

Jornadas de Automática

Aprendizaje Basado en Problemas para enseñanza de sistemas de tiempo real con Ada

Cañadas-Aránega, Fernando^{a,*}, Barrientos, Antonio^b, Berenguel, Manuel^a, Moreno, José C.^a, Rodríguez, Francisco^a

^aUniversidad de Almería, Dpto. de Informática, CIESOL, ceia3, Ctra. Sacramento s/n, 04120, Almería, España.

^bUniversidad Politécnica de Madrid, Dpto. de Automática, C/ José Gutiérrez Abascal, n° 2, 28006, Madrid, España.

To cite this article: Cañadas-Aránega, Fernando, Barrientos, Antonio, Berenguel, Manuel, Moreno, José Carlos, Rodríguez, Francisco. 2025. Problem-Based Learning for teaching real-time systems with Ada. *Jornadas de Automática*, 46. <https://doi.org/10.17979/ja-cea.2025.46.12236>

Resumen

El Aprendizaje Basado en Problemas (ABP) es una metodología eficaz para fomentar competencias entre distintos ámbitos. Su aplicación promueve la participación activa del alumnado y potencia su motivación, siendo útil a la hora de abordar contenidos complejos. El aprendizaje de los conceptos fundamentales de los Sistemas de Tiempo Real (STR), muy demandados por su aplicación en sistemas críticos, es un claro ejemplo de este tipo de contenido. Para abordar este reto, se plantea un proyecto centrado en Ada, con el objetivo de despertar el interés del alumnado de la asignatura Control por Computador del Grado en Ingeniería Electrónica Industrial y Automática de la Universidad de Almería. El proyecto implementa el control persecución pura de un robot Ackermann, permitiendo resolver un problema clásico de control reforzando los conceptos de STR. En la experiencia participaron 25 estudiantes durante cinco sesiones prácticas. La evaluación mediante cuestionarios y su comparación *ex ante* y *ex post* mostró mejoras significativas en la comprensión de los STR.

Palabras clave: ABP, Lenguaje de programación ADA, Robótica en educación, Control por computador, Programación concurrente, Redes de Petri, Robot móvil

Problem-Based Learning for teaching real-time systems with Ada

Abstract

Problem-Based Learning (PBL) is an effective methodology for fostering transferable skills across various fields. Its implementation encourages active student participation and enhances motivation, making it particularly valuable when addressing complex topics. The learning of basic concepts of Real-Time Systems (RTS), which are highly demanded due to their application in critical systems, is a clear example of this kind of topics. To address this challenge, a project was designed around an Ackermann mobile robot programmed in Ada, aiming to stimulate student interest in the Computer Control course of the Bachelor's Degree in Industrial Electronics and Automation Engineering at the University of Almería. The project implements the pure pursuit control algorithm, allowing students to solve a classic control problem while reinforcing RTS concepts. A total of 25 students participated in the experience over five practical sessions. Evaluation through questionnaires and *ex ante/ex post* comparison showed significant improvements in the students' understanding of RTS.

Keywords: PBL, ADA programming language, Robotics in education, Computer Control, Concurrent Programming, Petri Nets, Mobile Robot

1. Introducción

En la enseñanza universitaria, especialmente en disciplinas como la Ingeniería, es habitual que los cursos presenten

una fuerte carga teórica. Suele impartirse mediante clases magistrales, en las que el profesor asume un rol principal en la transmisión del conocimiento hacia el alumnado. No obstante, este enfoque puede fomentar una actitud pasiva y receptiva por

*Autor para correspondencia: fernando.ca@ual.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

parte del estudiantado, lo que, en ciertos casos, conlleva una disminución del interés y la motivación (González-Hernández et al., 2024). Para contrarrestar esta situación, resulta fundamental explorar nuevas estrategias que permitan presentar los contenidos teóricos de manera más atractiva y estimulante, promoviendo así una participación activa del alumnado en su propio proceso de aprendizaje. Entre las distintas alternativas metodológicas, el Aprendizaje Basado en Problemas (ABP) ha despertado un notable interés en la última década (Gil et al., 2024; González-Hernández et al., 2025).

Esta metodología ha sido ampliamente adoptada en la educación en ingeniería desde la segunda mitad del siglo-XX (Wood, 2003). Este tipo de metodología de enseñanza-aprendizaje se desarrolló por primera vez en la educación médica, como respuesta a los problemas que la simple memorización producía en los estudiantes de ciencias de la salud. Como resultado de estas experiencias piloto, se concluyó que el ABP no solo ayudaba a mejorar el rendimiento académico, y a mantener la atención y el compromiso del alumnado, sino que también permitía desarrollar habilidades técnicas claves para convertir al alumnado en profesionales exitosos (Lavado-Anguera et al., 2024). Para el estudiantado de ingeniería esta capacidad de resolución de problemas es, quizás, una de las habilidades más importantes para su futuro desarrollo profesional. Es por eso por lo que, la implementación de estas metodologías en la enseñanza de conceptos complejos constituye una estrategia didáctica clave para optimizar el aprendizaje en contextos educativos.

En este contexto, los STR constituyen una rama del conocimiento de gran importancia para los futuros ingenieros en cuya formación se incluyen contenidos relacionados con el control automático, debido a su aplicabilidad en entornos críticos que requieren de garantías de seguridad en la ejecución de la implementación de los controladores. A menudo, estos conceptos suelen producir rechazo a los alumnos debido a su complejidad (Rodríguez et al., 2016a). Por este motivo, el alumnado recibe formación específica sobre programación concurrente y el control de STR, para alcanzar objetivos que no son viables si se usan técnicas de programación que no tratan de forma explícita las características propias de estos sistemas (Greer and Heaney, 2004). No obstante, la programación de STR abarca numerosos conceptos abstractos que pueden resultar complejos de asimilar para los estudiantes. Habitualmente, se emplea un lenguaje de programación específico como método para facilitar el aprendizaje, lo que en algunos casos puede llevar al estudiante a percibir la asignatura como una mera continuación de los cursos introductorios de programación, dificultando así asimilar el enfoque sistémico y aplicado que esta disciplina requiere (Rodríguez et al., 2016a,b).

Por tanto, en este trabajo se plantea que el estudiantado se enfrente a un aprendizaje basado en un problema de control de persecución pura (*pure pursuit*) (Craig Coulter, 1992) aplicado a un robot móvil tipo Ackermann. Parten de una plantilla con una simulación cíclica y, a lo largo de cinco sesiones prácticas, desarrollan una planificación de tareas concurrentes.

El código implementado permite la interacción con las entradas y salidas del robot como recursos compartidos, gestionados desde un bucle de control principal. Esta estructura facilita el diseño de tareas concurrentes de forma intuitiva y motivadora para el alumnado. Para abordar este problema, se ha seleccionado el lenguaje de programación Ada, dada su orientación específica hacia el desarrollo de sistemas concurrentes y de tiempo real (Booch et al., 1994; Burns and Wellings, 2009). El compilador utilizado es GNAT, mantenido actualmente por AdaCore¹. Como entorno de desarrollo se emplea Visual Studio Code², una herramienta gratuita, ampliamente adoptada a nivel internacional, y especialmente atractiva por su versatilidad y facilidad de uso en entornos de ingeniería.

El estudio se ha llevado a cabo durante el segundo cuatrimestre del curso académico 2024/2025, en el marco de la asignatura Control por Computador, correspondiente al tercer curso del Grado en Ingeniería Electrónica Industrial y Automática en la Universidad de Almería (UAL). La experiencia se ha desarrollado con un total de 25 estudiantes, quienes recibieron previamente la formación teórica necesaria para adquirir los conocimientos fundamentales y afrontar con garantías la metodología ABP. Con el objetivo de evaluar la efectividad de la propuesta, se suministraron dos cuestionarios: uno al inicio de la experiencia y otro al finalizarla. Ambos instrumentos estuvieron orientados a analizar aspectos clave como la motivación del alumnado, el grado de aprendizaje alcanzado y el desarrollo de habilidades transversales, en particular de resolución de problemas y de trabajo en equipo. Los resultados obtenidos reflejan una valoración muy positiva respecto a los ejercicios planteados y la metodología empleada.

El resto del trabajo está organizado como se indica a continuación. La sección 2 presenta una descripción detallada del marco docente de aplicación, descripción y planteamiento de las herramientas en Ada con los conceptos de STR. La descripción y desarrollo del método de evaluación del ABP se describe en la sección 3. Por último, en la sección 4 se exponen las principales conclusiones del trabajo.

2. Diseño de la simulación

En esta sección se detalla el diseño y desarrollo de la metodología basada en ABP.

2.1. Marco de enseñanza

Control por computador es una asignatura de 6 créditos ECTS. Se divide en dos módulos: control de sistemas en tiempo discreto y control de STR. En el primero, el alumnado afronta contenidos relacionados con el análisis de sistemas y diseño de controladores en tiempo discreto. Los conceptos analizados en el primer módulo son implementados y adaptados a STR en el módulo dos. Este grado permite al alumnado desarrollar habilidades para trabajar en un contexto multidisciplinar, en el que se aborden problemas relacionados con los sistemas de control desde distintos ámbitos de conocimiento. El objetivo final se centra en la aplicación de conceptos de STR en el ámbito industrial.

¹<https://www.adacore.com/>

²<https://code.visualstudio.com/>

Con el fin de alcanzar los objetivos planteados, se diseñó y aplicó la siguiente metodología docente: Una vez finalizado el módulo uno, durante el mes siguiente se impartieron sesiones teóricas centradas en los fundamentos de los STR. Estas sesiones incluyeron, además, la resolución de ejercicios individuales orientados a afianzar los conceptos presentados. En los meses posteriores, se adoptó una metodología combinada, alternando sesiones teóricas con clases prácticas de programación, permitiendo así una asimilación progresiva del conocimiento. Finalmente, durante el último mes del curso, se conformaron equipos de trabajo de dos estudiantes que resolvían un caso de estudio simulado, diseñado para evaluar tanto las competencias específicas como transversales del estudiantado. Estas actividades representan el 50 % de la calificación final de la asignatura, lo que pone de manifiesto su relevancia formativa. Su desarrollo, que requiere una dedicación estimada de 25 horas por parte del alumnado, se integra como un componente esencial en el proceso de adquisición de competencias.

2.2. Descripción del problema

Se ha diseñado un problema de programación en Ada que el alumnado debe modificar y analizar mediante un guión de prácticas. Este ejercicio se basa en la implementación de un control de persecución pura, lo que implica una lectura de sensores (codificadores) que proporciona los datos necesarios para calcular la señal de control en términos de velocidad y dirección, cuyos valores se transmiten a los actuadores (motores eléctricos de corriente continua (CC) de dirección y tracción). Dado el carácter dinámico y concurrente del problema, el código desarrollado sigue el paradigma de programación en tiempo real y se estructura en cuatro tareas concurrentes, cada una dedicada a una actividad fundamental dentro del sistema. Este enfoque permite una gestión eficiente de la información sensorial y la respuesta de los actuadores, asegurando un comportamiento determinista en el proceso de control.

- La *tarea de control de tracción* se encarga de calcular la velocidad (v) requerida para que el robot se desplace hacia el nuevo punto destino. Esta tarea incluye la lectura de datos del codificador ($E1$), el cálculo de la variable de control (v) y la escritura del valor correspondiente en el actuador.
- La *tarea de control de dirección* es responsable del cálculo del ángulo de orientación del robot (ψ). Al igual que la tarea anterior, realiza una lectura del codificador ($E2$), calcula la señal de control (ψ) y transmite el resultado al actuador para su ejecución.
- La *tarea de almacenamiento de datos* está enfocada en la gestión de la información obtenida durante la ejecución del sistema. El alumnado registra los datos de interés en un archivo de texto (.txt), lo que permite su posterior análisis y representación gráfica.
- La *tarea de supervisión de alarmas* tiene como objetivo la detección y notificación de errores en la dirección del robot. Utilizando la pantalla como recurso compartido, genera una alerta cuando la desviación en la dirección supera 1° de manera consecutiva durante cinco periodos de muestreo.

Para emular el comportamiento de un sistema embebido real, se implementa una simulación de una tarjeta de adquisición analógica/digital encargada de gestionar las operaciones de entrada/salida del sistema, como la lectura de sensores y el envío de señales a los actuadores. Debido a la naturaleza de este tipo de dispositivos, su acceso debe restringirse a una única tarea a la vez, ya sea en modo de lectura o escritura, lo que requiere modelar la tarjeta como un recurso compartido y sincronizado dentro del sistema. En la Figura 1 se muestra la Red de Petri Temporal (RdPT) utilizada para la planificación de tareas en la fase de diseño del código. Se pueden apreciar las cuatro tareas concurrentes previamente descritas y la interacción entre las mismas mediante los distintos recursos compartidos.

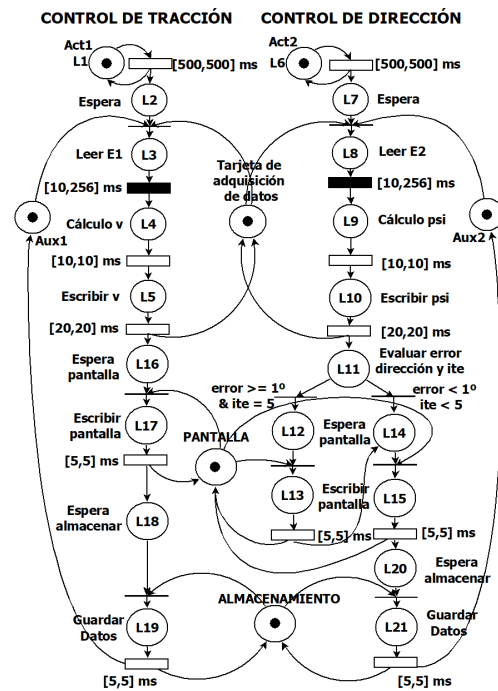


Figura 1: RdPT del proceso descrito

Esta descripción es la base con la que el alumnado inter actuará para resolver una serie de ejercicios (ver sección 2.4). Para estas tareas, el lazo de control se diseñó teniendo como referencia el esquema presentado en Cañadas-Aránega et al. (2024) para el esquema de persecución pura, pero eliminando los cálculos de técnicas de control (PID con *antiwindup* o control *feedforward*) presentadas en el artículo, con el objetivo de que los estudiantes se centren en los aspectos de STR.

En el diseño del problema de control planteado a cada equipo, la longitud del robot se establece como un parámetro variable, dependiendo del número del grupo asignado. Este enfoque permite generar distintos escenarios de análisis, garantizando que cada equipo trabaje con un modelo personalizado. El valor de la longitud (L) varía en un rango comprendido entre 0.4 m y 0.8 m, proporcionando una diferenciación clara entre los problemas de control propuestos.

Es relevante destacar que, aunque la longitud del vehículo puede variar entre grupos de trabajo, el valor empleado en los controladores se mantiene constante. Esta decisión no compromete el resultado, ya que estos valores han sido calculados

con el valor más desfavorable al que los alumnos pueden enfrentarse (0.8 m). Entonces, para longitudes menores, la única diferencia radica en que el sistema alcanza la referencia en menos tiempo debido a un menor radio de giro, logrando igualmente cumplir el objetivo con éxito. Esta es la idea en la que se basa este problema, ya que introduce variabilidad en cada caso de estudio sin comprometer el resultado, reforzando así la naturaleza diferenciadora de cada problema planteado.

2.3. Diseño de la simulación en Ada

Para poder guiar al alumnado de la forma más interactiva posible, buscando mejorar su motivación y aprendizaje, el código de simulación cuenta con estos tres archivos principales: `Robot_CpC2025.adb` (nucleo principal del sistema donde los alumnos interactúan), `Robot_Ackermann.ads` (Interfaz pública del robot Ackermann) y `Robot_Ackermann.adb` (cuerpo del robot Ackermann).

Los archivos proporcionados tienen como objetivo facilitar al alumnado el diseño de sistemas concurrentes de manera rápida y eficiente, permitiendo el desacoplamiento de las cuatro tareas descritas en la sección anterior. Este enfoque favorece la comprensión y aplicación de los principios de concurrencia en un entorno práctico. Para guiar a los estudiantes en la implementación del sistema, se ha puesto a disposición un proyecto completo acompañado de una detallada guía de instalación. Toda la documentación y el código fuente pueden consultarse en *GitHub*. Adicionalmente, se ha proporcionado una plantilla en *Python* para la representación de datos y la generación de gráficas. Esta herramienta permite a los estudiantes interactuar con este lenguaje de programación y mejorar su capacidad de análisis, enriqueciendo así su experiencia en el desarrollo del sistema.

2.4. Actividades planteadas

Para garantizar que los estudiantes y las estudiantes sigan una planificación estructurada dentro del enfoque ABP, se han diseñado una serie de actividades progresivas. Estas permiten que el alumnado avance desde conceptos fundamentales hasta aspectos más complejos de manera gradual y guiada. La resolución de los ejercicios se lleva a cabo en cinco sesiones prácticas de dos horas cada una, en las que los grupos cuentan con el asesoramiento continuo del profesor. En la fase inicial, cada ejercicio incluye una descripción detallada del proceso, facilitando la comprensión y el desarrollo autónomo de los problemas planteados, tal y como se describe a continuación:

El control multivariable de un robot móvil tipo Ackermann, junto con el registro de datos y la realización de labores de supervisión, requiere de la implementación de los siguientes procesos o actividades.

- *Control de la tracción (M1). Se realizará con un período de muestreo de 500 ms. Al ser una tarea de control, requiere realizar tres actividades; lectura del codificador E1 (tiempo de cómputo entre 10 ms y 256 ms); cálculo de la acción de control en el ordenador (tiempo de cómputo de 10 ms); envío de la señal calculada al motor CC (tiempo de cómputo de 20 ms).*
 - *Control de la dirección (M2). Se realizará con un período de muestreo de 500 ms. Al ser una tarea de control, requiere de tres actividades: la lectura del codificador E2; cálculo de la acción de control en el ordenador; envío de la señal calculada al motor DC. Considere las mismas especificaciones temporales que en la tarea anterior.*
 - *Almacenamiento de datos (A1). Se encarga de registrar en un archivo las variables de interés y que está en permanente comunicación con las dos actividades de control previas. El archivo de registro constituye un recurso compartido, dado que se ha de evitar la escritura simultánea desde diferentes tareas.*
 - *Supervisión y gestión de alarmas (S1). Sirve de enlace entre el operador humano y las actividades de control. En particular, deberá avisar por pantalla (recurso compartido) cuando el error de la dirección sea mayor de 1° durante más de 5 periodos de muestreo.*
- Con esta descripción, los equipos deben ser capaces de resolver los siguientes ejercicios:
- **Ejercicio 1:** Desarrollo de la representación de la RdPT, según se muestra en la Figura 1.
 - **Ejercicio 2:** Adaptación de la plantilla de código suministrada para representar de forma cíclica el modelo previamente desarrollado en la RdPT.
 - **Ejercicio 3:** Implementación de una programación completamente concurrente de las tareas, permitiendo una ejecución paralela y eficiente del sistema modelado.
 - **Ejercicio 4:** Los alumnos deben realizar el análisis de la planificación temporal para verificar el cumplimiento de los plazos de respuesta. Además, deben graficar los resultados y analizar las principales diferencias entre los ejercicios 2 y 3.
- La evaluación de las prácticas se centra principalmente en la correcta implementación de los conceptos fundamentales de programación en tiempo real, más allá del funcionamiento completamente operativo del código. El objetivo es que el alumnado demuestre la comprensión y aplicación de elementos clave como la concurrencia, la temporización y la gestión de recursos, incluso si el resultado final no alcanza una ejecución plenamente funcional.

3. Resultados

Con el fin de evaluar el impacto de esta metodología, se administró un cuestionario a los estudiantes y las estudiantes antes de iniciar los ejercicios prácticos y otro al finalizar los mismos. En general, se ha realizado un cuestionario a un grupo de 25 estudiantes (81.5 % hombres y 18.5 % mujeres) del grado en Ingeniería Electrónica Industrial y Automática, donde el 81.5 % es la primera vez que cursa la asignatura. La comparación de las respuestas obtenidas en ambos casos permite valorar la efectividad del enfoque propuesto, así como determinar si el problema planteado ha contribuido a incrementar la motivación del alumnado para la adquisición de los conceptos relacionados con STR.

3.1. Cuestionario 1

En el cuestionario previo, se realizan un total de 21 preguntas divididas en: A. Cuestiones generales, B. Comprensión teórica, C. Motivación en el ABP y D. Opinión abierta. En la Tabla 1 se muestra un resumen de las preguntas cuyos resultados arrojan la información más importante (el resto de preguntas se encuentra disponible en línea en la carpeta de cuestionarios en *GitHub*).

- **Preguntas generales:** En cuanto a las preguntas más generales, aunque el 96.3 % del estudiantado está interesado en la ingeniería de sistemas y automática, solo el 48.1 % considera que puede aprobar la asignatura.
- **Comprensión teórica:** En general, el alumnado muestra una preocupación con la programación de sistemas concurrentes en Ada. El 51.9 % considera tener un nivel muy bajo de programación y casi un 60 % se siente inseguro para detectar tareas concurrentes en un sistema aleatorio. Aunque las expectativas del alumnado son bajas, el 55.6 % considera que el concepto con el que se encuentra más seguro se centra en las RdPT.
- **Motivación en el ABP:** Del total del estudiantado, el 59.2 % se encuentra motivado para la realización de la práctica (del cual, el 14.8 % muy motivado), seguido de un 25.9 % que la calificó como "media". Aunque este resultado es esperanzador, el 25.9 % se encuentra con un nivel bajo de confianza y un 18.5 % con un nivel muy bajo.
- **Opinión abierta:** En cuanto a las dudas de carácter abierto, el 88.4 % presenta preocupación por la programación en Ada.

Estos resultados indican al profesorado los aspectos en los que debe de hacer especial hincapié en clase y donde se encuentran los principales problemas de los conceptos de STR.

3.2. Cuestionario 2

Tras la finalización de las prácticas, los alumnos completan un segundo cuestionario compuesto por 32 preguntas divididas en: A. Cuestiones generales, B. Satisfacción del ABP, C. Valoración de habilidades y D. Opinión abierta. Estas cuestiones se realizan a la misma muestra de alumnos, conservando las respuestas de las preguntas generales. En la Tabla 2, se presenta un análisis de las cuestiones que arrojan información más importante al profesorado (este cuestionario se encuentra disponible en *GitHub*). A continuación, se describen los resultados:

- **Preguntas generales:** Se planteó la pregunta "Considero que podré aprobar la asignatura", pasando de un 48.1 % en el cuestionario previo a un 95.7 % en el cuestionario de evaluación.
- **Satisfacción con el ABP:** Se presentan las cuestiones que están enfrentadas con los conceptos de concurrencia, que son aquellos que presentaban mayor problema

en el cuestionario previo. En este caso, un 70 % considera que este ABP le ha servido para mejorar considerablemente los conceptos relacionados con concurrencia. Además, la pregunta B.15 presenta un ejemplo corto donde deben seleccionar una sola solución correcta, habiendo acertado el 100 % del alumnado. Estos resultados, junto al resto de repuestas, han llevado al 91.7 % del alumnado a considerar que este ABP les ha servido para aprobar la asignatura, donde al 87 % les gustaría que se implementara en otras asignaturas.

- **Valoración de habilidades:** En general, tal y como se muestra en la pregunta C.24 de la Tabla 2, el 86.9 % considera que ha mejorado su capacidad de análisis y desarrollo de problemas técnicos.
- **Opinión abierta:** Como ya se vio en el cuestionario 1, los alumnos tenían una gran preocupación en la programación en Ada. Como resultado de este ABP, el 89 % han comentado que su habilidad para realizar la programación concurrente ha mejorado considerablemente, destacando que un 82.6 % está completamente satisfecho con su resultado final.

4. Conclusiones

Este trabajo presenta el diseño y aplicación de una estrategia didáctica basada en Aprendizaje Basado en Proyectos (ABP), dirigida a 25 estudiantes del Grado en Ingeniería Electrónica Industrial y Automática de la Universidad de Almería, durante cinco sesiones prácticas de la asignatura Control por Computador en el curso 2024/2025. La propuesta se centró en trabajar los conceptos clave de los Sistemas de Tiempo Real (STR), considerados especialmente complejos por el alumnado, utilizando como eje metodológico la simulación de un robot Ackermann programado en el lenguaje Ada.

Tras la exposición teórica inicial de los STR, se aplicó un cuestionario *ex ante* que determina el nivel de partida. A continuación, el alumnado desarrolló el proyecto práctico con acompañamiento docente. Al finalizar, se administró un segundo cuestionario *ex post* para evaluar el impacto de la intervención. Los resultados iniciales mostraban un bajo nivel de confianza, con un 60 % manifestando inseguridad en programación concurrente y solo un 48,1 % creyendo poder aprobar la asignatura.

Tras el ABP, se observó una mejora notable: el 95,7 % consideró que superaría la asignatura y el 70 % afirmó haber comprendido mejor los conceptos de concurrencia. El 100 % resolvió correctamente un ejercicio aplicado a un sistema real, el 86,9 % mejoró sus habilidades de análisis y resolución de problemas técnicos, y el 89 % percibió avances significativos en programación concurrente en Ada.

Esta metodología supone también una transformación en el rol docente, que pasa de la transmisión de contenidos a la fácil interacción activa del aprendizaje. El profesorado debe diseñar cuidadosamente los proyectos, anticipar dificultades técnicas y guiar de forma personalizada, lo que implica una mayor dedicación respecto a métodos tradicionales.

No obstante, la implantación del ABP presenta limitaciones según el contexto. En grupos numerosos (por ejemplo, 75

| Preguntas | | Respuestas | | | | |
|-----------|---|--|-----------------------------|------------------------------------|--------------------------------|--------------------------|
| A.9 | Considero que podré aprobar la asignatura | Sí (48.1 %) | | | No (51.9 %) | |
| B.10 | ¿Cómo valoras tu nivel sobre el lenguaje de programación ADA? | Muy bajo 51.9 % | Bajo 40.7 % | Medio 7.4 % | Alto 0 % | Muy alto 0 % |
| B.12 | Sobre un sistema real aleatorio, ¿Serías capaz de detectar las tareas que deben ejecutarse de manera concurrente? | No 18.5 % | Podría plantearlo 44.4 % | Sí, pero no estoy seguro 29.6 % | Con seguridad 7.4 % | |
| B.14 | ¿Cuál de los siguientes conceptos dominas mejor? | RdPT 55.6 % | Prog Cíclica 3.7 % | Prog Concurrente 0 % | Planificación de tareas 0 % | Ninguno 33.3 % |
| C.17 | ¿Qué grado de motivación consideras que tienes para afrontar esta práctica? | Muy bajo 3.7 % | Bajo 11.1 % | Medio 25.9 % | Alto 44.4 % | Muy alto 14.8 % |
| C.19 | ¿Con qué nivel de confianza crees que podrás aplicar los conceptos teóricos en el problema práctico? | Nada 18.5 % | Poco 25.9 % | Algo 40.7 % | Mucho 11.1 % | Totalmente 3.7 % |
| D.21 | ¿Qué aspectos te generan más dudas o inseguridad antes de comenzar este trabajo? | Respuesta abierta relacionada con la programación en Ada 88.4 % | | | | Otros aspectos 11.6 % |

Tabla 1: Encuesta con estudiantes del cuestionario 1

| Preguntas | | Respuestas | | | | |
|-----------|--|---|---------------|-----------------|---|---|
| A.9 | Considero que podré aprobar la asignatura | Sí (95.7 %) | | | No (17.4 %) | |
| B.13 | ¿El uso de esta simulación te ayudó a entender los conceptos relacionados con la comunicación y sincronización de tareas concurrentes? | Nada 0 % | Poco 8.7 % | Algo 17.4 % | Bastante 52.2 % | Mucho 17.3 % |
| B.15 | Imagina que estás analizando el funcionamiento interno de un sistema de control de tráfico. ¿Podrías identificar qué partes del sistema podrían beneficiarse si se hace uso de programación concurrente? | Ejecución de una tarea para análisis de datos 0 % | | | Ejecución de varias tareas en paralelo 100 % | Ejecución de una tarea para mostrar por pantalla 0 % |
| B.17 | ¿Consideras que esta práctica te ha servido para entender mejor el papel que juegan los conceptos de STR en la implementación de los sistemas de control por computador? | Sí 95.7 % | | | No 4.3 % | |
| B.18 | ¿Cómo valorarías tu nivel de motivación durante el desarrollo de la práctica? | Muy bajo 8.7 % | Bajo 13 % | Medio 47.8 % | Alto 21.7 % | Muy alto 8.7 % |
| B.19 | ¿Consideras que el enfoque basado en un problema, frente a clases más tradicionales, ha favorecido tu aprendizaje? | Nada 0 % | Poco 8.7 % | Algo 13 % | Bastante 56.5 % | Mucho 21.7 % |
| B.21 | ¿Te gustaría volver a usar este tipo de Aprendizaje Basado en Problemas (ABP) en otras asignaturas? | Sí 87 % | | | No 13 % | |
| B.22 | Considero que los conceptos aprendidos en esta práctica de control por computador me serán útiles para aprobar la asignatura. | Sí 95.7 % | | | No 4.3 % | |
| C.24 | ¿Cómo valorarías tu desarrollo en habilidades de análisis y resolución de problemas técnicos? | Muy bajo 4.3 % | Bajo 8.7 % | Medio 56.5 % | Alto 26.1 % | Muy alto 4.3 % |
| D.28 | ¿Qué aspectos de la práctica planteada destacarías como más positivos? | Mejora en la programación con Ada 89 % | | | Otros aspectos 11 % | |
| D.30 | ¿Te gustaría que se plantearan más problemas ABP como este en otras asignaturas? ¿Por qué? | Trabajar con un problema práctico ayuda a entender mejor los conceptos - 98 % | | | Otros aspectos 2 % | |
| D.32 | ¿Te sientes satisfecho/a con el resultado final del trabajo? | Sí 82.6 % | | | No 17.4 % | |

Tabla 2: Encuesta con estudiantes del cuestionario 2

estudiantes), sería difícil mantener el seguimiento individualizado sin recursos adicionales. Además, su efectividad depende del nivel de madurez del alumnado, pudiendo ser menos adecuada en cursos iniciales con estudiantes aún en desarrollo de competencias de autoaprendizaje.

En conclusión, los resultados reflejan que el ABP favorece no solo la adquisición de competencias técnicas complejas, sino también la motivación, autoconfianza y satisfacción del alumnado. Se presenta, por tanto, como una estrategia pedagógica eficaz en asignaturas de alto contenido técnico, siempre que se adapte a las condiciones del entorno educativo.

Agradecimientos

El primer autor, Fernando Cañadas-Aránega, cuenta con una beca FPI (PRE2022-102415) del Ministerio de Ciencia, Innovación y Universidades de España.

Referencias

- Booch, G., Bryan, D. L., Petersen, C. G., 1994. Software engineering with Ada. Vol. 30608. Addison-Wesley Professional.
- Burns, A., Wellings, A., 2009. Real-time systems and programming languages. Addison-Wesley Professional.
- Cañadas-Aránega, F., Moreno, J. C., Blanco-Claraco, J. L., 2024. A PID-based control architecture for mobile robot path planning in greenhouses. IFAC-PapersOnLine 58 (7), 503–508.
- Craig Coulter, R., 1992. Implementation of the pure pursuit path tracking algorithm. In: Technical Report CMU-RI-TR-92-01. The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
- Gil, J. D., González-Hernández, J., Cañadas-Aránega, F., Pataro, I. M. L., Hoyo, Á., Otálora, P., Rodríguez, F., Guzmán, J., Berenguel, M., 2024. Gamification in control engineering: An ongoing initiative at the University of Almería. IFAC-PapersOnLine 58 (26), 82–87.
- González-Hernández, J., Cañadas-Aránega, F., Hoyo, Á., Otálora, P., Pataro, I. M. L., Gil, J. D., 2024. Escape room in engineering of control: Una aventura gamificada desarrollada en la Universidad de Almería. Jornadas de Automática (45).
- González-Hernández, J., Cañadas-Aránega, F., Hoyo, Á., Otálora, P., Pataro, I. M. L., Vergel, J. D. G., 2025. Estimulando el aprendizaje a través del juego: Una experiencia gamificada en ingeniería de control en la universidad de almería. Revista Iberoamericana de Automática e Informática Industrial. Aceptado para publicación.
- Greer, L., Heaney, P. J., 2004. Real-time analysis of student comprehension: An assessment of electronic student response technology in an introductory Earth Science course. Journal of Geoscience Education 52 (4), 345–351.
- Lavado-Anguera, S., Velasco-Quintana, P.-J., Terrón-López, M.-J., 2024. Project-based learning (PBL) as an experiential pedagogical methodology in engineering education: a review of the literature. Education Sciences 14 (6), 617.
- Rodríguez, C., Guzmán, J. L., Berenguel, M., Dormido, S., 2016a. Teaching real-time programming using mobile robots. IFAC-PapersOnLine 49 (6), 10–15.
- Rodríguez, C., Guzmán, J. L., Berenguel, M., Moreno, J. C., Rodríguez, F., Dormido, S., 2016b. Robótica móvil para el aprendizaje de conceptos de programación en tiempo real. In: XXXVII Jornadas de Automática. Comité Español de Automática, pp. 821–826.
- Wood, D. F., 2003. Problem based learning. Bmj 326 (7384), 328–330.