

UNA EXPERIENCIA DOCENTE DE “TÉCNICAS DE DESARROLLO SOFTWARE”

Edmundo Tovar, Andrés Silva, Nelson Medinilla, Miguel García Cordero, Elías Chaves

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software
Facultad de Informática
Universidad Politécnica de Madrid
e-mail: etovar@fi.upm.es

RESUMEN

En esta ponencia se presentan los resultados de la experiencia docente en la asignatura de Sistemas de Información en la Facultad de Informática de la UPM, exponiendo cuál es la programación de contenidos, y la problemática surgida y la solución propuesta ante la decisión de incorporar una práctica en máquina como elemento fundamental en el aprendizaje y evaluación de un grupo numeroso de alumnos matriculados.

1. La construcción del software desde el punto de vista del desarrollo

El proceso de la Ingeniería del Software incluye un amplio rango de actividades realizadas por los ingenieros software, que se pueden clasificar en cuatro grandes grupos: de desarrollo, de control, de gestión y de operación. En el vigente Plan de Estudios de la Facultad de Informática de la Universidad Politécnica de Madrid, la enseñanza de las actividades de desarrollo, a través de las cuales se crean los productos que componen el sistema software, se concentra en la asignatura de Sistemas de Información. Esta asignatura, obligatoria y correspondiente al 4º curso, se ocupa de las técnicas utilizadas durante las fases de desarrollo en el ciclo de vida de un sistema software. Es decir, planteando la construcción del software como un caso más de resolución de problemas, en el que debe existir un primer paso en el que se define el qué (análisis y especificación de requisitos), el cómo hacerlo, diseño del sistema software, y a la realización de ese cómo o codificación. No se considera la fase de codificación, pues ya se trata en Entornos de Programación y en otras asignaturas del actual Plan de Estudios. Mientras, tanto las actividades de control como de gestión se abordan en las asignaturas de 6º curso, Planificación de Sistemas Informáticos e Ingeniería de Software.

El desarrollo de la docencia y de la programación de un curriculum para esta asignatura se centra en el uso y aprendizaje de técnicas, aunque no se puede olvidar que éstas están, normalmente, dirigidas y propuestas por la decisión de seguir una metodología. En este curso, se han considerado, entre otras, las de Yourdon, de Marco, Warnier, Jackson, y Coad y OMT, entre las Orientadas a Objetos.

Tema 3. Diseño Estructurado

Objetivos: En este tema se pretende que el alumno conozca las técnicas y estrategias más habituales para conseguir un buen diseño, teniendo en cuenta los conceptos de cohesión y acoplamiento, a partir del análisis estructurado realizado en la etapa previa.

Subtemas:

- 3.1. Conceptos fundamentales de diseño
- 3.2. Técnicas de diseño
- 3.3. Estrategias de diseño: análisis de transformación y de transacción
- 3.4. Guías de diseño
- 3.5. Calidad: acoplamiento y cohesión

Horas de clase: 12 horas de clase

Tema 4. Arquitecturas

Objetivos:

Aquí se resalta la importancia de ser capaz de reconocer paradigmas comunes de alto nivel. En este tema se pretende permitir elegir al Ingeniero Software, a través del conocimiento detallado de distintas arquitecturas y sus tipos de representaciones, el diseño adecuado al sistema que se quiere desarrollar.

Subtemas:

- 4.1. Evolución histórica
- 4.2. Estilos de arquitectura software

Horas de clase: 2 horas

Tema 5. Análisis y Diseño OO

Objetivos:

El tema de orientación a objetos dentro de la asignatura actual de Sistemas de Información debe ser introductorio por dos razones fundamentales. Es la primera vez que los alumnos se enfrentan con la orientación a objetos, aunque antes hayan visto temas relacionados. La forma de pensar que exige la orientación a objetos es básicamente distinta a todas las anteriores y, por el momento, le corresponde a esta asignatura iniciar este nuevo enfoque.

Dada esta situación, tiene mucho más sentido lograr una aproximación intuitiva que exponer una metodología en concreto. No obstante, se utiliza como notación principal la proporcionada por OMT, con la finalidad de hacerla familiar. Han surgido problemas mezclando OMT con la de Coad. Ambas metodologías tienen enfoques muy distintos (o radicalmente distintos) y la mayoría de los alumnos pueden confundirse con la polémica con los distintos autores. Sin embargo, la metodología OMT ha permitido utilizarla como esqueleto del primer parcial. A pesar de que es muy discutible el papel del modelo funcional dentro de una metodología OO, de esta forma es posible explicar, dentro de un mismo marco, las tres perspectivas fundamentales del análisis (estática, dinámica y funcional).

Por lo tanto, desde el punto de vista docente, resulta muy recomendable.

Subtemas:

- 5.1. Conceptos fundamentales
- 5.2. Modelización OO en análisis y diseño: metodologías de Coad y OMT

Horas de clase: 10 horas

Tema 6: Diseño de bajo nivel.

Dentro del ciclo de vida de un Sistema de Información, o por defecto, de cualquier producto software, existe un puente, que las nuevas metodologías, técnicas y herramientas pretenden inconscientemente soslayar, y que enlaza y relaciona el Análisis Funcional y Detallado con las Pruebas. Esta es la parte conocida como Diseño de Bajo Nivel (más comúnmente conocida como Programación y Codificación)

2. Organización y programación de contenidos

Los contenidos desarrollados en dicha asignatura están agrupados en cinco grandes bloques que, con excepción del primero, que es introductorio al proceso software, se corresponden con las etapas mínimas necesarias para resolver el problema de la construcción de un sistema software. Es decir, la obtención y especificación de sistemas software, el diseño, en dos niveles, de alto nivel o preliminar, y de bajo nivel o diseño detallado y, por último, la realización de pruebas. En concreto, éstas son:

1. Introducción al proceso de construcción de software
2. Ingeniería de Requisitos Software
3. Diseño de alto nivel
4. Diseño detallado o de bajo nivel
5. Verificación y Validación de software

No se realiza una práctica completa de principio a fin en máquina, pues ello ocupa una parcela importante de la asignatura de 6º curso de Ingeniería del Software. Sin embargo, la filosofía del curso pretende tener un marcado carácter práctico, a través de tres aspectos:

- a) Descripción de las técnicas a través de ejemplos.
- b) Realización de pequeños ejercicios fuera de clase, fundamentalmente prácticos.
- c) Desarrollo de una práctica en máquina de modelización orientada a objetos

A continuación se describen los temas en los que han programado los contenidos:

Tema 1. Introducción al Proceso Software

Objetivos:

En este tema se pretende proporcionar al alumno el contexto del curso: qué pretende la Ingeniería del Software, dónde se sitúa dentro de la Informática,...

Horas de clase: 6 horas.

Tema 2. Ingeniería de Requisitos Software

Objetivos:

Se resalta la idea de realizar todas estas tareas como parte de un equipo de trabajo, tomando conciencia del carácter multidisciplinar de los procesos de educación y de análisis del problema.

Se exponen una serie de técnicas, utilizadas por diversas metodologías, que pueden ayudar a llevar a cabo cada uno de los procesos que son necesarios realizar. Por tanto, es importante proporcionar los conocimientos suficientes para seleccionar la técnica más adecuada en cada caso, pero comprendiendo el modelo general que sea válido para todos los casos.

Con la impartición de este tema se trata de ser capaz de dividir un sistema en componentes y de aplicar métodos de especificación a cada uno de estos componentes.

Subtemas:

- 2.1. Introducción a la Ingeniería de Requisitos
- 2.2. Técnicas de educación de requisitos
- 2.3. Técnicas de modelización
- 2.4. El documento ERS

Horas de clase: 18 horas

cuyo contenido y cometido es desarrollar y adecuar los requisitos del usuario, ya transformados por el Análisis previo, a un nivel de detalle y profundidad, ya se está prácticamente al lado del ordenador, que resuelva y haga viable cada uno de los problemas presentados por todos y cada uno de los requisitos.

Para la realización de esta tarea es necesario, para no incurrir en los males endémicos que tiene la Ingeniería de Software, que exista una forma y modos de realizarla, que hagan legible el desarrollo del trabajo transcurrido un cierto tiempo. Esto se consigue mediante las metodologías aplicadas a este nivel de desarrollo. De las metodologías se ha hablado ya al principio de la materia de este curso y no se va a insistir en su contenido, solamente se insistirá en que el alumno, al final de este tema, estaría, tanto en un plano técnico como psicológico, con la capacidad de:

- Explicar en qué consisten las metodologías presentadas resumiendo sus características fundamentales, aplicar los conceptos y las abstracciones del método correspondiente a casos concretos, identificar los elementos esenciales de las metodologías, de cada una de las técnicas y las interrelaciones entre ellas, señalando la conveniencia o no de su uso en cada caso, y por último, estudiar una nueva metodología por comparación con otra ya conocida, siempre y cuando sean de la misma orientación (datos, procesos, objetos, etc.)
- Evaluar la metodología y las técnicas expuestas u otras que conozca o estudie en el futuro, enjuiciando su aplicación a su propio entorno.
- Poder apreciar la importancia del empleo de técnicas en la obtención de los productos software.
- Encontrarse dispuesto a apoyar la implantación y utilización de técnicas y metodologías para la programación como lenguaje común entre los diseñadores de software.

Subtemas:

- 6.1. Introducción (Repaso de conceptos de programación estructurada)
- 6.2. Y, para cada metodología (Warnier, Jackson, Yourdon y tradicional):
 1. Descripción de cada metodología y sus bases teóricas
 2. Metodología básica
 3. Controles
 4. Tipos de datos y su tratamiento
 5. Estructuras de entrada, salida y de proceso y sus interrelaciones
 6. Estudio de sistemas más complejos
 7. Comparación con las otras metodologías

Horas de clase:

Toda esta teoría, acompañada de su práctica correspondiente ha sido preparada para un total de 20 horas.

Tema 7. Verificación y Validación del software

Objetivos:

En este tema se pretende comprender los conceptos generales de la prueba y entender que la prueba total es imposible, comprender y distinguir los diferentes tipos de pruebas que se deben realizar al software, y cuándo se deben realizar y, entender los diferentes entornos para la realización de la prueba y comprender las diferentes funciones de los especialistas de pruebas.

Subtemas:

- 7.1. Introducción, definición y conceptos fundamentales
- 7.2. Verificación y Validación en el proceso de evaluación. Métodos de evaluación
- 7.3. Pruebas de caja blanca y caja negra
- 7.4. Otros tipos de pruebas: de requisitos de Diseño de código, Inspecciones del código, Walkthroughs, revisiones de mesa, evaluación de pares, Pruebas con uso de la computadora: de unidad, integración y del sistema
- 7.5. Conclusiones

Horas de clase: 20 horas.

3. Criterios de evaluación

La evaluación de cada alumno son resultado de la consideración de las evaluaciones de los dos exámenes parciales, la evaluación de la practica en máquina, y la realización de los ejercicios propuestos fuera de clase. Estos ejercicios, son propuestos por cada profesor, al menos uno por tema, fuera de clase. Son realizados por grupos, *obligatorios*, y su entrega requisito indispensable, por escrito y con el formato propuesto, para presentarse a los exámenes. La calificación de estos ejercicios sirve para mejorar los resultados de aprobado de los exámenes correspondientes. Es decir, la nota de los ejercicios fuera de clase se suma a la calificación obtenida en el examen, siempre y cuando éste resulte aprobado. Sólo en el tema de Verificación y Validación, el ejercicio propuesto, y por razones de indisponibilidad de tiempo, al impartirse en las fechas previas a los exámenes de junio tiene carácter optativo.

En concreto, el primer parcial comprende los temas 1, 2, 3 y S. Su evaluación se compone, con el mismo peso, de un examen teórico de los contenidos recogidos en las clases impartidas, y de la entrega y presentación y defensa ante uno de los profesores, el tutor, de la práctica en máquina de modelización orientada a objetos.

El segundo parcial, comprende los temas 4, 6 y 7. Teniendo en cuenta que el potencial de alumnos a matricularse en esta asignatura puede ser muy elevado, que no es posible hacer un seguimiento detallado de todos ellos, y con el fin de ser lo más objetivo posible, se deben realizar a todos los alumnos dos tipos de evaluación en las fechas señaladas para los exámenes correspondientes: - una de tipo teórico, en forma de test, con una batería de cuestiones proporcional al tiempo asignado al tema. - otra de tipo práctico que puede consistir en la resolución de un supuesto problema adecuado al tiempo asignado, o la corrección de una solución propuesta a un problema. Se cree que de esta forma se puede evitar la subjetividad debida a las interrupciones en las correcciones por la gran cantidad de ejercicios a corregir. En el tema de diseño de bajo nivel se plantearon dos ejercicios fuera de clase: una relativa a la teoría de recursos abstractos y otra, relativa a la aplicación del contenido del apartado 6. Estas dos prácticas se hicieron en grupo, normalmente de 4 o 5 personas, aunque los había de 1, 2 ó 3 pero en menor número. Los grupos fueron 94 que representaban un colectivo de 419 alumnos. Como los resultados más significativos e importantes son los del segundo ejercicio, nos referiremos a ella solamente.

Prácticamente la totalidad de los grupos la entregaron en la fecha señalada. En los aspectos de forma lo han hecho muy bien. En los aspectos de fondo, en una primera revisión fueron admitidas 47, necesitaban información adicional, 38 y el resto no eran admitidas si no eran completamente revisadas.

A este respecto cabe señalar que el 60 por 100 de los grupos hicieron consultas, con un promedio de 2,5 consultas por grupo, aunque la mayor parte de ellas se produjeron en fechas cercanas a la fecha tope de entrega, que la gran mayoría de las consultas se hicieron entre el 24 de Abril y la fecha tope indicada. Durante este último período, las consultas se producían durante toda la jornada sin respetar el horario de tutorías.

4. La práctica en máquina

Tradicionalmente, quizás por falta de recursos, en esta asignatura no se había realizado ninguna práctica en máquina. Sin embargo, considerando cuál es el futuro profesional de gran parte de los alumnos que reciben formación en esta Facultad no parecía conveniente que el conocimiento y uso de una herramienta de desarrollo se postergara hasta el último año de carrera. Ésta ha sido la razón por la que, en este curso y por primera vez, se haya introducido una experiencia de este tipo.

La práctica de la asignatura consistió en el desarrollo de un modelo, orientado a objetos, de un sistema informatizado para un tipo de problema que fuera conocido por todos. Este punto quedó enteramente resuelto, proponiendo una práctica anterior a ésta, en la que requería entregar un documento de requisitos software, utilizando una técnica de educación y con la misma formación en grupos, en el mismo dominio: en este caso, la automatización de una empresa de alquiler de coches. El modelo se restringía, en esencia, a la fase de análisis y debía ser realizado en una herramienta de desarrollo software. Para ello, se contaba con dos herramientas de desarrollo orientada a objetos, una en entorno Unix, con 10 licencias disponibles, y otra en entorno de PC, con una licencia de red e instalada en el Centro de Cálculo. Desde un primer momento se buscaba no depender de una única herramienta, ni proporcionar una visión restringida o monopolística de las herramientas disponibles en el mercado. Para la gran mayoría de alumnos, esta práctica en máquina ha resultado ser su primer contacto con el mercado de herramientas comerciales de desarrollo software, por lo que este tipo de aspectos deben ser tenidos en cuenta (la primera experiencia suele impactar en mayor medida que el resto).

Por supuesto, se contaba con el soporte del Centro de Cálculo de este Centro, para poder dar servicio a casi 100 grupos de entre 3 y 5 personas que se formaron. El tiempo estimado de uso de la máquina, entre familiarización con el entorno y desarrollo de la práctica era de 15 horas, tiempo distribuido entre todos los distintos terminales, con diferentes horarios, a lo largo, principalmente, de dos meses.

La práctica se hizo por grupos de tres a cinco miembros. En total fueron casi cien grupos. A partir de esta primera premisa, nuestra primera preocupación fue la de gestionar el volumen de trabajo de los alumnos. Los tres profesores disponibles para tal tarea eran claramente insuficientes. Ello dió lugar a buscar otras vías complementarias de recursos humanos. Aquí aparecen las figuras de monitor y voluntarios cuyas responsabilidades quedan expuestas a continuación.

En primer lugar, los monitores se han encargado de apertura de cuentas para cada grupo y de dar soporte a sus compañeros en las salas de máquinas durante las horas asignadas a la asignatura, sirviendo de consultores sólo respecto al manejo de las herramientas de desarrollo. Pero para cumplir con esta misión han necesitado una formación con una doble vertiente: fue necesario prepararlos con antelación desde el punto de vista teórico y práctico. De la teoría se ocuparon los profesores de la asignatura, anticipadamente al resto de sus compañeros, y de la práctica con las herramientas de desarrollo, tres estudiantes de cursos superiores, los voluntarios, que se habían entrenado previamente. Estos monitores prepararon, durante su formación y como parte de ella, unas pequeñas guías de uso de las herramientas que se pusieron a disposición de todos los estudiantes. Además, se encargaron de impartir los conocimientos básicos acerca de las herramientas al sus compañeros, y, por último, recopilaron las dudas más frecuentes de los alumnos que se utilizarán en los próximos cursos.

En total, y para apoyar el trabajo en máquina de esta cantidad de grupos, se prepararon quince monitores, escogidos de entre los mismos estudiantes que cursaban la asignatura.

A cada uno de los grupos de prácticas se les asignó un tutor de prácticas de entre los tres profesores. Ello permitió un contacto más directo con los alumnos, que de otra forma, hubieran pasado inadvertidos entre los 473 matriculados en esta asignatura. Por eso, el resultado, probablemente, más importante de la práctica fue su acción sobre el aprendizaje de los alumnos, sobre todo, durante las tutorías y las

discusiones con cada uno de los grupos en la evaluación particularizada que se realizó. A cambio, el coste ha sido alto:

- La organización, preparación y control de los monitores fue una parte de ese coste.
- La otra, fue el tiempo empleado en la atención directa a los alumnos en las consultas y evaluación de los trabajos, dado el volumen de grupos. Cada profesor atendía unos treinta y tantos grupos de tres a cinco estudiantes.
- Y, por qué no decirlo, en el esfuerzo suplementario consecuencia directa de nuestra inexperiencia para gestionar tanta información. A esto habría que sumar problemas de explotación de las herramientas software, su instalación y uso, obras en el Centro de Cálculo en los plazos de tiempo de realización de la práctica,...

5. Conclusiones

Antes de nada, decir que evaluar o extraer conclusiones de la docencia de la asignatura durante este curso no es posible hacerlo correctamente sólo con la opinión del profesor, sin tener en cuenta la del alumno. Si fuera posible, lo ideal sería preguntar a los alumnos, no hoy, sino dentro de cinco años, tras pasar por la experiencia laboral, qué es lo que les ha parecido la asignatura.

El problema inicial con que nos encontramos es que los alumnos no saben cómo enfrentarse con un problema real, y todavía, en esta asignatura, no ven el desarrollo de un proyecto completo (esto queda relegado a la asignatura de Ingeniería de Software de 6º curso). A través del curso se pretende aminorar este problema utilizando el mismo dominio de aplicación tanto en la práctica en máquina como en la mayor parte de ejercicios propuestos fuera de clase.

Como aspectos positivos cabe señalar, el gran trabajo desarrollado por los alumnos en la realización de las prácticas, así como en su presentación, la posibilidad de pasar de una metodología, de la misma orientación, a otra con gran rapidez, siempre que se haya profundizado bastante en la primera o en la que se tome de referencia.

Es muy difícil transmitir ciertas ideas a quien no tiene una mínima experiencia de trabajo en desarrollo Software. Esta asignatura es eminentemente práctica, la más cercana a lo que luego será su trabajo profesional, y sin embargo no puede reducirse a la enseñanza y uso de unas herramientas marginando los fundamentos teóricos que lo soportan. Las prácticas están aún supeditadas a la teoría, donde realmente los alumnos aprenden. Esto supone que, o con la teoría no aprenden, o creen que no aprenden. Más bien esto último. Los docentes de esta asignatura históricamente nos hemos dejado conquistar gran parte del terreno. En la preocupación del alumno pesan mucho más otras asignaturas que realmente no tienen nada que ver (o muy poco) con las demandas de trabajo actuales, o con su trabajo futuro. Por tanto, la idea que se debe transmitir, pensamos, es que cuando se enfrenten a la construcción de grandes y complejos sistemas software sean capaces de afrontarlo con la seguridad de quien sabe que posee los conocimientos adecuados para tal tarea. Aquí ya no se trata de programar algoritmos, sino de construir sistemas, y no debe cundir el pánico cuando se vean ante ello. La complejidad puede domarse.

Además, dado el intrusismo existente en esta profesión, habría que tomar conciencia de que los conocimientos impartidos en la asignatura son los que harán de ellos profesionales mejor cualificados que los procedentes de otros campos de la ingeniería. Para ello contribuye el motivarles en el manejo de bibliografía. Especialmente en una materia como ésta, en la que no todo el mundo está de acuerdo en tantas y tantas cuestiones clave. El espíritu crítico surgirá de lo que lean, y el espíritu crítico es lo que distingue a la enseñanza universitaria de una formación puramente profesional.