrollar el software con respeto a los derechos de propiedad intelectual, seguridad informática, realización de comunicaciones electrónicas o la realización de la contratación informática.

En síntesis, de lo dispuesto hasta aquí, se deduce que los tópicos relacionados con esta asignatura van a ser útiles en la formación de cualquier ingeniero informático pues se trata de una materia que le va a proporcionar unas competencias específicas transversales en su desarrollo profesional, como se verá más adelante. Pero, en este momento conviene significar la función de esta asignatura en los siguientes perfiles, englobados a su vez en tres perfiles profesionales generales.

Perfil profesional de desarrollo de software:

- Desarrollo de software y aplicaciones: para el diseño de software. Es esencial saber el ámbito a que se extiende la protección jurídica del software, bases de datos, y sus componentes, internet y comercio electrónico.
- Arquitectura y diseño de software: para el diseño de software es esencial saber el ámbito a que se extiende la protección jurídica del software, bases de datos, y sus componentes, así como la normativa de telecomunicaciones.
- Diseño multimedia: el conocimiento de las normas de propiedad intelectual proporcionan los instrumentos para la protección jurídica de los productos multimedia, software e internet.

Perfil profesional de sistemas:

- Ingeniería de comunicación de datos: es fundamental para este perfil conocer las normas de seguridad jurídica de los sistemas informáticos y comunicaciones, y en consecuencia de los datos e información en que se encuentren. Es muy importante el conocimiento de las normas que regulan la transferencia electrónica de datos y la contratación informática.
- Diseño de redes de comunicación: es esencial tener conocimiento de las normas en seguridad informática y de telecomunicaciones.
- Asistencia técnica: en este perfil conviene tener conocimiento de la protección jurídica del software, hardware, protección de datos, redes de telecomunicaciones e instrumentos de la sociedad de información.
- Ingeniería de integración y pruebas e implantación y pruebas: este perfil requiere conocimientos básicos. Conviene tener conocimiento de la protección jurídica del software, hardware, protección de datos, redes de telecomunicaciones e instrumentos de la sociedad de información.

Perfil profesional de gestión y explotación de tecnologías de la información:

- Consultoría de empresas de TI: en sus tareas le corresponde supervisar y coordinar los aspectos de la seguridad de los datos e implantación del sistema informático. En consecuencia necesita un conocimiento del sector, conocer las tendencias de la tecnología y adquirir capacidades para el diseño y arquitectura del sistema, de modo que será muy útil conocer la normativa de seguridad del sistema informático y de las nuevas tecnologías para implantarlo adecuadamente en la empresa

- Especialista en sistemas: en dicho perfil es necesario aportar los conceptos de bases de datos, redes, aplicaciones de internet, protección del software y hardware, conocimiento de la legislación que será complementario a sus capacidades técnicas
- Desarrollo de investigación y tecnología: la participación y gestión de proyectos ya sean de pequeña envergadura o proyectos de investigación internacionales, requieren de estrategia, planificación y control. Es relevante el conocimiento de las telecomunicaciones y de la sociedad de la información.
- Dirección de TIC (marketing, proyectos, dirección general): este perfil requiere un conocimiento amplio del resto de áreas tecnológicas de las Tecnologías de la Información y las Comunicaciones (TIC) y, por tanto, también de los instrumentos jurídicos que proporciona la normativa de la sociedad de la información, para hacer posible su tarea de consultoría y poner a disposición de los clientes productos que deben cumplir los condicionantes legales para su protección.

Esta adecuación de los tópicos relacionados con la asignatura a los distintos perfiles profesionales queda resumida en la siguiente tabla:

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Desarrollo de software y aplicaciones	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la protección jurídica del software, bases de datos y sus componentes, internet y comercio electrónico.
Arquitectura y diseño de software	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la protección jurídica del software, bases de datos, y sus componentes, junto con la de telecomunicaciones.
Diseño multimedia	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la protección jurídica del software, bases de datos, obras multimedia e Internet
Ingeniería de comunicación de datos.	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio y aplicación de las normas de seguridad jurídica de los sistemas informáticos y de comunicaciones.
Diseño de redes de comunicación	<ul style="list-style-type: none"> • Conocimiento y habilidad de la aplicación de las normas de telecomunicaciones
Asistencia técnica	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de los instrumentos de la sociedad de la información, de la protección de datos y protección jurídica de software, hardware y redes de telecomunicaciones.
Ingeniería de integración y pruebas e implantación y pruebas	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio y aplicación de protección jurídica del software y de las bases de datos.

PERFIL TITULACIÓN	PERFIL ASIGNATURA
Consultoría en empresas de TI	<ul style="list-style-type: none"> • Conocimiento y habilidad para el estudio de la protección jurídica del software, protección de datos, e instrumentos de la sociedad de la información.
Especialista en sistemas	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la protección jurídica del software, bases de datos y redes de telecomunicaciones e internet.
Desarrollo de investigación y tecnología	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de los instrumentos de la sociedad de la información, protección del software, telecomunicaciones y protección de datos.
Dirección de TIC	<ul style="list-style-type: none"> • Conocimiento y habilidad en la aplicación de la protección jurídica del software, protección de datos, instrumentos de la sociedad de la información.

Cuadro 1. *Adecuación de los tópicos relacionados con la asignatura Relaciones Jurídicas Básicas*

1.2 UBICACIÓN Y RELACIONES EN EL PLAN DE ESTUDIOS

La asignatura Relaciones Jurídicas Básicas forma parte del primer curso de Ingeniería Técnica en Informática de Gestión, como asignatura obligatoria, y de la Ingeniería Informática e Ingeniería Técnica en Informática de Sistemas, como asignatura optativa. En la titulación de Ingeniería Técnica en Informática de Gestión, una parte de la asignatura tiene una relación transversal con dos asignaturas de primer curso, aunque conviene precisar que Relaciones Jurídicas Básicas en su conjunto mantiene una relación transversal y objetiva con otras asignaturas de segundo y tercer curso, así como con otras optativas presentes en el resto de las titulaciones en Informática, pues es necesario conocer, al menos, los conceptos básicos de la materia de que se trata con la finalidad de adecuar el sistema informático a la legislación vigente. Concretamente, dichas asignaturas de primer curso son:

- **Fundamentos de programación I y II:** forma parte del primer y segundo cuatrimestre, respectivamente, del primer curso de las tres titulaciones de Informática como asignatura troncal de carácter anual. Dentro de sus descriptores aparece: análisis y diseños de programas, que constituirán el objeto de protección jurídica por las normas de propiedad intelectual, relevante para el entendimiento de la asignatura.

Además de esta relación, en cursos más avanzados, existen otras asignaturas cuyo objeto o actividad requiere la protección por el derecho informático:

- **Bases de datos I y II:** Base de Datos I es una asignatura troncal de segundo en las tres titulaciones de informática y Bases de datos II es una asignatura obligatoria en la titulación de ingeniería en informática y en

ingeniería técnica en informática de gestión, y optativa para la titulación de ingeniería técnica en informática de sistemas, que se imparten en el primer y segundo cuatrimestre, respectivamente. Sus descriptores se refieren a la estructura de información, ficheros y bases de datos, diseño y gestión de las bases de datos, las cuales gozan de una protección jurídica específica en las normas de propiedad intelectual

- **Programación y estructuras de datos:** forma parte del primer cuatrimestre del segundo curso de las tres titulaciones como asignatura troncal. Los descriptores de esta asignatura son: estructuras de datos y tipos abstractos de datos.
- **Diseño de sistemas de software:** se trata de una asignatura troncal de tercer curso en la titulación de Ingeniería técnica en informática de gestión, y además se ofrece como optativa en la Ingeniería técnica en informática de sistemas, impartida en ambos casos en el segundo cuatrimestre. Los descriptores de su contenido que guardan relación con la asignatura son: diseño, propiedades y mantenimiento del software de gestión
- **Metodología de Análisis de sistemas de información:** se trata de una asignatura troncal de tercer curso en la titulación de Ingeniería técnica en informática de gestión, que se desarrolla en el primer cuatrimestre. Los descriptores de su contenido son: planificación y gestión de proyectos informativos; análisis de aplicaciones de gestión.
- **Gestión de proyectos de software:** asignatura de primer cuatrimestre y obligatoria de tercer curso en la titulación de Ingeniería en informática y en ingeniería técnica en informática de gestión. Tiene como descriptor: la planificación de proyectos, documentación de proyectos, y herramientas software para gestión y control de proyectos.
- **Diseño y programación avanzada de aplicaciones:** asignatura obligatoria de tercer curso en la titulación de Ingeniería técnica en informática de gestión. Se ofrece como optativa en la titulación de Ingeniería técnica en informática de sistemas, que se imparte en el segundo cuatrimestre. Tiene como descriptor: aplicaciones internet.
- **Administración de servicios de Internet:** se trata de una asignatura optativa en las tres titulaciones de informática. Entre sus descriptores se encuentran: intranets y extranets de gestión de red. Servidores web.
- **Bases de datos multidimensionales:** se trata de una asignatura optativa de segundo cuatrimestre en las tres titulaciones. En sus contenidos figura el descriptor: diseño conceptual y lógico de las bases de datos multidimensionales, junto con diseño de almacenes de datos en Internet
- **Herramientas de programación:** se trata de una asignatura obligatoria de primer cuatrimestre en segundo curso, para la titulación de Ingeniería en informática, y optativa en la titulación de Ingeniería técnica en informática

de gestión y de sistemas. Los descriptores relacionados consisten en: organización del código fuente, gestión de proyectos, control de versiones y programación por contrato.

- **Aprendizaje computacional y extracción de la información:** se ofrece como asignatura optativa en las tres titulaciones. En sus contenidos figura el descriptor: extracción y recuperación de información de documentos electrónicos
- **Programación en Internet:** se ofrece como asignatura optativa en las tres titulaciones para el segundo cuatrimestre. Los descriptores de esta materia se refieren a: desarrollo y programación de sistemas de acceso a bases de datos de Internet; planificación, diseño y administración de sitios Web; migración de aplicaciones a entornos en Internet; herramientas de desarrollo; diseño y programación de elementos multimedia en Internet.
- **Sistemas y servicios de telecomunicación:** se ofrece como asignatura optativa en las tres titulaciones en informática para el segundo cuatrimestre. Los descriptores de esta materia se refieren a: redes públicas de telefonía; comunicaciones por cable: comunicaciones por móviles.
- **Sistemas operativos:** asignatura del segundo cuatrimestre que forma parte del segundo curso de la titulación de Ingeniería Técnica en Informática de Sistemas como asignatura troncal. Sus descriptores contienen: organización, estructura y servicio de los sistemas operativos; gestión y administración de memoria y de procesos; gestión de entrada salida; sistemas de ficheros; seguridad.
- **Administración de sistemas operativos en red:** asignatura de segundo cuatrimestre y obligatoria en tercer curso de la titulación de Ingeniería Técnica en Informática de Sistemas. Entre sus descriptores trata el sistema de archivo y seguridad, y los sistemas cliente/servidor.
- **Sistemas de transporte de datos:** asignatura de segundo cuatrimestre de carácter troncal de cuarto curso en la titulación de Ingeniería en Informática y obligatoria en tercer curso de la titulación de Ingeniería Técnica en Informática de Sistemas. Entre sus descriptores trata: comunicaciones e interconexión de redes
- **Teoría de la información y de la codificación:** se ofrece como asignatura optativa de primer cuatrimestre en las tres titulaciones. Entre sus contenidos figura el descriptor: criptografía, útil en materia de seguridad informática
- **Ingeniería del software I y II:** se trata de una asignatura de segundo cuatrimestre de carácter troncal de cuarto y quinto curso de la titulación de Ingeniería en Informática. Los descriptores de la asignatura son: diseño, propiedades y mantenimiento de software; planificación y gestión de proyectos informáticos.

- **Sistemas informáticos:** forma parte del quinto curso de la titulación de Ingeniería en informática como asignatura troncal. Los descriptores se refieren a los siguientes contenidos: diseño, gestión y evaluación de sistemas informáticos; entornos de sistemas informáticos; tecnologías avanzadas de sistemas de información, bases de datos y sistemas operativos; proyectos de sistemas informáticos.
- **Administración de bases de datos:** se ofrece como asignatura optativa de segundo cuatrimestre en la titulación de ingeniería en informática. Los descriptores contienen: seguridad, recuperación y administración
- **Tecnologías web:** se ofrece como asignatura optativa en las tres titulaciones en el primer cuatrimestre. En sus contenidos figura el descriptor: programación de clientes web y seguridad

La siguiente tabla, describe las asignaturas de las tres titulaciones de acuerdo con su carácter troncal, obligatoria y optativa:

ASIG. TRONCAL	ASIG. OBLIGATORIA	ASIG. OPTATIVA
Fundamentos de programación I y II	Gestión de proyectos de software	Diseño y programación avanzada de aplicaciones
Bases de datos I y II	Diseño y programación avanzada de aplicaciones	Administración de servicios de Internet
Programación y estructuras de datos	Herramientas de programación	Bases de datos multidimensionales
Diseño de sistemas de software	Administración de sistemas operativos en red	Herramientas de programación
Metodología de Análisis de sistemas de información	Sistemas de transporte de datos	Aprendizaje computacional y extracción de la información
Sistemas operativos		Programación en Internet
Sistemas de transporte de datos		Sistemas y servicios de telecomunicación
Ingeniería del software I y II		Teoría de la información y de la codificación
Sistemas informáticos		Administración de bases de datos
		Tecnologías web

Cuadro 2. Descripción de las asignaturas

La siguiente figura resume las relaciones entre la asignatura de Relaciones Jurídicas Básicas con las materias descritas en la tabla anterior:

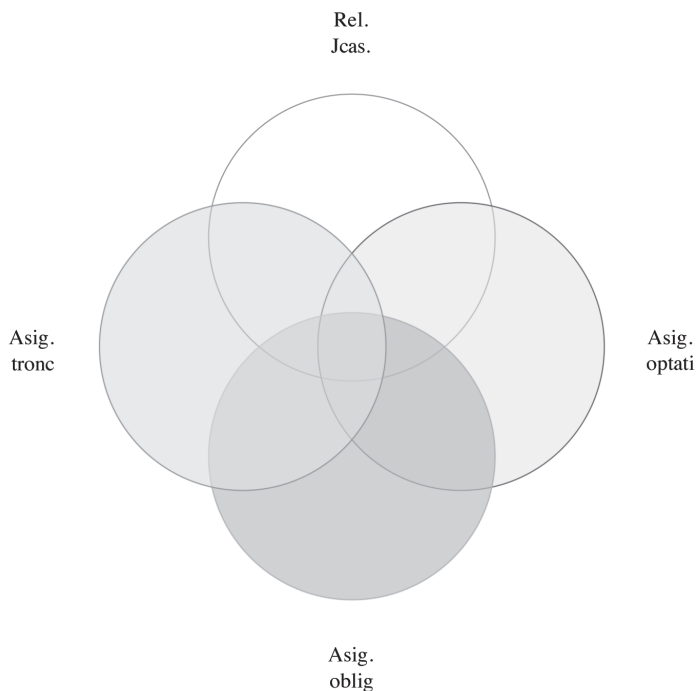


Ilustración 19. relaciones de Relaciones Jurídicas Básicas con resto de asignaturas

2. OBJETIVOS

2.1 OBJETIVOS GENERALES

2.1.1 Objetivos instrumentales generales (saber y saber hacer)

Además de los objetivos instrumentales generales cOI1, cOI2, cOI3, cOI4, cOI5 y cOI6, desarrollados en el artículo introductorio de este libro, como objetivos comunes a todas las asignaturas, planteamos los siguientes:

- OI1: Comprender e interpretar la distribución de los poderes del Estado, clases de Administraciones Públicas y el sistema de fuentes en derecho administrativo
- OI2: Comprender, interpretar y analizar las normas jurídicas que regulan el uso de los sistemas informáticos y nuevas tecnologías.
- OI3: Comprender, interpretar y analizar el encaje de las normas del derecho de la informática en el ordenamiento jurídico.

- OI4: Ser capaz de identificar los tipos de Administraciones Públicas y conocer el sistema de fuentes de derecho administrativo.
- OI5: Adquirir aquellos conceptos jurídicos básicos para solucionar los conflictos que puedan surgir en la aplicación de la protección del software, protección de datos, y de la regulación de las nuevas tecnologías
- OI6: Conocer la importancia de la seguridad jurídica informática y ser capaz de aplicarla especialmente en dicho ámbito.
- OI7: Adquirir la capacidad necesaria para precisar la inadecuación del sistema informático de que se trate, a la normativa existente.
- OI8: Desarrollar su capacidad de conocer los instrumentos necesarios para asimilar nuevos cambios normativos en la materia que pueda necesitar en su vida profesional.

2.1.2 Objetivos interpersonales generales (ser y estar)

Estos objetivos se corresponden con los objetivos interpersonales generales comunes a todas las asignaturas de segundo curso. Concretamente, estos son los objetivos cOIP1, cOIP2 y cOIP3, detallados en el artículo introductorio de este libro.

Junto a ellos, se añade el siguiente objetivo interpersonal:

- OIP1: Capacidad de análisis y de síntesis
- OIP2: Comunicación oral y escrita
- OIP3: Razonamiento crítico

2.1.3 Objetivos sistémicos generales

Estos objetivos se corresponden con los objetivos sistémicos generales, comunes a todas las asignaturas de segundo curso. Concretamente, estos son los objetivos cOS1, cOS2, cOS3 y cOS4, detallados en el artículo introductorio de este libro.

2.2 COMPETENCIAS

2.2.1 Competencias instrumentales (saber y saber hacer)

Dentro de las competencias instrumentales, distinguiremos entre habilidades cognitivas, capacidades metodológicas, destrezas tecnológicas y destrezas lingüísticas.

Habilidades cognitivas (saber):

Además de tener en cuenta las habilidades cognitivas cCIC1 y cCIC2, desarrolladas en artículo introductorio de este libro, tendremos en cuenta las siguientes:

- CIC1: Entender la distribución de los poderes del Estado y los tipos de Administraciones Públicas.

- CIC2: Comprender el sistema de fuentes de derecho administrativo.
- CIC3: Entender la diferencia entre informática jurídica y derecho de la informática.
- CIC4: Comprender la metodología básica para poder realizar un informe con validez ante la administración pública y los tribunales.
- CIC5: Entender los mecanismos de protección del software y los requisitos para la inscripción en el registro de la propiedad intelectual.
- CIC6: Entender el objeto de la protección de las bases de datos.
- CIC7: Saber la normativa de protección de datos de carácter personal y entender su aplicación a una empresa.
- CIC8: Saber y entender los instrumentos para el comercio electrónico, la seguridad de los datos y la contratación electrónica.
- CIC9: Saber los aspectos para dar eficacia a los documentos electrónicos y la firma electrónica.
- CIC10: Entender las diferencias entre una infracción civil, penal y administrativa
- CIC11: Comprender y saber la normativa en materia de telecomunicaciones

Capacidades metodológicas (saber hacer):

Las capacidades metodológicas se han agrupado por bloques temáticos de la asignatura. Además de éstas, se consideran las capacidades metodológicas cCIM1, cCIM2 y cCIM3, introducidas a nivel general en el artículo introductorio de este libro.

- CIM1: Aplicar los requisitos previstos en la normativa existente sobre comunicaciones electrónicas y software
- CIM2: Ser capaz de inscribir los ficheros que contienen datos de carácter personal en el registro de protección de datos, y elaborar el documento de seguridad
- CIM3: Ante una situación real concreta, ser capaz de aplicar la metodología a seguir para poder adecuar la normativa existente y tomar decisiones correctas
- CIM4: Ser capaz de realizar un informe.
- CIM5: Ser capaz de aplicar los contratos informáticos.

Destrezas tecnológicas (saber hacer):

Además de la destreza tecnológica cCIT1, incluida en el artículo introductorio de este libro, se consideran las siguientes:

- CIT1: Manejar con fluidez las bases de datos jurídicas y páginas web que indique el profesor que servirán de guía para la realización de las actividades relacionadas con la asignatura.

Destrezas lingüísticas (saber hacer):

Además de las destrezas lingüísticas cCIL1 y cCIL2, incluidas en el artículo introductorio de este libro, se considera la siguiente:

- CIL1: Adquirir y utilizar con fluidez un buen lenguaje jurídico en esta materia, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier medio de protección jurídica utilizado para el sistema informático y en la redacción de los informes técnicos en relación con la normativa informática.

2.2.2 Competencias interpersonales (ser y estar)

Las competencias interpersonales se han dividido en competencias para tareas colaborativas y competencias relativas al compromiso con el trabajo.

Competencias para tareas colaborativas:

Las competencias relativas a tareas colaborativas se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTC1 y cCIPTC2.

Compromiso con el trabajo:

Las competencias relativas al compromiso con el trabajo se refieren a las competencias comunes dadas en el artículo introductorio de este libro. Más concretamente, estas competencias son las etiquetadas como cCIPTR1, cCIPTR2, cCIPTR3 y cCIPTR4.

2.2.3 Competencias sistémicas

Las competencias sistémicas hacen referencia a la integración de capacidades cognitivas, destrezas prácticas y disposiciones recogidas en el artículo introductorio de este libro. Dichas competencias se recogen con las etiquetas cCS1, cCS2, cCS3, cCS4 y cCS5.

La siguiente tabla describe los objetivos generales de la asignatura y las competencias específicas de la asignatura (saber):

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
1. Conceptuales (saber)	1. Conceptuales (saber)
<ul style="list-style-type: none"> • OI1: Comprender e interpretar la distribución de los poderes del Estado, clases de Administraciones Públicas y el sistema de fuentes en derecho administrativo • OI2: Comprender, interpretar y analizar las normas jurídicas que regulan el uso de los sistemas informáticos y nuevas tecnologías. • OI3: Comprender, interpretar y analizar el encaje de las normas del derecho de la informática en el ordenamiento jurídico. 	<ul style="list-style-type: none"> • CIC1: Entender la distribución de los poderes del Estado y los tipos de Administraciones Públicas • CIC2: Comprender el sistema de fuentes de derecho administrativo • CIC3: Entender la diferencia entre informática jurídica y derecho de la informática • CIC4: Comprender la metodología básica para poder realizar un informe con validez ante la administración pública y los tribunales.

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
1. Conceptuales (saber)	1. Conceptuales (saber)
<ul style="list-style-type: none"> • OI4: Ser capaz de identificar los tipos de Administraciones Públicas y conocer el sistema de fuentes de derecho administrativo. • OI5: Adquirir aquellos conceptos jurídicos básicos para solucionar los conflictos que puedan surgir en la aplicación de la protección del software, protección de datos, y de la regulación de las nuevas tecnologías • OI6: Conocer la importancia de la seguridad jurídica informática y ser capaz de aplicarla especialmente en dicho ámbito. • OI7: Adquirir la capacidad necesaria para precisar la inadecuación del sistema informático de que se trate a la normativa existente. • OI8: Desarrollar su capacidad de conocer los instrumentos necesarios para asimilar nuevos cambios normativos en la materia que pueda necesitar en su vida profesional. 	<ul style="list-style-type: none"> • CIC5: Entender los mecanismos de protección del software y los requisitos para la inscripción en el registro de la propiedad intelectual • CIC6: Entender el objeto de la protección de las bases de datos • CIC7: Saber la normativa de protección de datos de carácter personal y entender su aplicación a una empresa • CIC8: Saber y entender los instrumentos para el comercio electrónico, la seguridad de los datos y la contratación electrónica • CIC9: Saber los aspectos para dar eficacia a los documentos electrónicos y la firma electrónica. • CIC10: Entender las diferencias entre una infracción civil, penal y administrativa • CIC11: Comprender y saber la normativa en materia de telecomunicaciones

Cuadro 3. Análisis de coherencia: competencias conceptuales (saber)

La siguiente tabla describe los objetivos generales de la asignatura y las competencias específicas de la asignatura (saber hacer):

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
2.Procedimentales (saber hacer)	1. Procedimentales (saber hacer)
<ul style="list-style-type: none"> • cOI1: Aplicar los conocimientos adquiridos mediante la resolución de problemas y prácticas de ordenador. • cOI2: Utilizar con fluidez el software necesario en las prácticas relacionadas con cada asignatura. • cOI3: Adquirir y utilizar un buen lenguaje formal, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier proceso. • cOI4: Conocer y utilizar la terminología usual de cada asignatura. • cOI5: Adquirir un buen manejo de la bibliografía existente en la asignatura, de forma que se potencia su autosuficiencia a la hora de completar su formación. • cOI6: Comprender el ámbito de acción de la asignatura dentro de la titulación de Informática y dentro de los perfiles profesionales. 	<ul style="list-style-type: none"> • cCIC1: Conocer y entender las relaciones que mantiene la asignatura con el resto de asignaturas de la titulación de informática a partir de la introducción explícita en clase de ejemplos y comentarios aprovechando puntos relevantes del temario. • cCIC2: Conocer y entender algunas de las capacidades, aptitudes y conocimientos que la asignatura aporta para el desarrollo de los diversos perfiles profesionales, ayudándose de la introducción explícita en clase de ejemplos y comentarios en puntos relevantes del temario. • cCIM1: Ser capaz de tomar decisiones de manera razonada. • cCIM2: Capacidad de análisis y síntesis. • cCIM3: Ser capaz de manejar la bibliografía relacionada con la asignatura. • cCIT1: Habilidades básicas de consulta de la red informática para la obtención y manejo de información relacionada con la asignatura. • CIL1: Adquirir y utilizar con fluidez un buen lenguaje científico, tanto oral como escrito, siendo riguroso en las explicaciones de fenómenos o experimentos. • CIL2: Conocer y utilizar la terminología usual de la asignatura, tanto en castellano y/o en valenciano, y conocer dicha terminología en inglés.

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
2.Procedimentales (saber hacer)	1. Procedimentales (saber hacer)
	<ul style="list-style-type: none"> • CIM1: Ser capaz de aplicar los requisitos previstos en la normativa existente sobre comunicaciones electrónicas y software. • CIM2: Ser capaz de inscribir los ficheros que contienen datos de carácter personal en el registro de protección de datos, y elaborar el documento de seguridad • CIM3: Ante una situación real concreta, ser capaz de aplicar la metodología a seguir para poder adecuar la normativa existente y tomar decisiones correctas • CIM4: Ser capaz de realizar un informe. • CIM5: Ser capaz de aplicar los contratos informáticos. • CIT1: Manejar con fluidez las bases de datos jurídicas y páginas web que indique el profesor que servirán de guía para la realización de las actividades relacionadas con la asignatura. • CIL1: Adquirir y utilizar con fluidez un buen lenguaje jurídico en esta materia, tanto oral como escrito, siendo riguroso en las explicaciones de cualquier medio de protección jurídica utilizado para el sistema informático y en la redacción de los informes técnicos en relación con la normativa informática

Cuadro 4. *Análisis de coherencia: competencias procedimentales (saber hacer)*

La siguiente tabla describe los objetivos generales de la asignatura y las competencias específicas de la asignatura (ser y estar):

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
3. Disposiciones (ser y estar)	1. Disposiciones (ser y estar)
<ul style="list-style-type: none"> • cOIP1: Destrezas para la participación responsable: capacidad de coordinación, asistencia, contribuciones al grupo, etc. • cOIP2: Capacidad de trabajar en equipo adquiriendo y mejorando las habilidades sociales y la inteligencia emocional. • cOIP3: Comprometerse de forma ética con el trabajo, con el resto de integrantes del grupo y consigo mismo. cOS1: Capacidad de integrar los conocimientos, métodos, algoritmos y destrezas prácticas de cada asignatura para resolver situaciones reales relacionadas con la informática y otras disciplinas relacionadas. • cOS2: Reforzar el hábito de plantearse interrogantes; ante un problema deben preguntarse por el número de soluciones, la relación entre ellas, cómo afectaría en las condiciones iniciales alguna modificación, etc. • cOS3: Capacidad de aplicar y relacionar, de forma autónoma, los contenidos de cada asignatura de forma interdisciplinar. 	<ul style="list-style-type: none"> • cCIPTC1: Ser capaz de realizar opcionalmente un trabajo en equipo de ampliación de la materia estudiada en cada asignatura. • cCIPTC2: Ser capaz de trabajar en equipo para resolver cuestiones y problemas relacionados con la materia estudiada en cada asignatura. • cCIPTR1: Se ha de definir un plan de trabajo en el que el volumen de trabajo de todos los miembros del equipo sea similar. • cCIPTR2: Una vez finalizado el trabajo, todos los miembros del grupo deben conocer en profundidad todo el desarrollo realizado. • cCIPTR3: Se debe cumplir el plazo de entrega de dichos trabajos. • cCIPTR4: Es importante adquirir un compromiso ético entre todos los componentes del grupo. • cCS1: Capacidad de aplicar los conocimientos, métodos y algoritmos vistos en cada asignatura a situaciones y problemas concretos del área de informática y de otras disciplinas relacionadas.

OBJETIVOS GENERALES ASIGNATURA	COMPETENCIAS ESPECÍFICAS ASIGNATURA
<p>3. Disposiciones (ser y estar)</p> <ul style="list-style-type: none"> • cOS4: Adquirir una comprensión del método científico, a través de diversas actividades realizadas en la asignatura, y asimilar su importancia como manera de pensar y actuar en su labor de científico e ingeniero, fomentando su capacidad de abstracción y su espíritu crítico 	<p>1. Disposiciones (ser y estar)</p> <ul style="list-style-type: none"> • cCS2: Capacidad de aprender y aplicar, de forma autónoma e interdisciplinar, nuevos conceptos y métodos relacionados con cada asignatura. • cCS3: Motivación por la calidad y por la creatividad. • cCS4: Capacidad de adoptar el protocolo dado por el método científico en el planteamiento y realización de trabajos diversos tanto a nivel académico como profesional. • cCS5: Capacidad de asimilación y adaptación a la evolución del estado del arte en el ámbito de desarrollo profesional (adaptación a nuevas situaciones).

Cuadro 5. *Análisis de coherencia: competencias disposicionales (ser y estar)*

3. PRERREQUISITOS

La asignatura de Relaciones Jurídicas Básicas no presentan ningún prerrequisito específico, al tener en cuenta que se trata de una materia que va a suponer la primera toma de contacto del alumno con ordenamiento jurídico, y en particular, con el derecho de las nuevas tecnologías. En consecuencia, cuando el alumno haya cursado la asignatura dispondrá de unas nociones teóricas y prácticas que serán reforzadas con el manejo de las fuentes legales, doctrinales, jurisprudenciales necesarias para garantizar la actualización de sus conocimientos en el futuro. En definitiva, se trata que el alumno adquiera los conceptos básicos y las habilidades precisas para interpretar las normas que se refieren a las nuevas tecnologías y para su aplicación en su técnica profesional.

4. BLOQUES Y TEMAS DE LA ASIGNATURA

4.1 BLOQUES DE CONTENIDOS DE APRENDIZAJE

La asignatura de Relaciones Jurídicas Básicas ha sido dividida en dos bloques temáticos. Cada uno engloba diversas unidades temáticas, que se exponen a continuación de forma detallada.

Bloque 1: Introducción al derecho informático: aspectos generales del derecho

- Tema 1: Los poderes en el Estado Constitucional.
- Tema 2: Las normas jurídicas
- Tema 3: La Ley y las normas equiparadas a la Ley.
- Tema 4: El Reglamento.

Bloque 2: El contenido básico del Derecho Informático

- Tema 5: Introducción al derecho de la informática.
- Tema 6: La protección jurídica del software
- Tema 7: La protección de datos

Bloque 3: Otros sectores del derecho informático

- Tema 8: El comercio electrónico y la contratación electrónica
- Tema 9: La contratación informática
- Tema 10: El documento electrónico

4.2 TEMAS O UNIDADES DE CONTENIDO. DESARROLLO

Bloque 1.Tema 1. Los poderes en el Estado Constitucional.

1. La Constitución española y el Estado Social y Democrático de Derecho.
2. Los derechos fundamentales y las libertades públicas.
3. El Tribunal Constitucional y el Defensor del Pueblo.
4. La distribución territorial de los poderes del Estado. Esquema de las Administraciones existentes.

Bloque 1.Tema 2. Las normas jurídicas

1. Las normas jurídicas.
2. Principio de estatalidad de las normas.
3. Principio de racionalidad de las normas jurídicas.
4. Clasificación de las fuentes del Derecho.
5. El nuevo modelo de articulación de fuentes primarias.
6. Esquema de las fuentes en el Derecho español.

Bloque 1.Tema 3. La Ley y las normas equiparadas a la ley

1. La Ley.
2. Normas no parlamentarias equiparadas a la Ley.
3. La vigencia en España de la normativa con rango de ley de la Comunidad Europea.

Bloque 1.Tema 4. El Reglamento

1. Concepto.
2. Fundamentos de la potestad reglamentaria.
3. Titulares de la potestad reglamentaria.
4. Clases de reglamentos.
5. Límite de la potestad reglamentaria.
5. Procedimiento de elaboración de los reglamentos.

Bloque 2. Tema 5. Introducción al derecho de la Informática

1. Introducción.
2. La informática jurídica.
3. El Derecho de la informática.

Bloque 2. Tema 6 La protección jurídica del software.

1. Generalidades: Justificación, antecedentes y evolución.
 - 1.1 La exclusión de la patentabilidad
 - 1.2 El origen de la cuestión
 - 1.3 Derecho internacional y comparado.
 - 1.4 El derecho comunitario y su transposición
2. La protección jurídica del software.
 - 2.1 El derecho de autor en general
 - 2.2 El régimen de los programas de ordenador en la regulación sobre propiedad intelectual:
 - 2.2.1 Aspectos generales
 - 2.2.2 Régimen jurídico
 - 2.2.2.1. El concepto legal de programa de ordenador y la relevancia del mismo para su protección jurídica
 - 2.2.2.2 Titulares
 - 2.2.2.3 Duración
 - 2.2.2.4 Contenido
 - 2.2.2.5 Los límites y el concepto de usuario legítimo
 - 2.2.2.6 La infracción de los derechos de propiedad intelectual
 - 2.2.2.7 Cláusula de salvaguardia
 - 2.2.2.8 Tipos de protección de los programas de ordenador: el registro de la Propiedad Intelectual
 - 2.2.2.9 Protección penal
3. La protección jurídica de las bases de datos.
 - 3.1 Origen de la protección específica
 - 3.2 Concepto y clases
 - 3.3 Personas que intervienen en las bases de datos
 - 3.4 Relaciones entre creador y distribuidor
 - 3.5 Relaciones entre distribuidor y usuario
 - 3.6 Objeto de protección
 - 3.7 Contenido
 - 3.8 Duración.
4. Supuestos problemáticos
 - 4.1 Obras multimedia
 - 4.2 Internet.

Bloque 2. Tema 7. La protección de datos.

1. Concepto.
2. Clases de datos.
3. Fundamento de la protección de datos: El derecho a la autodeterminación informativa.

4. *Ámbito internacional y comunitario*
5. El régimen jurídico español de la protección de datos de carácter personal
 - 5.1 El tratamiento de los datos de carácter personal
 - 5.2 El contenido del tratamiento: derechos y obligaciones de las partes
 - 5.3 Tratamientos especiales
 - 5.4 Especial referencia al movimiento internacional de datos
 - 5.5 Medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal.
6. La Agencia de Protección de Datos
7. Protección civil de los datos
8. Protección penal de los datos

Bloque 3. Tema 8. El comercio electrónico y la contratación electrónica

1. La contratación electrónica
2. Los nombres de dominio
3. El comercio electrónico: la Ley de servicios de la sociedad de la información
4. Objeto y ámbito de aplicación
5. Obligaciones y régimen de responsabilidad de los prestadores de servicios de la sociedad de la información
6. Comunicaciones comerciales y contratación electrónica
7. Solución judicial y extrajudicial de conflictos
8. Infracciones y Sanciones

Bloque 3. Tema 9. La contratación informática.

1. Concepto
2. El objeto en el contrato informático
3. Clases
4. Fases del contrato
5. Partes del contrato
6. Cláusulas tipo

Bloque 3. Tema 10. El documento electrónico

1. Concepto: los documentos electrónicos
2. La conservación de documentos en soporte informático
3. Problemas de autenticación y problemas de prueba
4. La firma electrónica: características
 - 4.1 Concepto y caracteres
 - 4.2 Autoridades de certificación privadas: la prestación de servicios de certificación
 - 4.3 Obligaciones del prestador de servicios

- 4.4 La certificación
- 4.5 Utilización de la firma electrónica
- 5.Experiencias: El DNI electrónico

5. METODOLOGÍA Y ESTRATEGIAS DE APRENDIZAJE

5.1 METODOLOGÍA DOCENTE

La metodología docente que se ha desarrollado en el artículo introductorio de este libro desde el punto de vista general para el segundo curso de la titulación de informática, junto con la que se describe en este apartado, completa la metodología de la asignatura de Relaciones Jurídicas Básicas. En este ámbito, se trata de implantar una metodología acorde una actitud participativa y activa del alumno en el proceso de aprendizaje, siempre en esta materia desde el enfoque de la lección magistral, pues la totalidad de los créditos de la asignatura son de índole teórica. En consecuencia, para evitar que la adquisición de conocimientos pueda resultar tediosa, la idea básica consiste en que el profesor dedica el tiempo de clase en explicar los conceptos básicos y jurídicos más relevantes, desarrollar ejercicios relativos al tema correspondiente, e introducir al alumno en las herramientas adecuadas para aplicar el derecho de la informática y comprender mejor su contenido. El alumno va a dedicar la mayor parte de su tiempo de aprendizaje a preparar la asignatura y adquirir los conocimientos, así como en la realización de los trabajos no terminados en clase.

Por tanto, se presenta un modelo didáctico que ya no se basa exclusivamente en la clásica lección magistral, en el que combinan determinadas actividades complementarias con la finalidad de mejorar el estudio del alumno y orientar su tiempo de aprendizaje. Las actividades que se proponen son las siguientes:

Lección magistral: la lección teórica es el método utilizado con carácter general para desarrollar las asignaturas relacionadas con el derecho. La clase teórica no va a consistir en una mera reproducción de conocimientos sino que va a cumplir una función dinamizadora del estudio de la materia. La clase de teoría se orienta a proporcionar al alumno tanto el conocimiento como la aplicación práctica del sistema jurídico relacionado con el fenómeno informático, siempre desde la óptica de fomentar la participación activa del estudiante y su capacidad crítica en torno a la legislación y jurisprudencia en esta materia. En definitiva, se pretende flexibilizar el contenido de la clásica lección magistral, haciendo más asequible el aprendizaje al alumno en el aula, y lograr que sea un buen medio para enseñar.

Clases de teoría con apoyo de material audiovisual: en el caso de las disciplinas jurídicas, como ocurre en el derecho de la informática, el apoyo audiovisual resulta imprescindible. La clase de teoría se apoya con el material audiovisual elaborado por el profesor y disponible para el alumnado, que le puede servir de guía sobre los contenidos más importantes de la asignatura. Además, el profesor de la asignatura ha elaborado unos materiales docentes que

incluyen la totalidad de los contenidos de la misma, aparte de los manuales recomendados. De esta forma, el alumno puede prestar mayor atención en el aula y asimilar mejor la explicación dado que no tiene que estar tan preocupado por la toma de apuntes en las clases teóricas. En las explicaciones de clase, se muestran al alumno las herramientas disponibles on line, se comentan ejemplos prácticos y sentencias, siempre que el contenido así lo requiera.

Trabajos complementarios: los ejercicios propuestos en clase de carácter teórico-práctico que tienen como finalidad sistematizar el contenido de la materia así como facilitar la comprensión y el aprendizaje de la teoría objeto de estudio; sirven como trabajos complementarios, en cuanto que pueden mejorar la nota final de la asignatura y deberán realizarse de forma individual y voluntaria.

Tutorías de atención al alumno: se trata de tutorías individualizadas para el alumno que lo necesita, en las que tiene a su disposición al profesorado de esta disciplina para atender sus dudas sobre el contenido de la materia, orientación en la realización de los ejercicios propuestos en clase, o también, en relación con la organización de la asignatura.

5.2 ESTRATEGIAS DE APRENDIZAJE

Además de las estrategias de aprendizaje establecidas nivel común en el artículo introductorio de este libro, y que se corresponden con los medios tradicionales como las transparencias o apuntes, se utilizarán las presentaciones por ordenador, así como las herramientas que pone a disposición tanto del profesor como del alumno, el campus virtual, tales como: anuncios, tutorías, materiales, debates o test de autoevaluación, o la ficha de la asignatura. Todo ello, sin obviar la posible creación de una página web que incluya toda la información sobre la misma. Aquí, el impulso y el desarrollo de estas actuaciones tienen un mayor sentido si cabe dada la relación de esta titulación con el desarrollo de las nuevas tecnologías.

Así, la estrategia de aprendizaje detalla las actuaciones concretas para la aplicación de la metodología que se ha propuesto en el apartado anterior, de acuerdo con las siguientes fases:

Recopilación de toda la documentación de la asignatura.

- Los alumnos disponen en el campus virtual de toda la información y normativa relacionada con el desarrollo de la asignatura: temas, evaluación, bibliografía, enlaces, etc.
- El profesor proporciona la información sobre la asignatura debidamente actualizada en el campus virtual, de modo que, sirve como lugar de comunicación con los alumnos
- Los alumnos conocerán y manejarán con fluidez esa documentación
- Habitualmente, el alumno debe consultar el campus virtual para estar al corriente del desarrollo de la asignatura y de la información relativa a la misma.

Planificación de las clases teóricas

- Las clases teóricas se dedican tanto a la explicación de contenidos nuevos para el alumno como a la resolución de los ejercicios propuestos.
- El alumno debe haber realizado una lectura previa de los epígrafes correspondientes a la sesión de teoría que se trate.
- En la clase, el profesor presentará un resumen de lo acontecido en la sesión anterior, para a continuación presentar el contenido de la clase y el contexto en el que se desarrolla, dando la oportunidad a los alumnos a que expongan los posibles conceptos que presenten problemas derivados de la lectura previa que han realizado. Finalmente, el profesor procederá a la explicación de los contenidos del tema de que se trate y facilitará las preguntas y ejercicios que mejoren la comprensión de los contenidos vistos en la sesión de teoría.
- Una vez realizada la clase de teoría, se debe estudiar de forma autónoma su contenido y en caso de no entender algo intentar primero contrastarlo con otros compañeros o utilizando la bibliografía recomendada. Si esto no es suficiente, se acudirá a tutorías para intentar solucionar el problema.

Ejercicios en el aula. Es este periodo, los alumnos resolverán los ejercicios y prácticas facilitadas por el profesor, en las que se desarrollan técnicas para la resolución de los conflictos y se valora el cumplimiento o no de la legalidad vigente aplicada a un caso concreto. En este momento de la clase, se intenta orientar al alumno en el uso de las herramientas para resolver estas cuestiones y método de su aprendizaje.

- El alumno debe plantear sus dudas que le hayan surgido en la realización de los ejercicios.
- El alumno debe elaborar un cuaderno de trabajo para el seguimiento de las actividades por el profesor.
- Al final de los temas 5, 6,7,8,9,y 10, se realizará una autoevaluación.

Autoevaluación: esta fase es muy importante en el desarrollo de a confianza de los alumnos en relación con la cuantificación del conocimiento que uno mismo posee en la materia tratada. Por ello, una vez que ha seguido las indicaciones anteriores para la planificación de las clases teóricas y ha resuelto los ejercicios propuestos, debe ser capaz de comprobar si el tema ha sido totalmente entendido. En caso de no ser así, debe incidir en el estudio de los contenidos que crea tener más flojos, utilizando si lo cree conveniente las tutorías y el tiempo para solucionar las dudas en clase o bien haciendo uso de la bibliografía. Es conveniente que no se utilicen los apuntes la primera vez que se intentan resolver los ejercicios propuestos después de haber realizado una lectura detenida del tema.

Evaluación final: si el resultado de los ejercicios de autoevaluación y de las prácticas propuestas ha sido satisfactorio, el estudiante estará en disposición de superar la asignatura. En ese caso, se encontrará preparado para realizar el examen final. De forma opcional se podrá hacer trabajos individuales complementarios, para subir la nota siempre y cuando el trabajo realizado a lo largo del curso se considere satisfactorio y se haya obtenido en el examen final una puntuación mínima de 4,5 puntos.

6. PLAN DE TRABAJO DE LOS ALUMNOS. ESPECIFICACIÓN DEL TIEMPO Y ESFUERZO DE APRENDIZAJE

En las siguientes tablas se esquematiza cuál va a ser el plan de trabajo de esta asignatura. Se distingue entre horas presenciales dedicadas a la realización de actividades en las aulas, donde el profesorado juega un papel primordial, y horas no presenciales dedicadas al trabajo y esfuerzo personal realizado en la asignatura, de forma autónoma, por los estudiantes.

Las horas presenciales se dedican a la realización de actividades en las aulas, y se componen de clases convencionales de teoría donde el profesor expone y explica la materia a tratar siguiendo la metodología docente expresada anteriormente, junto con el comentario y resolución de ejercicios en grupo.

NÚMERO DE HORAS PRESENCIALES			
ACTIVIDAD		Clases de teoría	
		Lección magistral	Ejercicios en el aula
Presentación			1
BLOQUE 1	Tema 1	2	1
	Tema 2	1,75	0,25
BLOQUE 2	Tema 3	2	1
	Tema 4	0,75	0,25
BLOQUE 3	Tema 5	1	1
	Tema 6	2,5	1,5
	Tema 7	2,5	1,5
	Tema 8	2	1
	Tema 9	1	1
	Tema 10	1	1
Preparación del examen final:		1	
Examen final:		1	
TOTAL: 45		29	
TOTAL HORAS CON PROFESOR		49,5	

Cuadro 6. Plan de trabajo de la asignatura, número de horas presenciales

Las horas no presenciales se dedican a completar el desarrollo del temario analizado en clase por el profesor y hacer los ejercicios que completan y favorecen la comprensión de la materia. En este tiempo, además se realizan las tareas de recopilación de información, autoevaluación y preparación del examen final.

NÚMERO DE HORAS NO PRESENCIALES			
ACTIVIDAD		Estudio de la asignatura	
		Teoría	Ejercicios en el aula
Presentación:			0
BLOQUE 1:	Tema 1	3,5	0,25
	Tema 2	3,06	0,06
BLOQUE 2:	Tema 3	3,5	0,25
	Tema 4	1,31	0,06
BLOQUE 3:	Tema 5	1,75	0,25
	Tema 6	4,375	0,375
	Tema 7	4,375	0,375
	Tema 8	3,5	0,25
	Tema 9	1,75	0,25
	Tema 10	1,75	0,25
Recogida de información		5	
Tutorías			
Preparación del examen final:		42,51	
TOTAL: 128,3		78,75	

Cuadro 7. Plan de trabajo de la asignatura, número de horas no presenciales

7. BIBLIOGRAFÍA

7.1 BIBLIOGRAFÍA BÁSICA

Davara Rodríguez, M.A. (2005) *Manual de Derecho Informático*, Aranzadi, Cizur Menor.

Martín Mateo, R. (2005) *Manual de Derecho Administrativo*, 24 ed., Aranzadi.

7.2 BIBLIOGRAFÍA COMPLEMENTARIA

APDCM (2003) *Manual de Protección de datos para las Administraciones Públicas*, Madrid: Thomson-Civitas.

BALLESTEROS MOFFA, L.A (2005) *La privacidad electrónica. Internet en el centro de protección*, Valencia: Tirant lo Blanch.

Dir. GARCÍA MEXÍA, P. (2002) *Principios de Derecho de Internet*, Valencia: Tirant lo blanch.

FERNANDEZ SALMERÓN, M. (2003) *La protección de los datos personales en las Administraciones Públicas*, Madrid: Thpmson-Civitas.

- INAP (2003) Libro Blanco sobre la Administración electrónica y la protección de datos personales, INAP, julio, 2003.
- MUÑOZ MACHADO, S. (2000) La regulación en la red. Poder y Derecho en Internet, Madrid: Taurus.
- RODRIGUEZ ADRADOS, A. (2004) *Firma electrónica y documento electrónico*. Colegios Notariales de España, Madrid.
- VELAZQUEZ BAUTISTA, R. (2001). *Derecho Tecnologías de la Información y comunicaciones* (TIC), 1ª edición, Colex.
- VELAZQUEZ BAUTISTA, R. (2004) 100 Interrogantes fundamentales en Derecho de Tecnologías de la Información y las Comunicaciones (TIC), Madrid: Colex.

7.3 OTROS RECURSOS

En esta asignatura, el alumnado dispone de un guión de la asignatura y de un libro de texto, del que se ayudan en las clases de teoría. Además, dispone en el campus virtual de todas las actividades que deben realizar y las transparencias utilizadas en clase como esquema del contenido del tema de que se trate. Pero, además de estos recursos básicos, debemos incluir otros recursos de especial interés, tanto para el profesorado como para el alumnado.

Publicaciones periódicas:

- Informática y derecho: revista iberoamericana de derecho informático. Universidad Nacional de Educación a Distancia (España) Centro Regional de Extremadura
- Revista Aranzadi de Derecho y Nuevas Tecnologías
- Revista Electrónica de Derecho Informático
- Enredando: <http://www.noticias.juridicas.com/links/links.php?URL=http%3A%2F%2Fenredando.com>

Asociaciones y grupos de interés:

- Asociación Española de Derecho de las Telecomunicaciones y de las Tecnologías de la Información: <http://www.noticias.juridicas.com/links/links.php?URL=http%3A%2F%2Fwww.adeti.es>
- Centro Informático Científico de Andalucía (CICA): <http://www.noticias.juridicas.com/links/links.php?URL=http%3A%2F%2Fwww.cica.es>
- Comisión del Mercado de las Telecomunicaciones (CMT): <http://www.noticias.juridicas.com/links/links.php?URL=http%3A%2F%2Fwww.cmt.es>
- Comité Europeo de Normalización Electrotécnica (CENELEC): <http://www.noticias.juridicas.com/links/links.php?URL=http%3A%2F%2Fwww.cenelec.be>

Páginas web de interés:

Algunas páginas Web muy interesantes, que proporcionan enlaces o recopilaciones de enlaces a recursos jurídicos (organizaciones, revistas electrónicas, conferencias, bibliografía, etc.) son las siguientes:

- Centro de Documentación Europea: <http://www.cde.ua.es/eur/so.htm>
- Comisión Europea: http://europa.eu.int/information_society/index_es.htm: http://europa.eu.int/pol/infso/index_es.htm
- Gobierno Valenciano: <http://www.gva.es>: <http://www.cidaj.gva.es>
- Boletín Oficial del Estado: <http://www.boe.es>
- Tribunal Constitucional: <http://www.tribunalconstitucional.es>
- Agencia de Protección de datos: <http://www.agpd.es>
- Base de datos Aranzadi: <http://www.ua.es/es/bibliotecas/index.html>
- Delitos informáticos: <http://www.mir.es>: <http://www.delitosinformaticos.com>
- Portal Internacional de la Universidad de Alicante sobre Propiedad Industrial e Intelectual y Sociedad de la Información <http://www.uaipit.com/>
- Ministerio de Industria: <http://www2.mityc.es/dgdsi/>
- Ministerio de Administraciones Públicas: http://www.administracion.es/portadas/perfiles/administracion_electronica/
- <http://www.portaley.com/>
- <http://www.contratosinformaticos.com>
- <http://www.ua.es/es/servicios/juridico/informatica.htm>

Por otra parte, cabe destacar que actualmente el correo electrónico puede considerarse un recurso docente ya que se hace uso de él para resolver dudas sobre la asignatura. La mayoría de las veces es fácil la resolución de dudas de esta forma. No obstante, cuando la duda es de compleja resolución, es preferible la asistencia a tutorías ya que es la forma de asegurarnos que se ha entendido la explicación.

8. EVALUACIÓN DE LOS PROCESOS Y RESULTADOS DE APRENDIZAJE. SISTEMA DE EVALUACIÓN

8.1 PROCEDIMIENTOS DE EVALUACIÓN

Dado que se trata de una asignatura con un contenido teórico en su totalidad, tanto para realizar la metodología docente como para evaluar al alumnado, se intenta hacer una aproximación cercana a la forma de evaluación continua mediante las distintas actividades propuestas a lo largo del curso con la finalidad de adaptarse en la medida posible a las exigencias del EEE. Para ello, se propone dos métodos de evaluación: uno, dirigido a aquellos alumnos que asistan a clase regularmente, y otro, para los alumnos que realicen el estudio de la asignatura de forma no presencial absolutamente. De esta manera, a los alumnos que

decidan asistir a las clases teóricas programadas y realizar las actividades en grupo, se les realizará un seguimiento de los conocimientos adquiridos, de la realización de los ejercicios prácticos y tests de autoevaluación, en el horario de clase, así como de la actitud mostrada en las clases. Atendiendo a esto, la nota de la asignatura se desglosa de la siguiente forma:

NOTA DE RELACIONES JURÍDICAS BÁSICAS	
Ejercicios de los temas 1, 2, 3 y 4	30%
Ejercicios de los temas 5, 6 y 7	30%
Ejercicios de los temas, 8, 9 y 10	20%
Test de autoevaluación del tema 5 y 6	5%
Test de autoevaluación del tema 7	5%
Test de autoevaluación del tema 8 y 9	5%
Test de autoevaluación del tema 10	5%

Cuadro 8. *Desglose de la nota de la asignatura*

Siempre que la nota obtenida con los trabajos anteriores sea mayor o igual que 5, podrá ser aumentada hasta en un punto atendiendo a la actitud presentada en la asignatura: se valorará entre otros, la asistencia a clase y a las actividades en grupo, llevar el trabajo al día, presentación de las prácticas entregadas y realización de los tests de autoevaluación. Los alumnos que no superen un test de evaluación determinado tendrán opción de hacer recuperación en el siguiente. Si la nota de los test es inferior a 5, se les aplicará el sistema de los alumnos que elijan la asignatura no presencial en su totalidad.

Los alumnos que no asistan a clase deberán hacer un examen final y entregar todas las prácticas y actividades realizadas a lo largo del curso al comienzo del mismo. La nota del examen tendrá un peso del 80% y la del resto de actividades un 20%. Pero sólo se superará la asignatura si en ambas partes se obtiene una nota mayor o igual que 5 sobre 10.

8.2 CRITERIOS DE EVALUACIÓN

La calificación se hará de acuerdo a las siguientes pautas:

Calificación	rango	Pautas
Sobresaliente	9,0-10	<ul style="list-style-type: none"> • El conocimiento sobre la asignatura es profundo y se extiende más allá del trabajo cubierto por el programa. • La comprensión conceptual es sobresaliente. • Los problemas y procedimientos estadísticos relacionados con la asignatura son resueltos con eficiencia y precisión; los procedimientos estadísticos y de resolución de problemas se ajustan a la naturaleza del problema. • Las destrezas experimentales son ejemplares y muestran un completo análisis y evaluación de los resultados. • La actuación en las destrezas transferibles es generalmente muy buena. • La participación en las clases y distintas actividades ha sido muy correcta y muy satisfactoria.
Notable	7,0-8,9	<ul style="list-style-type: none"> • El conocimiento sobre estadística cubre de manera satisfactoria el programa. • La comprensión conceptual es notable. Los problemas y procedimientos jurídicos son resueltos cómodamente con buenos resultados. • La actuación en las destrezas establecidas es generalmente buena. • La participación en las clases y distintas actividades ha sido correcta y bastante satisfactoria.
Aprobado	5,0-6,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso es básico. • Los problemas jurídicos planteados relacionados con la asignatura son generalmente resueltos de forma adecuada. • Las destrezas se han conseguido con un nivel básico. • La participación en las clases y distintas actividades ha sido correcta pero no siempre satisfactoria.
Suspenso	0,0-4,9	<ul style="list-style-type: none"> • El conocimiento y la comprensión del contenido cubierto en el curso no ha sido aceptable. • Los problemas jurídicos relacionados con la asignatura no son, generalmente, resueltos de forma adecuada. • Las destrezas que se han transmitido por el profesor se han adoptado con un nivel deficiente. • La participación en las clases y distintas actividades ha sido escasa y deficiente.

Cuadro 9. Criterios de evaluación

9. ANÁLISIS DE COHERENCIA DE LA GUÍA DOCENTE

En la siguiente tabla se presenta el análisis de coherencia de la guía docente de Relaciones Jurídicas Básicas. Se han relacionado los objetivos y competencias con los bloques de contenido, el plan de trabajo de propuesto para el alumnado y el sistema y el criterio de evaluación.

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI1	CIC1, CIC2 cCIC1 cCIC2	1, 2, 3, 4			Enseñanza presencial (lección magistral /Trabajo de aula en grupo). Enseñanza no presencial (aprendizaje campus virtual, estudio personal/realización de ejercicios y prácticas propuestas). Tutorías individualizadas.	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de comprensión, y análisis de las fuentes del derecho y el ordenamiento jurídico-administrativo.
OI2 OI3	CIC3 cCIC1 cCIC2	5			Enseñanza presencial (lección magistral /trabajo de aula en grupo). Enseñanza no presencial (aprendizaje autónomo y on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas	<u>Procedimientos:</u> Actividades en grupo. Test de autoevaluación. <u>Criterios:</u> Grado de comprensión, interpretación, análisis y aplicación de los conceptos relativos a la informática jurídica y derecho de la informática

Cuadro 10. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI5	De CIC5 a CIC11 cCIC1 cCIC2		6,7	8,9,10	Enseñanza presencial (lección magistral trabajo de aula en grupos). Enseñanza no presencial (estudio personal/realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Ejercicios propuestos Test de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual y aplicación del contenido normativo
OI6	De CIC5 a CIC9 cCIC1 cCIC2		6,7	8,9,10	Enseñanza presencial (lección magistral trabajo de aula en grupos). Enseñanza no presencial (estudio personal/realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Ejercicios propuestos y test de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual y aplicación del contenido normativo.
OI7	De CIC4 a CIC11		6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje autónomo y on-line/realización de ejercicios y prácticas propuestos). Tutorías individualizadas.	<u>Procedimientos:</u> Ejercicios propuestos y test de autoevaluación. <u>Criterios:</u> Grado de comprensión conceptual y aplicación del contenido normativo.

Cuadro 11. Análisis de coherencia: competencias instrumentales (saber)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Instrumentales (saber hacer)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
OI4	CIT1 CIL1	1,2,3,4	5		Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje autónomo).	<u>Procedimientos:</u> Ejercicios en grupo. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
OI6	De CIM1 a CIM5 CIL1 cCIL1		6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje autónomo/ realización de ejercicios y prácticas propuestos). Tutorías individualizadas	<u>Procedimientos:</u> Test de evaluación. Actividad en grupo. <u>Criterios:</u> Grado de comprensión conceptual y aplicación del contenido normativo
OI7	De CIM1 a CIM5 CIL1 cCIL1		6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupos). Enseñanza no presencial (aprendizaje autónomo/realización de ejercicios y prácticas propuestos). Tutorías individualizadas	<u>Procedimientos:</u> Test de evaluación. Actividad en grupo. <u>Criterios:</u> Grado de comprensión conceptual y aplicación del contenido normativo
OI8	CIT1 CIL1	1,2,3,4	5		Enseñanza presencial (lección magistral). Enseñanza no presencial (aprendizaje autónomo).	<u>Procedimientos:</u> Ejercicios en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.

Cuadro 12. Análisis de coherencia: competencias instrumentales (saber hacer)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Interpersonales (ser y estar)	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
cOIP1	CIPTC1 cCIPTC1 cCIPTC2	1,2,3,4	5,6,7	8,9,10	Enseñanza presencial (trabajo de aula en grupos) Enseñanza no presencial (realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP2	cCIPTR1 cCIPTR2 cCIPTR3	1,2,3,4	5,6,7	8,9,10	Enseñanza presencial (trabajo de aula en grupo) Enseñanza no presencial (realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos.
cOIP3	cCIPTR4	1,2,3,4	5,6,7	8,9,10	Enseñanza presencial (trabajo de aula en grupo) Enseñanza no presencial (realización de ejercicios y prácticas propuestos).	<u>Procedimientos:</u> Actividades en grupo. <u>Criterios:</u> Grado de destreza en trabajos participativos

Cuadro 13. Análisis de coherencia: competencias interpersonales (ser y estar)

OBJETIVOS GENERALES	COMPETENCIAS ESPECÍFICAS	BLOQUES DE CONTENIDOS			PLAN DE TRABAJO DE LOS ALUMNOS	PROCEDIMIENTOS Y CRITERIOS DE EVALUACIÓN
	Sistémicas	Bloque 1 (temas)	Bloque 2 (temas)	Bloque 3 (temas)		
cOS1	cCS1 cCS4 cCS5		5,6,7	8,9,10	Enseñanza presencial (lección magistral / trabajo de aula en grupo. Enseñanza no presencial (aprendizaje personal/ realización de ejercicios y prácticas propuestas). Tutorías individualizadas	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS2	cCS2 cCS3		5,6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupo. Enseñanza no presencial (aprendizaje personal/realización de ejercicios y prácticas propuestas). Tutorías individualizadas	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación del contenido normativo aplicado
cOS3	cCS1 cCS2 cCS4 cCS5		5,6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupo. Enseñanza no presencial (aprendizaje personal/realización de ejercicios y prácticas propuestas). Tutorías individualizadas	<u>Procedimientos:</u> Prácticas. Trabajo complementario. <u>Criterios:</u> Nivel de actuación en las destrezas transferibles.
cOS4	cCS2		5,6,7	8,9,10	Enseñanza presencial (lección magistral /trabajo de aula en grupo. Enseñanza no presencial (aprendizaje personal/realización de ejercicios y prácticas propuestas). Tutorías individualizadas	<u>Procedimientos:</u> Actividades en grupo. Trabajo complementario. <u>Criterios:</u> Grado de análisis y evaluación del contenido normativo aplicado..

Cuadro 14. Análisis de coherencia: competencias sistémicas

INDICE DE AUTORES

Álvarez Sánchez, Rafael	59
Arques Corrales, Pilar	141
Cachero Castro, Cristina	113-293
Calera Rubio, Jorge	217
Cantó López, M ^a Teresa	439
Corbí Bellot, Antonio-M.	187
Ferrández Rodríguez, Antonio	327
Forcada Zubizarreta, Mikel L.	217
Fuentes Gómez, Pedro	383
Fuster Guillo, Andrés	89
Gallardo López, Domingo	241
García Rodríguez, José	89
Garrido Alenda, Alicia	187
Gil Méndez, David	383
Gómez Ballester, Eva	113
Grediaga Olivo, Ángel	163
Luján Mora, Sergio	327
Martínez Barco, Patricio	113
Micó Andrés, María Luisa	217
Migallón Gomis, Violeta	59
Molina Carmona, Rafael	141
Montoyo Guijarro, Andrés	113
Mora Gimeno, Francisco José	409
Mora Mora, Higinio	409
Moreda Pozo, Paloma	113
Muñoz Guillena, Rafael	113
Oncina Carratalá, José	187
Palomar Sanz, Manuel	113
Penadés Martínez, José	59
Peral Cortés, Jesús	327
Pomares Puig, Cristina	241
Rico Soliveres, Llüisa	163
Rubio Serna, Ramón	187
Saiz Noeda, Maximiliano	9
Saquete Boró, Estela	293
Serra Pérez, José Antonio	89
Soriano Payá, Antonio	89, 163
Suárez Cueto, Armando	113
Trujillo Mondéjar, Juan Carlos	113
Vázquez Pérez, Sonia	113
Verdú Más, José Luis	217
Viché Clavel, Ignacio	141

