



Universitat d'Alacant
Universidad de Alicante

Memorias del Programa de Redes de investigación en docencia universitaria

Convocatoria
2021-22

Memòries del Programa de Xarxes de investigació en docència universitària

Convocatòria
2021-22

Satorre Cuerda, Rosana (Coordinación)
Menargues Marcilla, María Asunción; Díez Ros, Rocío; Pellín Buades, Neus (Eds.)

UA

UNIVERSITAT D'ALACANT
UNIVERSIDAD DE ALICANTE
Vicerectorat de Transformació Digital
Vicerrectorado de Transformación Digital
Institut de Ciències de l'Educació
Instituto de Ciencias de la Educación

Memorias del Programa de Redes de investigación en docencia universitaria. Convocatoria 2021-22 / Memòries del Programa de Xarxes d'investigació en docència universitària. Convocatòria 2021-22

Organització: Institut de Ciències de l'Educació (Vicerectorat de Transformació Digital) de la Universitat d'Alacant/ Organización: Instituto de Ciencias de la Educación (Vicerrectorado de Transformación Digital) de la Universidad de Alicante

Edició / Edición: Rosana Satorre Cuerda (Coord.), Asunción Menargues Marcilla, Rocío Díez Ros, Neus Pellin Buades

Revisió i maquetació: ICE de la Universitat d'Alacant/ Revisión y maquetación: ICE de la Universidad de Alicante

Primera edició / Primera edición: desembre 2022

© De l'edició/ De la edición: Rosana Satorre Cuerda, Asunción Menargues Marcilla, Rocío Díez Ros & Neus Pellin Buades

© Del text: les autores i autors / Del texto: las autoras y autores

© D'aquesta edició: Universitat d'Alacant / De esta edición: Universidad de Alicante

ice@ua.es

Memorias del Programa de Redes de investigación en docencia universitaria. Convocatoria 2021-22 / Memòries del Programa de Xarxes d'investigació en docència universitària. Convocatòria 2021-22

© 2022 by Universitat d'Alacant / Universidad de Alicante is licensed under CC BY-NC-ND 4.0

ISBN: 978-84-09-45382-5

Qualsevol forma de reproducció, distribució, comunicació pública o transformació d'aquesta obra només pot ser realitzada amb l'autorització dels seus titulars, llevat de les excepcions previstes per la llei. Adreceu-vos a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necessiteu fotocopiar o escanejar algun fragment d'aquesta obra. / Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sólo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra.

Producció: Institut de Ciències de l'Educació (ICE) de la Universitat d'Alacant / Producción: Instituto de Ciencias de la Educación (ICE) de la Universidad de Alicante

Aquesta publicació s'ha fet seguint les directrius d'accessibilitat UNE-EN 301549:2020 / Esta publicación se ha hecho siguiendo las directrices de accesibilidad UNE-EN 301549:2020.

EDITORIAL: Les opinions i continguts dels treballs publicats en aquesta obra són de responsabilitat exclusiva de les autores i dels autors. / Las opiniones y contenidos de los trabajos publicados en esta obra son de responsabilidad exclusiva de las autoras y de los autores.

13. INNOVACIÓN Y APLICACIÓN DE METODOLOGÍAS DE ENSEÑANZA DE PROGRAMACIÓN Y DISEÑO DE AUTOMATISMOS

Jorge R. Pérez Beltrán¹; Ramón P. Neco García²; Nicolás M. García Aracil²; José M. Catalán Orts²; Iván Fabra Ramón¹

jperez@dccia.ua.es ; ramon.neco@umh.es; nicolas.garcia@umh.es; jose.catalan@umh.es; ifrf2@alu.ua.es

¹ Escuela Politécnica Superior de Alicante, Universidad de Alicante

² Escuela Politécnica Superior de Elche, Universidad Miguel Hernández de Elche

Resumen (Abstract)

En el presente proyecto se han desarrollado y aplicado métodos y estrategias de enseñanza de la programación a estudiantes para los que no es una materia básica o central para su formación. En estos métodos se combinan diferentes paradigmas como la enseñanza de programación estructurada y programación orientada a objetos, así como técnicas de gamificación. La metodología se enfoca en la adquisición de las destrezas o capacidades de abstracción en la formalización y resolución de problemas, que son imprescindibles para el futuro laboral del estudiantado en las titulaciones y niveles que han sido objeto del trabajo del proyecto. Las metodologías docentes diseñadas se han aplicado en asignaturas de grados de ingeniería del ámbito industrial y también han sido adaptadas al caso de la enseñanza de Informática en algunos cursos de Enseñanza Secundaria. Como resultado principal del trabajo realizado se ha podido comprobar que se ha obtenido una mejora cuantitativa en las calificaciones del estudiantado respecto de cursos previos, o de los bloques

en los que no se ha aplicado la metodología, y que se han asimilado de una manera más efectiva los conocimientos de estas materias, ayudando a mejorar la percepción negativa que en ocasiones se tiene de las mismas.

Palabras clave: Fundamentos de la programación, docencia de automatización industrial, técnicas de gamificación, formalización de problemas

1. Introducción

En el presente proyecto se propone y desarrolla una metodología para la enseñanza de conceptos de programación de ordenadores a estudiantes de grado para los cuales esta materia no es una materia básica o principal. La metodología docente empleada en el proyecto ha utilizado, entre otras, técnicas de gamificación para la docencia de programación de ordenadores (Chen, 2019). Las tareas implicadas en la metodología están inspiradas en problemas reales de ingeniería y se trata de motivar a los alumnos para conseguir que obtengan soluciones eficientes y eficaces.

Adicionalmente al trabajo anterior, en el proyecto también se ha adaptado la propuesta de metodología para la enseñanza de la programación informática en varios cursos de Educación Secundaria. En este caso, en la metodología propuesta se explora la posibilidad de combinar la docencia de la programación estructurada con la programación visual orientada a objetos con el fin de aprovechar los beneficios de ambos paradigmas, incluyendo el diseño de actividades y mecanismos para mejorar las habilidades de formalización del lenguaje natural utilizado en la resolución de problemas. En la implementación de la metodología utilizamos técnicas de gamificación para motivar y reforzar el aprendizaje, al igual que en el caso del nivel universitario.

1.1 Problema o cuestión específica del objeto de estudio

Los estudiantes de la mayoría de grados de ingeniería que no son especialistas en informática necesitan conocer conceptos de programación de ordenadores

como una herramienta básica para su futuro trabajo, a pesar de que esta materia no se considera básica o principal en sus estudios. En nuestro trabajo como docentes hemos detectado que los estudiantes de estos grados necesitan un refuerzo en los contenidos de programación de ordenadores y, especialmente, en su motivación para estudiar esta herramienta de la ingeniería.

Por esta razón, en el presente proyecto se plantea el problema del diseño y aplicación de una metodología para reforzar los conocimientos básicos de programación necesarios para estos estudiantes e involucrarlos en el aprendizaje de los conceptos de programación necesarios en sus futuros trabajos como ingenieros (Medeiros, 2019).

En el caso de la Educación Secundaria, se ha detectado que la mayoría de los estudiantes tienen problemas para comprender y especificar de manera formal los ejercicios y trabajos de laboratorio que normalmente se especifican en lenguaje natural. Además, la mayoría de los estudiantes son programadores novatos y por esta razón tienen dificultades para crear sus algoritmos para diferentes tipos de problemas.

1.2 Revisión de la literatura

Como se ha mencionado previamente, las asignaturas correspondientes a los fundamentos de la programación tienen una gran importancia para la vida y el futuro profesional de los estudiantes de ingeniería en general. El contenido de estas asignaturas, los objetivos y los métodos docentes han sido objeto de numerosos trabajos previos (Sobral, 2020). En numerosos trabajos de investigación en los que se estudia la metodología docente de estas asignaturas, se constata que el estudiantado a menudo tiene la percepción de que los docentes se centran más en el aprendizaje de la sintaxis del lenguaje de programación, llevado a centrarse en actividades de implementación en lugar de actividades tales como planificación o prueba, lo cual puede ser la causa del abandono de las asignaturas (Sobral, 2020b; Félix, 2020). Para superar estas dificultades se han propuesto métodos y estrategias que han sido ampliamente estudiados y han tratado, especialmente, la motivación de los estudiantes. Algunas de estas metodologías son las siguientes: metodologías activas (Bonwell, 1991; Hundhausen, 2007; Kao, 2008), basadas en proyectos aprendizaje (Blumenfeld, 1991), metodologías ágiles como SCRUM (Dybå,

2008), y gamificación y trabajos en grupos (Dori, 2005; McDowell, 2006; Pedersen, 2003).

En (Medeiros, 2019) se hace una revisión exhaustiva de la bibliografía relacionada con las principales metodologías usadas en fundamentos de programación y automatismos, relacionadas con la desarrollada en el proyecto. En la mayoría de estos trabajos previos se opta por realizar la docencia usando alguno de los paradigmas de programación de forma independiente (por ejemplo, programación estructurada, programación gráfica o programación orientada a objetos). Una de las principales novedades del trabajo realizado en el proyecto ha sido el uso combinado de paradigmas (programación estructurada y programación gráfica orientada a objetos), junto con el uso de técnicas de gamificación y trabajo en equipo.

1.3 Propósitos u objetivos

El propósito u objetivo principal del trabajo realizado en el proyecto, es la mejora en la motivación y el aprendizaje de los conceptos básicos de programación informática para estudiantes de grados de ingeniería no especialistas en informática y en Educación Secundaria, repercutiendo positivamente en el proceso de enseñanza-aprendizaje en estos contextos. En los resultados se pretende reflejar las opiniones recogidas de los alumnos tras finalizar el curso y, finalmente, extraemos nuestras conclusiones y resumimos las posibilidades de adaptar la metodología a otras materias y niveles educativos.

2. Método

2.1. Descripción del contexto y de los participantes

Enseñanza universitaria.

Los alumnos que han participado en las experiencias realizadas en el contexto del proyecto son alumnos de segundo curso del Grado en Ingeniería Mecánica y tercero del Grado en Ingeniería Eléctrica de la Universidad Miguel Hernández de Elche. En el primer caso, la materia objeto de estudio fue la “Teoría de

Sistemas” y en el segundo caso la “Automatización Industrial”.

En el curso “Teoría de Sistemas” (Grado en Ingeniería Mecánica) se estudian conceptos sobre modelado, análisis, diseño y simulación de sistemas de ingeniería, especialmente relacionados con sistemas mecánicos (Abelson, 1996). Esta materia tiene un fuerte contenido teórico en Matemáticas y Física (conceptos como transformada de Laplace, transformada de Fourier, transformada Z, ecuaciones diferenciales de sistemas físicos continuos, ecuaciones en diferencias de sistemas discretos, etc.) y, paralelamente, trabajos de laboratorio en los que se utilizan herramientas de simulación como MATLAB/Simulink y Scilab/Scicos. Para realizar las prácticas de análisis y simulación de sistemas con estas herramientas es imprescindible un conocimiento mínimo de sus correspondientes lenguajes de programación (por ejemplo, conceptos sobre variables, estructuras de control lógico, bucles, estructura de programas, etc.).

En la asignatura “Automatización Industrial” (Grado en Ingeniería Eléctrica) se estudian los fundamentos del diseño de sistemas automáticos usando controladores lógicos programables (PLC) (Schach, 2006). Esta materia tiene contenido teórico sobre estructuras de control automático utilizadas en la industria, tipos de PLC disponibles en el mercado, fundamentos sobre sensores y actuadores utilizados en automatización y robótica, control de procesos, diseño de control secuencial discreto de procesos y sus aplicaciones a la automatización de procesos industriales, componentes básicos de un sistema automatizado, metodologías de diseño de automatismos secuenciales y desarrollo de proyectos de automatización industrial. Una parte básica de este curso es la introducción a la programación de autómatas utilizando sus lenguajes de programación estándar (diagrama de escalera y lenguaje estructurado) y las operaciones básicas de estos lenguajes (Schach, 2006). En las sesiones de laboratorio, después de realizar el diseño de sistemas automáticos, los alumnos deben implementar los sistemas que han diseñado en PLCs reales disponibles en el laboratorio, por lo que es fundamental que tengan un buen conocimiento de cómo programarlos.

Por lo tanto, ambas materias son cursadas por estudiantes de ingeniería que no estudian Programación de Computadores como materia básica en su formación académica, pero para las cuales es fundamental que conozcan y comprendan los fundamentos de la programación estructurada.

Todo el estudiantado de estas materias ha estudiado previamente una

asignatura introductoria de Informática de 60 horas en su primer curso en la universidad. En esta asignatura se estudian conceptos sobre el uso y programación de computadoras, sistemas operativos, bases de datos y programas de cálculo con aplicaciones en ingeniería. Los contenidos de esta asignatura se dividen en sistemas de representación interna, arquitectura de ordenadores, sistemas operativos, introducción a las bases de datos e introducción a la programación. Tras superar esta asignatura, el estudiantado tiene una idea general sobre la Informática y sus aplicaciones en la ingeniería, así como una idea básica sobre la Programación de ordenadores utilizando el lenguaje de programación C o Python (según el curso académico). Sin embargo, como se ha comentado previamente, en nuestro trabajo como docentes hemos detectado que una vez que los alumnos completan el curso mencionado anteriormente, necesitan un refuerzo en los contenidos de programación de ordenadores y, especialmente, en su motivación para estudiar esta herramienta de la ingeniería.

En las experiencias realizadas en el contexto del proyecto en el nivel universitario han participado 122 estudiantes (102 de la asignatura “Teoría de Sistemas” y 20 de la asignatura “Automatización Industrial”).

Adaptación a la Enseñanza Secundaria

En el caso de la adaptación a la Enseñanza Secundaria la metodología se ha aplicado a las asignaturas optativas de Informática en los cursos de 3º y 4º de ESO y 2º curso de Bachillerato (estudiantes con edades entre los 14-15 y 15-16 años en ESO, y de 17-18 años en Bachillerato). En el proyecto se han realizado experiencias con la adaptación de la metodología en pequeños grupos de estudiantes en asignaturas como “Taller de videojuegos I”, “Informática”, “Taller de videojuegos II”, “Tecnologías de la Información y la Comunicación, TIC” y “Tecnologías de la Información y la Comunicación II, TIC II”. Un total de 61 estudiantes de secundaria han participado en las actividades realizadas en el proyecto.

2.2. Instrumento

Para evaluar la experiencia educativa desarrollada en el proyecto se ha utilizado fundamentalmente cuestionarios realizados en Moodle (evaluación cuantitativa) y encuestas diseñadas para recopilar las opiniones de los estudiantes. También se han recopilado datos a partir de las tareas de laboratorio entregadas de forma online por los estudiantes. Al final de todo el proceso se realiza un examen (cuestionario) en Moodle con el objetivo de comparar los resultados y poder comprobar las posibles mejoras que se producen con la nueva metodología. Por último, como parte de los resultados del proyecto, se realiza una encuesta a los alumnos sobre la nueva metodología en la que podremos observar cuál es su opinión sobre la metodología propuesta.

2.3. Descripción de la experiencia / Procedimiento

Como se ha mencionado previamente, la metodología docente empleada en el proyecto ha utilizado, entre otras, técnicas de gamificación para la docencia de programación de ordenadores (Chen, 2019). Básicamente, los estudiantes se dividen en pequeños grupos y se les da una lista de asignaciones (tareas de programación) en un nivel de dificultad creciente. Las soluciones se envían una por una, y se prueban y califican. Las tareas están inspiradas en problemas reales de ingeniería y los alumnos que obtienen una solución más eficiente y eficaz son los que obtienen mejores calificaciones. La eficiencia de las soluciones obtenidas por los alumnos se mide mediante métricas definidas antes del inicio de la resolución de tareas. Hemos descubierto que esta estructura gamificada ofrece la oportunidad de aumentar el nivel de compromiso y motivación personal de los estudiantes (Cárdenas-Cobo, 2020; Fliz, 2014).

La recopilación de datos durante las experiencias se ha realizado mediante cuestionarios y exámenes de Moodle, tareas de laboratorio y encuestas. Los alumnos involucrados en la experiencia son de las asignaturas “Automatización Industrial” en el Grado de Ingeniería Eléctrica y “Teoría de Sistemas” en el Grado de Ingeniería Mecánica (Universidad Miguel Hernández, España). El número aproximado de alumnos que han participado en las experiencias ha sido de unos 122 (102 de Grado en Ingeniería Mecánica y 20 de Grado en Ingeniería Eléctrica).

En las siguientes subsecciones describimos la metodología basada en algunas de las tareas propuestas a los estudiantes. La metodología utiliza como base la idea de gamificación de tareas prácticas para mejorar la motivación e implicación de los estudiantes (Medeiros, 2019). Como se verá más adelante, es posible mejorar los resultados académicos y reducir la tasa de abandono en cada una de las asignaturas objeto del proyecto. Por razones de espacio, en el presente informe solamente se describe la experiencia correspondiente a la asignatura “Automatización Industrial”. La experiencia de la asignatura “Teoría de Sistemas” es similar, y puede consultarse en las publicaciones que se han derivado del proyecto referidas en el apartado 3.1.

Asignatura “Automatización Industrial”

En la asignatura “Automatización Industrial” se ha aplicado la metodología a varias prácticas de laboratorio. En esta sección se describe la metodología basada en uno de estos trabajos de laboratorio, que consiste en la automatización de maquetas a escala de un proceso industrial real (Fig. 1). Para llevar a cabo este trabajo, los alumnos se dividen en grupos o equipos de 4 personas y cada uno de estos grupos realizará las siguientes fases en el laboratorio:

- **Fase 1.** Familiarización con el equipo (maquetas) y con la tarea a realizar (maniobra de automatización).
- **Fase 2.** Diseño del automatismo utilizando las técnicas de diseño explicadas en las sesiones teóricas (diseño estructurado usando GRAFCET y GEMMA). Este automatismo consiste básicamente en una maniobra de agarre de piezas de diferentes materiales y colores, y su clasificación automática y almacenamiento en un punto determinado según la clasificación.
- **Fase 3.** Programación del diseño realizado en la fase anterior utilizando el lenguaje de los autómatas (en nuestro caso autómatas Siemens S7-1200), en concreto el lenguaje Ladder y el Lenguaje Estructurado (SL).
- **Fase 4.** Prueba de la ejecución de la maniobra y evaluación de la funcionalidad y eficiencia de la solución de cada equipo del grupo.

En el presente informe nos centraremos en las fases 3 y 4 (fases de programación y prueba/evaluación), ya que las dos fases anteriores tratan

temas de teoría y diseño, no relacionados directamente con la programación (objeto del proyecto). Dado que los recursos en el laboratorio son limitados (solo se dispone de una maqueta de cada tipo), se establecen turnos rotativos de 20 minutos para que cada grupo de estudiantes pueda probar su solución en cada maqueta. Previamente, los estudiantes han podido comprobar el funcionamiento de su programa o bloque de programación en un autómata que no está conectado a la maqueta.

De esta forma, los grupos de alumnos se turnan rotando por las cuatro maquetas en las que se divide la experiencia. Dado que el tiempo de cada turno es limitado, una parte de la evaluación/competencia es la rapidez y eficacia en su estrategia de programación del módulo correspondiente, y este tiempo se ha tenido en cuenta en las métricas de evaluación.

Las sesiones de laboratorio en las que se realiza esta tarea tienen una duración total de 6 horas (3 sesiones de 2 horas cada una). En una sesión adicional de 2 horas, se prueba y evalúa la solución de cada grupo.

El criterio para evaluar cada solución es previamente conocido por los alumnos y consta básicamente de los siguientes elementos:

1. **Funcionalidad.** El programa diseñado debe ejecutar la tarea propuesta inicialmente (recoger la pieza, corregir la clasificación por material y colocarla en la ubicación correspondiente según la clasificación realizada por los sensores).
2. **Eficiencia.** Utilizando métricas de software previamente determinadas, se obtiene un índice de eficiencia del programa diseñado. Durante las clases de teoría y las sesiones de laboratorio, los estudiantes aprenden algunas técnicas simples de programación eficiente, tales como: (1) Minimización del uso de variables auxiliares y marcas de bits de los autómatas. (2) Reutilización de bucles y estructuras ya diseñadas en módulos anteriores del programa. (3) Reutilización y uso de variables compartidas utilizadas en diferentes funciones, si es posible. (4) Disminución del tamaño del programa mediante la eliminación de segmentos redundantes. (5) Disminución del tamaño de los segmentos del programa. (6) Uso eficiente de las funciones (utilización de las mismas solamente cuando sea estrictamente necesario en el módulo específico). (7) Optimización del número de temporizadores y contadores a utilizar en el programa mediante un análisis detallado de los tiempos y conteos necesarios.

3. **Uso de la memoria.** Utilizando métricas de software previamente determinadas, se obtienen índices de eficiencia en el uso de la memoria. Algunos de estos índices son generados automáticamente por la herramienta de programación utilizada y están relacionados con los índices de eficiencia del programa.
4. **Evaluación de casos especiales.** En la programación de proyectos de automatización industrial es crucial el análisis de los casos especiales o situaciones excepcionales que se puedan presentar en la operación en tiempo real. Los estudiantes deben diseñar sistemáticamente un punto de referencia de prueba para tener en cuenta todas estas situaciones especiales. Para el diseño de estos ensayos se utiliza el método de la guía GEMMA, combinado con la programación estructurada de sistemas de control.

En la cuarta sesión de laboratorio, cada grupo es evaluado con estas métricas por los demás miembros del grupo y, al final, por el profesor. Combinando estas dos valoraciones, el grupo obtiene un número entre 0 y 100 que se corresponde con la nota global del proyecto. Esta evaluación se repite para cada grupo, estableciéndose al final de todas las sesiones una clasificación ordenada por puntuaciones.

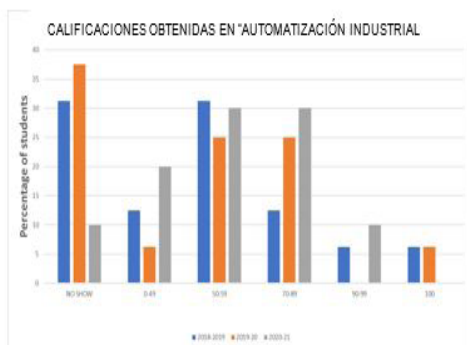


Figura 1. Calificaciones obtenidas en “Automatización Industrial”

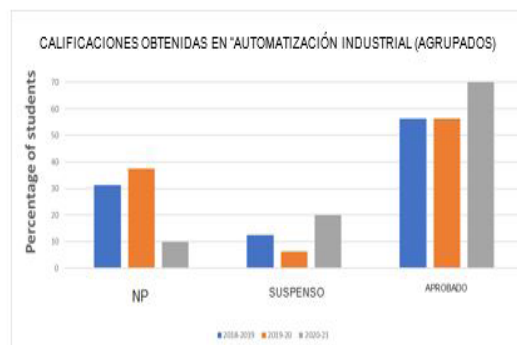


Figura 2. Calificaciones agregadas obtenidas en “Automatización Industrial”.

Adaptación a la Enseñanza Secundaria

En el caso de Enseñanza Secundaria se ha planteado una adaptación de la metodología propuesta basada en la combinación del uso de Programación Orientada a Objetos y Programación Estructurada (POO+PE), que se basa

en la explicación de la programación estructurada (PE) (Abelson, 1996) pero apoyándose en la simplicidad de algunas herramientas visuales de programación orientada a objetos (POO), como Scratch. Es habitual que, al iniciar el proceso de aprendizaje de la programación estructurada, los alumnos encuentren serias dificultades a la hora de comprender el significado y los efectos reales de las funciones que tienen que utilizar para que el programa haga lo que ellos quieren (Haduong, 2019; Medeiros, 2019). En muchos casos se limitan a copiar las instrucciones sin tener claro lo que realmente están haciendo. Otro problema al que se enfrentan es la cantidad de mensajes de error que devuelve el compilador, en la mayoría de los casos por problemas de sintaxis, todos ellos evitables al utilizar la herramienta Scratch donde no se requiere control de sintaxis ya que la mayoría de las operaciones son bloques de instrucciones predefinidas.

La aplicación de la metodología comienza explicando los fundamentos de la programación estructurada (PE) de forma convencional y al final se miden los logros con la metodología convencional. Posteriormente se aplica la nueva metodología POO+PE en las explicaciones de las distintas herramientas de programación, volviendo a medir los resultados. Para finalizar, se realiza una encuesta sobre la opinión que tienen los alumnos sobre la metodología, que nos dará información muy valiosa sobre la utilidad de esta nueva metodología. En la explicación de programación estructurada (PE) se imparten clases explicando los conceptos y herramientas de programación necesarios y realizando ejercicios donde se aplican las diferentes herramientas adquiridas. Los diferentes ejercicios se compilan y los resultados se muestran en la consola de la computadora. Posteriormente, se realiza la evaluación de conocimientos por medio de un examen (cuestionario) en Moodle.

Una vez completada la explicación básica de la PE, se comienza con la aplicación de la nueva metodología POO+PE. Se explican los conceptos de la programación estructurada pero conectando en paralelo los conceptos tanto en POO (programación visual orientada a objetos con Scratch), como en PE (Lenguaje C a través de CODE::BLOCKS). Es decir, cuando se explica un concepto o herramienta de programación se explican también las similitudes entre POO y PE. Posteriormente, se resuelven los mismos ejercicios utilizando POO y PE. De esta forma, el alumno es capaz de asimilar mejor la abstracción del conocimiento y les resulta más útil aprender y retener las instrucciones de PE que se utilizan, alcanzando una comprensión y abstracción más profunda sobre la utilidad de las funciones e instrucciones de la programación estructurada. Esto se consigue fundamentalmente al estudiar su equivalencia

con un lenguaje de POO más “amigable”, en el que la sintaxis no determina si el programa funciona o no y por lo tanto su lógica permanece intacta en los razonamientos (Cárdenas-Cobo, 2020; Filiz, 2014).

Al final de todo este proceso se realiza un examen en Moodle para poder comparar los resultados y poder ver las posibles mejoras que se producen con la nueva metodología. Por último, como parte de los resultados del proyecto, se realiza una encuesta a los alumnos sobre la nueva metodología en la que podremos observar cuál es su opinión sobre la metodología POO+PE.

Los exámenes realizados constan de 17 preguntas. En la pregunta 1, los estudiantes marcan el curso al que pertenecen. En la pregunta 2, los estudiantes deben desarrollar la solución en lenguaje C a un problema que se les presenta de acuerdo a su nivel educativo. Las preguntas 3 a 17 se agruparán por herramientas de programación de la siguiente manera: preguntas sobre la estructura general de un programa (3 a 5), preguntas sobre las funciones de entrada y salida de un programa (6 a 9), preguntas sobre la declaración e inicialización de variables (10 a 12), sobre creación y uso de condiciones (13 a 16), y una pregunta sobre estructura de decisión (17).

3. Resultados

Como se ha comentado previamente, la recopilación de datos se ha realizado mediante cuestionarios y exámenes de Moodle, tareas de laboratorio y encuestas. Los estudiantes involucrados en la experiencia son de las carreras “Automatización Industrial” (AI) del Grado en Ingeniería Eléctrica y “Teoría de Sistemas” (TS) del Grado en Ingeniería Mecánica (Universidad Miguel Hernández de Elche). El número aproximado de alumnos que han participado en las experiencias ha sido de 122 (102 de la asignatura “Automatización Industrial” y 20 de la asignatura “Teoría de Sistemas”). En esta sección resumimos los principales resultados obtenidos en las experiencias. junto con la comparación con los resultados de cursos anteriores y se incluyen los de la asignatura “Automatización Industrial”. Los resultados correspondientes a la asignatura “Teoría de Sistemas” son similares, y pueden consultarse en las publicaciones que se han derivado del proyecto referidas en el apartado 3.1.

Asignatura “Automatización Industrial”

En la figura 1 se muestran las calificaciones obtenidas por los alumnos de la asignatura “Automatización Industrial” en los cursos académicos 2018-19,

2020-21 y 2021-22. Las calificaciones se dividen en 6 intervalos, de acuerdo a la escala de calificación de nuestra universidad, considerados de 0 a 100 (0-49 es reprobado, 50-59 es aprobado, 70-89 es bueno, 90-99 es excelente y 100 es excelente con mención especial). La figura 2 muestra los mismos resultados, pero agregados en las categorías “NO PRESENTADO”, “FALLO” y “PASO”.

De estas cifras podemos ver que, con el uso de la nueva metodología, los resultados de la evaluación han mejorado considerablemente. Así, en los cursos 2018-19 y 2020-21, los porcentajes de “no presentados” fueron del 31,25% y 37,5%, respectivamente, mientras que en el curso 2021-22 (en el que se aplicó la metodología) fue de solo 10 %, Hemos observado una disminución muy pronunciada en el porcentaje de alumnos que no se presentan a los exámenes y pruebas de evaluación de la asignatura utilizando la metodología propuesta.

En la Fig. 5 podemos ver que el porcentaje de alumnos que aprobaron la asignatura en los cursos anteriores 2018-19 y 2020-21 fue del 56,25% en ambos cursos, mientras que en el curso 2021-22 este porcentaje ascendió al 70% (un aumento de aproximadamente 14%). Podemos concluir que, desde el punto de vista de los resultados académicos, con el uso de la metodología propuesta, se ha logrado reducir la tasa de abandono de los alumnos, así como el número de alumnos que superan con éxito la asignatura.

Adaptación a Enseñanza Secundaria

Tras el análisis de los exámenes y cuestionarios realizados antes y después de la aplicación de la metodología, presentamos los resultados y conclusiones extraídas de los mismos en el caso de su adaptación y aplicación a cursos de Enseñanza Secundaria.

Como se puede observar en todos los resultados presentados, la aplicación de la metodología combinada (POO+PE) mejora, en general, los resultados obtenidos de la metodología convencional (PE) en el aprendizaje de la programación estructurada y razonamiento abstracto. El promedio de puntuación en las respuestas para todas las preguntas siempre es mayor con la metodología combinada.

Resultados para todas las categorías

La figura 3 muestra los resultados generales para el tercer, cuarto y sexto año de educación secundaria (14-15-16 y 17-18 años). En esta figura puede observarse que el uso de la metodología POO-PE mejora los resultados

generales en todas las preguntas.

El número promedio de puntuaciones en las respuestas para todas las preguntas siempre es mayor con el método combinado. Cabe señalar que en el tercer gráfico (edad 17) hay varias preguntas donde no hay mejora ya que los estudiantes han respondido el 100% de las preguntas utilizando ambas metodologías. Este caso solo se da con más frecuencia en esta asignatura, posiblemente porque los alumnos de mayor edad tienen mayor capacidad de abstracción y estudio y, por tanto, son capaces de asimilar más conceptos de programación con metodología tradicional.



Figura 3. Resultados generales en Enseñanza Secundaria.

Encuestas de opinión a los estudiantes

Como se ha comentado previamente, basándonos en la herramienta de cuestionarios de Moodle, hemos realizado una encuesta a cada nivel y grupo para conocer la opinión de los alumnos sobre el aprendizaje de la programación y su utilidad, las metodologías utilizadas para su enseñanza, en especial el uso de la nueva programación orientada a objetos, la metodología POO+PE, y algunas preguntas genéricas. Esta información se combinará con la información objetiva obtenida de las diferentes pruebas y herramientas que se utilizaron para verificar el nivel de aprendizaje con la metodología (PE) respecto a la metodología combinada (POO+PE) para el aprendizaje de programación estructurada.

El cuestionario consta de 25 preguntas divididas en tres apartados de interés para el estudio, que son los siguientes: aprendizaje del alumno (13), motivación del alumno (9), cuestiones generales y aplicación de la programación en la sociedad (3). A continuación describimos los principales resultados del cuestionario.

Respecto a las preguntas dirigidas a conocer cómo han funcionado las diferentes metodologías en el aprendizaje de los alumnos, podemos obtener las siguientes conclusiones: El 93% de los estudiantes considera que la programación para la vida real es “muy útil” o “útil”. Además, el 95% considera que la programación es muy útil o útil para aprender y comprender las Ciencias de la Computación. En cuanto a la preferencia de un lenguaje de programación específico, el 74% eligió el lenguaje C sobre POO (Scratch) que es elegido solo por el 19%. Las preferencias a la hora de elegir POO (Scratch) sobre el lenguaje C son: 77% por la facilidad de uso, 5% por la potencia de la herramienta para resolver problemas y 18% por ser divertido. Sin embargo, al observar las preferencias para elegir el lenguaje C sobre POO (Scratch), el 92% eligió el poder de esta herramienta para resolver problemas, el 7% por la facilidad de uso y el 0% por diversión. Cuando se les pregunta sobre qué herramientas elegirían para programar, el 69% responde que Lenguaje C, el 18% OOP (Scratch) y el 14% otros. En la pregunta sobre si el uso de la combinación POO+PE les ha ayudado a comprender mejor la programación estructurada, el 76% respondió que les ha sido muy útil o útil frente al 7% que respondió que no fue muy útil o nada útil. Además, el 55% respondió que la nueva metodología les ha ayudado a obtener una mejor calificación en el examen (los estudiantes no saben la calificación cuando realizan la encuesta, ni en el primer examen ni en el final) frente al 21% que no y 24% que no sabe.

De las respuestas a las preguntas dirigidas a conocer la motivación de los estudiantes a la hora de aprender a programar, obtuvimos los siguientes resultados: La combinación de las herramientas POO+PE para aprender programación estructurada hace que la tarea de programar sea mucho o mucho más interesante para un 41% frente a un 21% que piensa poco o nada. Los estudiantes consideran el 74% frente al 8% que la programación es muy útil o útil para resolver tareas de la vida real. Además, el 54%, frente al 20%, considera que son más originales y creativos después de haber aprendido a programar. Asimismo, el 59% de los estudiantes, frente al 15%, piensa que el uso de la programación les ha hecho pensar y reflexionar sobre cuestiones no planteadas anteriormente. El 51% de los estudiantes cree que el profesor debería utilizar la metodología combinada POO+PE en la docencia de programación, mucho o bastante. Solo el 21% piensa que la metodología combinada no se debe utilizar, poco o nada.

3.1 Publicaciones realizadas en el marco del proyecto.

Durante el año 2022, el trabajo realizado en el proyecto ha dado lugar a las publicaciones siguientes:

- Pérez, J.R., Ñeco, R.P., García, N.M., & Catalán, J.M. (2022, July 4-6). *A Proposed Approach to Teaching Introductory Programming. Experiences for non-specialists in Computer Science*. International Conference on Education and New Learning Technologies, Palma de Mallorca, Spain.
- Pérez, J.R., & Ñeco, R.P. (2022, July 4-6). *Educational Activities Proposal to Motivate Students to Learn the Basics of Computer Programming in Secondary Education*. International Conference on Education and New Learning Technologies, Palma de Mallorca, Spain.

También se ha realizado una publicación (en forma de póster y como resumen en las actas) en las XX Jornadas de Redes de Investigación en Docencia Universitaria (REDES-INNOVAESTIC 2022) de la Universidad de Alicante (presentación el 14 de julio).

En la fecha de la redacción del presente informe se han enviado para su revisión 1 artículo adicional que está pendiente de aceptación, en el que se recopilan los últimos datos. Con la recopilación de los últimos resultados y conclusiones del proyecto se pretende redactar un nuevo artículo para enviarlo para su revisión y posible publicación en una revista.

4. Conclusiones

Como resultado principal de nuestras investigaciones y experiencias, encontramos que hemos obtenido una mejora en la enseñanza de la programación de computadores para cursos y grados en los que puede ser difícil motivar a los estudiantes en esta materia. Los efectos de este impacto han sido que los estudiantes pueden asimilar de una manera más efectiva los conocimientos de estas materias, y hemos encontrado que puede ayudar a mejorar la percepción negativa de que la programación informática es difícil de aprender. Esto se ha traducido en una mejora en las calificaciones de los módulos de las asignaturas en las que se utilizan conceptos de programación, y una disminución en el número de alumnos que abandonan el curso. Además, estas metodologías se pueden adaptar fácilmente a diferentes paradigmas de enseñanza y cursos. Como trabajo futuro, pretendemos extender la aplicación

de la metodología a nuevas sesiones prácticas en las materias mencionadas en este trabajo, así como a materias correspondientes a cursos más avanzados.

En el trabajo de adaptación de la metodología a niveles educativos no universitarios, la principal conclusión es que con la aplicación de la metodología propuesta se ha obtenido una mejora cuantificable en la enseñanza de la programación en los niveles educativos mencionados, observándose un impacto positivo en el proceso de enseñanza-aprendizaje. Además, hemos observado que la mayoría de los alumnos implicados en las experiencias han podido interiorizar y asimilar de forma más eficaz los conocimientos de la materia. Todo lo anterior viene avalado por las conclusiones que hemos extraído de las encuestas realizadas a los alumnos sobre la metodología propuesta y las pruebas objetivas realizadas. Podemos concluir diciendo que la metodología POO+PE se puede aplicar en diferentes niveles educativos de forma sencilla, mejorando el proceso de enseñanza-aprendizaje de la programación estructurada.

Para los próximos cursos académicos estamos explorando la posibilidad de aplicar parte de la metodología (POO+PE) en la asignatura “Fundamentos de Programación”, en el Grado de Ingeniería de Imagen y Sonido en Telecomunicación de la Universidad de Alicante, y la asignatura “Teoría de Sistemas” en el Grado en Ingeniería Mecánica de la Universidad Miguel Hernández de Elche. La comprensión, en etapas iniciales, de conceptos básicos sobre programación estructurada mejoraría claramente si aplicáramos esta metodología, adaptándola al entorno universitario. El uso de herramientas visuales de programación orientada a objetos sería muy prometedor para comprender los conocimientos básicos y las herramientas de la programación estructurada.

5. Tareas desarrolladas en la red

Se enumera a continuación cada uno de los componentes y se detallan las tareas que ha desarrollado en la red.

Tareas:

[1] Diseño de la estrategia de enseñanza para las asignaturas “Taller de videojuegos I”, “Informática”, “Taller de videojuegos II”, “Tecnologías de la

Información y la Comunicación, TIC” y “Tecnologías de la Información y la Comunicación II, TIC II”.

[2] Aplicación estrategia de enseñanza para las asignaturas “Taller de videojuegos I”, “Informática”, “Taller de videojuegos II”, “Tecnologías de la Información y la Comunicación, TIC” y “Tecnologías de la Información y la Comunicación II, TIC II” (diciembre 2021-abril 2022).

[3] Diseño de la estrategia de enseñanza para la asignatura “Teoría de Sistemas”

[4] Diseño de la estrategia de enseñanza para la asignatura “Automatización Industrial”

[5] Diseño de la estrategia de enseñanza para la asignatura “Control Avanzado”.

[6] Aplicación estrategia de enseñanza para las asignaturas “Automatización Industrial” y “Control Avanzado”

[7] Aplicación estrategia de enseñanza para las asignaturas “Teoría de Sistemas”

[8] Recogida y formateo de datos

[9] Análisis de datos y obtención de conclusiones

Participante de la red	Tareas que desarrolla
Jorge Ramón Pérez Beltrán	[1], [2], [3], [4], [6], [7], [8], [9]
Ramón P. Ñeco García	[1], [2], [3], [5], [6], [7], [8], [9]
Nicolás M. García Aracil	[3], [4], [5], [6], [8], [9]
José M. Catalán Orts	[3], [4], [5], [6], [8], [9]
Iván Fabra Ramón	[8], [9]

6. Referencias bibliográficas

Abelson, H., Sussman, G.J. (1996). *Structure and Interpretation of Computer Programs*. Cambridge, Massachusetts: MIT Press.

Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M., & Palincsar, A.

- (1991). Motivating project-based learning: sustaining the doing, supporting the learning. *Educ. Psychol.* 26(3–4), 369–398.
- Bolton, W. (2015). *Programmable Logic Controllers*. Elsevier.
- Bonwell, C.C., Eison, J.A. (1991). *Active learning: creating excitement in the classroom*. ERIC Clearinghouse on Higher Education.
- Cárdenas-Cobo, J., Puris, A., Novoa-Hernández, P., Galindo, J., & Benavides, D. (2020). Recommender Systems and Scratch: An Integrated Approach for Enhancing Computer Programming Learning. *IEEE Trans. on Learn. Tech.*, 13, 387–403.
- Díaz-Ramírez, J. (2020). Gamification in engineering education – An empirical assessment on learning and game performance. *Heliyon*, 6, 49-72.
- Dichev, C., & Dicheva, D. (2017). Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International journal of educational technology in higher education*, 14(1), 1-36.
- Dori, Y.J., & Belcher, J.W.. (2005). How does technology-enabled active learning affect undergraduate students' understanding of electromagnetism concepts? *J. Learn. Sci.* 14(2), 243–279.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: a systematic review. *Inf. Softw. Technol.* 50(9–10), 833–859.
- Chen, C., Haduong, P., Brennan, K., Sonnert, G., & Sadler, P. (2019). The effects of first programming language on college students' computing attitude and achievement: a comparison of graphical and textual languages. *Computer Science Education*, 29(1), 23-48.
- Felix, C., & Sobral, S.R. (2020, April 27-30). *Predicting students' performance using survey data* [Paper presentation]. In: EDUCON2020 – IEEE Global Engineering Education Conference, Porto, Portugal.
- Filiz, K., & Yasemin, G. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Hundhausen, C., & Brown, J. (2007). What you see is what you code: a “live” algorithm development and visualization environment for novice learners. *J. Vis. Lang. Comput.* 18(1), 22–47.

- Kao, G.-M., Lin, S., Sun, C.-T. (2008). Beyond sharing: engaging students in cooperative and competitive active learning. *Educ. Technol. Soc.* 11(3), 82–96.
- Kasahara, R. (2019). Applying gamification to motivate students to write high-quality code in programming assignments. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*.
- McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Commun. ACM*, 49(8), 90–95.
- Medeiros, R.P., Ramalho, G.L., & Falcão, T.P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Trans. Educ.*, vol. 62, 77–90.
- Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall.
- Paravizo, E., Chaim, O.C., Braatz, D., Muschard, B., & Rozenfeld, H. (2018). Exploring gamification to support manufacturing education on industry 4.0 as an enabler for innovation and sustainability. *Procedia Manufacturing*, 21, 438-445.
- Park, J. (2019). Learning to be better at the game: Performance vs. completion contingent reward for game-based learning. *Computers & Education*, 139, 1-15.
- Pedersen, S., & Liu, M. (2003). Teachers' beliefs about issues in the implementation of a student centered learning environment. *Educ. Tech. Res. Dev.* 51(1), 57–76.
- Sánchez, D.R., Langer, M., & Kaur, R. (2020). Gamification in the classroom: Examining the impact of gamified quizzes on student learning. *Computers & Education*, 144, 10-36.
- Schach, r. (2010). *Object-Oriented and Classical Software Engineering*. McGraw Hill.
- Sobral, S.R. (2020). CS1 & CS2 curriculum recommendations: learning from the past to try not to rediscover the wheel again. *Advances in Intelligent Systems and Computing*, 3, 182-191.
- Sobral, S.R. (2020b) The first programming language and freshman year in

computer science: characterization and tips for better decision making.
Advances in Intelligent Systems and Computing, 3, 150-156.