

Aprendizaje activo de Bases de Datos orientado a competencias en Ingeniería Informática

Miguel A. Martínez-Prieto*, Paula Mielgo†, Aníbal Bregón‡, Jorge Silvestre§

Departamento de Informática, Universidad de Valladolid

*miguelamp@uva.es, †paula.mielgo@uva.es, ‡anibal.bregon@uva.es §jorge.silvestre@uva.es

Resumen—Las asignaturas obligatorias de bases de datos se imparten en los primeros cursos del Grado en Ingeniería Informática, lo que dificulta la integración de sus competencias con las propias del desarrollo de software. Para abordar esta situación, presentamos una propuesta metodológica que combina aprendizaje basado en proyectos y prácticas ágiles en un contexto de aprendizaje activo, donde los estudiantes diseñan, crean, cargan y manipulan una base de datos realista. Su aplicación en una asignatura de 2º curso evidencia una mejora progresiva en la adquisición de competencias, así como en la madurez y cohesión de los equipos de estudiantes, favoreciendo un aprendizaje más profundo y transferible al ámbito profesional.

Abstract—Mandatory database courses are scheduled in the early years of Computer Engineering Bachelor's degrees, making it difficult for students to integrate database with software development skills. This paper presents a methodology to address this issue. It integrates project-based learning and agile practices in an active learning context where students learn to design, build, load and manipulate a near-real-world database. The reported results of a 2nd year database subject have shown a progressive improvement in the acquisition of competences, as well as in the maturity and cohesion of the student teams, enabling a deeper and more transferable learning to the professional field.

Palabras clave—Bases de datos, aprendizaje activo, aprendizaje basado en proyectos, Scrum, competencias

I. INTRODUCCIÓN

Las bases de datos son un pilar fundamental de los sistemas informáticos, por lo que su ciclo de vida debe integrarse completamente en el proceso de desarrollo del software, garantizando que, desde su diseño inicial hasta su implantación y evolución, respondan a las necesidades del sistema en su conjunto. Asimismo, los desarrolladores deben poseer las competencias necesarias para abordar cualquier etapa de este ciclo de vida, además de ser capaces de comprender cómo los resultados producidos en ellas se integran en el proceso de desarrollo del software y de desenvolverse satisfactoriamente en él, colaborando en el equipo de trabajo que lo lleva a cabo. El desarrollo de estas competencias depende, en gran medida, de la formación que reciben los estudiantes en las titulaciones de Ingeniería Informática.

Las asignaturas obligatorias de bases de datos en los Grados en Ingeniería Informática en España tienden a concentrarse en 2º curso, dejando en general los cursos superiores para asignaturas optativas. Aproximadamente la mitad de las titulaciones incluyen solo una asignatura obligatoria sobre la materia, por lo que muchos estudiantes tienen una única exposición al aprendizaje de bases de datos en una fase temprana de su formación, cuando aún no han consolidado competencias en desarrollo de software ni cursado asignaturas de Ingeniería

del Software. Nuestra asignatura *Sistemas de Bases de Datos* (SBD) es un ejemplo de esta situación. SBD tiene una carga de trabajo de 6 ECTS, se imparte en el primer cuatrimestre de 2º curso del Grado en Ingeniería Informática de Servicios y Aplicaciones de la Universidad de Valladolid y los estudiantes llegan a ella habiendo cursado apenas dos asignaturas de programación y una de arquitectura de computadores (además de las asignaturas de matemáticas, física y organización de empresas que se imparten en 1º). Así, las competencias con las que los estudiantes cursan la asignatura son limitadas en comparación con las que dispondrán cuando apliquen su aprendizaje en el ámbito profesional. Esta brecha evidencia la necesidad de replantear la enseñanza de bases de datos, proporcionando a los estudiantes un contexto que les permita desarrollar las competencias propias de la materia y lograr un aprendizaje profundo y transferible al entorno profesional [1].

Este artículo presenta una propuesta de *aprendizaje basado en proyectos* (ABP) fundamentada en la naturaleza iterativa del diseño de bases de datos, que es la piedra angular de nuestro proyecto educativo. Connolly y Begg [2] describen el diseño de datos como un *proceso de aprendizaje*, compuesto por refinamientos sucesivos que permiten desarrollar una comprensión profunda y estructurada del dominio de una organización, mediante el modelado de sus datos y los usos previstos para ellos. Esta misma dinámica subyace en Scrum [3], que es uno de los marcos de trabajo más utilizados en la gestión de proyectos y cuyas prácticas han sido adaptadas con éxito al ámbito educativo [4]. A partir de estas experiencias, nuestra propuesta incluye dos contribuciones principales:

- La definición de un *proyecto* que estructura las competencias relacionadas con el ciclo de vida de una base de datos que deben adquirir los estudiantes de Ingeniería Informática [5] y propone instrumentos específicos para su evaluación formativa y sumativa.
- Una *metodología* que organiza el proyecto como una actividad de ABP en equipo y desarrolla una dinámica de trabajo ágil, en la que se adaptan los roles, eventos y artefactos de Scrum para abordar las necesidades propias de un entorno de aprendizaje.

El artículo también presenta los resultados de nuestra propuesta a través de PLAYROCK, un proyecto centrado en el desarrollo de la base de datos de una red social para fans del rock. Los estudiantes de SBD valoran positivamente la similitud de PLAYROCK con un proyecto real, así como la forma en la que acompaña y evalúa el aprendizaje. Los

resultados académicos muestran una evolución positiva en todas las tareas del ciclo de vida, reflejando tanto el desarrollo de habilidades técnicas como el fortalecimiento de la cohesión y madurez de los equipos, factores clave en este rendimiento.

El resto del artículo establece los antecedentes de nuestro trabajo en la sección II. Las secciones III y IV presentan de forma detallada el proyecto y la metodología de trabajo, antes de evaluar su impacto en la sección V. Finalmente, la sección VI presenta las conclusiones y las líneas de trabajo futuro.

II. ANTECEDENTES

La enseñanza de bases de datos ha evolucionado desde un enfoque más teórico, centrado en el modelo relacional, hasta una visión más amplia, que incluye la gestión del ciclo de vida y la administración de la base de datos, así como otras tecnologías más recientes. De acuerdo con el *Computer Science Curricula 2023* (CS2023) [5], la formación universitaria en bases de datos debe abordar competencias de modelado de datos y construcción de consultas, además de proporcionar la capacidad de organizar cualquier colección de datos, acceder a ella y actualizarla utilizando los recursos proporcionados por un sistema gestor de bases de datos. En definitiva, establece un equilibrio entre fundamentos teóricos y prácticos, en el marco de las primeras etapas del ciclo de vida de una base de datos. Además, CS2023 reconoce la creciente relevancia de aspectos legales, tecnológicos y profesionales vinculados a la gestión y explotación de datos. La forma en la que se concretan todas estas directrices varía dependiendo de la orientación de los planes de estudios y de la importancia relativa de las bases de datos dentro de ellos.

A continuación, el apartado II-A analiza brevemente la situación de la enseñanza de bases de datos en la Universidad española y el apartado II-B completa esta sección con una breve revisión del trabajo relacionado con nuestra propuesta.

II-A. Enseñanza universitaria de bases de datos

La información analizada en este apartado se ha obtenido del Registro de Universidades, Centros y Títulos (RUCT), mediante una búsqueda de las titulaciones de *Grado* que contienen *Ingeniería Informática* en su denominación. Posteriormente, hemos revisado la última publicación en el BOE y/o las páginas web institucionales de cada una de ellas para recopilar las asignaturas que incluyen *Bases de Datos* en su título. En total, se han identificado 165 asignaturas en 80 titulaciones impartidas en 70 universidades.

Nuestro análisis revela que todas las titulaciones incluyen al menos una asignatura obligatoria de bases de datos, siendo la única sobre la materia en 17 de ellas, y que casi la mitad (37) ofertan dos o más asignaturas obligatorias. En total, se identifican 124 asignaturas obligatorias, dos tercios de las cuales se concentran en los dos primeros cursos. Además, solo 7 titulaciones retrasan su primera asignatura obligatoria hasta el tercer curso, mientras que las 30 que incluyen asignaturas optativas las ubican en tercer y cuarto curso.

Estos datos muestran que la enseñanza de bases de datos suele comenzar en una etapa temprana, lo que puede provocar que los estudiantes las perciban como conceptos aislados, en

lugar de como piezas clave en la construcción de sistemas software, lo que limita la profundidad y aplicabilidad de su aprendizaje. Por ello, la enseñanza de bases de datos debe apoyarse en metodologías que reflejen la naturaleza integrada del diseño y gestión de bases de datos en entornos reales [6].

II-B. Trabajo relacionado

Ishaq y otros [7] señalan que la enseñanza de bases de datos ha ido incorporando metodologías de aprendizaje más activas y experienciales, capaces de mejorar el rendimiento académico de los estudiantes, aunque a costa de una mayor planificación y demanda de recursos para el profesorado. En particular, destacan el aprendizaje basado en proyectos (ABP) [8] por su capacidad para conectar el diseño de bases de datos con la resolución de problemas reales, creando entornos de aprendizaje más alineados con la realidad profesional. La naturaleza constructivista del ABP fomenta el aprendizaje activo, ya que los estudiantes deben desarrollar un proyecto aplicando los conocimientos y habilidades técnicas adquiridos en la asignatura, además de fortalecer habilidades transversales como el trabajo en equipo, la toma de decisiones y la resolución de problemas [9].

Las primeras experiencias de ABP en bases de datos se remontan a tres décadas atrás [10]. En ellas se destaca la capacidad del ABP para mejorar la calidad del aprendizaje y desarrollar competencias como la autonomía y el pensamiento crítico, aunque también se indica que su éxito depende de que los estudiantes cuenten con un marco de referencia que guíe su trabajo en el proyecto [10]. Siguiendo esta línea, Domínguez y Jaime [11] integran un proceso de *scaffolding* [12] en una metodología que combina ABP y técnicas de gestión de proyectos para la enseñanza del diseño de bases de datos. Su propuesta incluye un esquema estructurado con las fases y tareas que los equipos de estudiantes deben completar para diseñar y construir la base de datos de un sistema software de su elección. Los resultados de cinco cursos muestran mejoras en el rendimiento académico y la motivación estudiantil, aunque con un aumento en la carga de trabajo para estudiantes y docentes en comparación con los métodos “tradicionales”.

Esta necesidad de estructura puede abordarse usando marcos de trabajo ágiles, diseñados para la gestión de proyectos mediante dinámicas iterativas y adaptativas. Su aplicación en entornos educativos ofrece una guía clara, sin renunciar a la autonomía del estudiante, lo que ha impulsado el interés por la educación ágil [13]. Bwye y otros [14] analizan los desafíos de incorporar Scrum [3] en actividades de aprendizaje en equipo, señalando que los estudiantes suelen carecer de la formación necesaria para aplicarlo. Esto, junto con su falta de experiencia en planificación y trabajo en equipo, puede dificultar el desarrollo del proyecto y afectar al aprendizaje.

Esta revisión refuerza las fortalezas del ABP en la adquisición de conocimientos y habilidades sobre bases de datos, así como la oportunidad de integrar prácticas ágiles para mitigar la incertidumbre que enfrentan los estudiantes durante su ciclo de vida. Asimismo, evidencia la necesidad de equilibrar la autonomía en el desarrollo del proyecto con una supervisión estructurada que guíe su aprendizaje sin limitarlo.

III. PROYECTO

El objetivo principal del proyecto es proporcionar a los estudiantes un entorno estructurado de aprendizaje donde aplicar las competencias de la materia en un contexto similar al profesional. Para ello, contempla una serie de tareas que, en su conjunto, materializan el ciclo de vida de una base de datos conforme a los objetivos de aprendizaje de la asignatura. La realización de estas tareas requiere de un conjunto de competencias, entendidas como la integración de los conocimientos, las habilidades y las disposiciones profesionales [5] que los estudiantes deben adquirir durante la asignatura.

Además de los conocimientos y habilidades técnicas, completar estas tareas exige desarrollar habilidades transversales y disposiciones clave para el futuro profesional. La orientación del proyecto hacia el trabajo en equipo demanda que cada estudiante sea capaz de comunicarse y coordinarse de forma efectiva con el resto de compañeros. A este respecto, una buena gestión del tiempo es fundamental para cumplir con la planificación del proyecto y mantener un ritmo de trabajo constante que evite los “atracones finales”, que suelen darse en las actividades de aprendizaje. El desarrollo del proyecto también exige capacidad para resolver los problemas que se manifiestan en un proceso tan complejo como el ciclo de vida de una base de datos y que, en este caso, se acentúan dado que quienes los enfrentan están en una etapa formativa. Por lo tanto, la ocurrencia de fallos es esperable, lo que demanda que los estudiantes sean capaces de asumirlos y abordarlos como una oportunidad para seguir mejorando su aprendizaje, siempre con atención a la excelencia de los resultados producidos en cada tarea del proyecto. La constancia en el trabajo y la atención al detalle completan las disposiciones necesarias para mantener la coherencia de los resultados obtenidos en el proyecto y asegurar un aprendizaje de calidad y duradero.

A continuación, el apartado III-A presenta las tareas que definen el alcance del proyecto y el apartado III-B describe los instrumentos propuestos para evaluar el grado de adquisición de las competencias requeridas en cada una de ellas.

III-A. Alcance

El alcance del proyecto abarca cinco tareas, que corresponden a las primeras etapas del ciclo de vida de una base de datos. En ellas se trabajan su diseño y puesta en funcionamiento, sin incluir aspectos de administración u optimización. La Figura 1 ilustra las tareas y sus respectivos resultados, que en su conjunto definen las competencias *core* que debe adquirir un estudiante de Ingeniería Informática [5].

Diseño conceptual: consiste en identificar, organizar y especificar los requisitos de información según su relevancia y significado en el dominio de la base de datos. Estos requisitos se estructuran en el *modelo entidad-relación* y se documentan en el *diccionario de datos*, que son los resultados de esta tarea [2]. Los estudiantes deben comprender los fundamentos de ambas técnicas y saber aplicarlas correctamente para obtener un diseño conceptual sólido. Esta tarea representa uno de los mayores desafíos del proyecto, ya que requiere una comprensión profunda del contexto de negocio y, a partir de ella, que los estudiantes sean capaces de identificar y especificar con

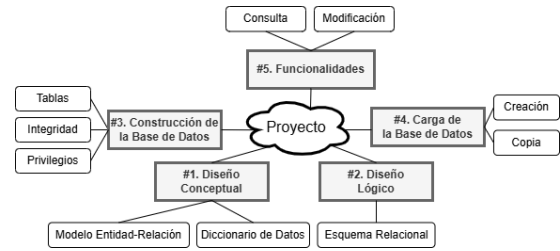


Figura 1. Descripción de las tareas incluidas del proyecto.

precisión las entidades, relaciones y atributos que definen los requisitos de información y garantizan el cumplimiento de las reglas de negocio del proyecto.

Diseño lógico: transforma el diseño conceptual en un modelo lógico de datos que, en este caso, corresponde al modelo relacional [2]. El resultado de esta tarea es el *esquema relacional* de la base de datos, donde se define el esquema de cada tabla, incluyendo sus claves primarias y foráneas. Para completar esta tarea, los estudiantes deben comprender los fundamentos del modelo relacional, el concepto y la implicación del dominio de las columnas, así como la función de claves primarias y foráneas, además de conocer y aplicar correctamente las reglas de transformación del modelo entidad-relación. Aunque esta tarea es más sistemática que la anterior, sigue requiriendo sentido crítico para aplicar las reglas de transformación y garantizar que el esquema resultante refleje fielmente el diseño conceptual, minimizando la redundancia y asegurando la consistencia de la información. Cualquier modificación en el diseño conceptual implicará una actualización del esquema relacional.

Construcción de la base de datos: implementa el diseño lógico en un sistema gestor de bases de datos relacional. Así, el primer resultado es la creación de las *tablas* definidas en el diseño lógico, garantizando que cada columna tenga un dominio compatible con los tipos de datos y restricciones de obligatoriedad, unicidad y demás condiciones establecidas en el diccionario de datos. El segundo resultado consiste en la declaración de las restricciones de *integridad* necesarias para mantener la integridad referencial mediante claves foráneas y para soportar reglas de negocio más avanzadas mediante condiciones de *check* o *triggers*. Además, en esta tarea también se establecen los *privilegios* que determinan qué operaciones pueden realizar los distintos tipos de usuario sobre la base de datos, según lo establecido en el pliego del proyecto. Para completar esta tarea, los estudiantes deben conocer y utilizar el DDL (*Data Definition Language*) de SQL [2], comprender qué es y cómo crear *triggers*, y conocer los mecanismos discrecionales de seguridad en bases de datos. Esta tarea también requiere habilidades de gestión de información, para garantizar que todo el código generado se adhiera a lo especificado en el diccionario de datos.

Carga de datos: juega un papel clave en el proceso de aprendizaje, ya que hasta este punto los estudiantes han diseñado y construido las estructuras necesarias para gestionar los datos, pero a partir de este momento es cuando comienzan a trabajar directamente con ellos. El objetivo de esta

tarea es poblar las tablas mediante procesos de *creación* y *copia* de datos. En el primer caso, los estudiantes registran manualmente información conforme a lo establecido en el pliego del proyecto, y en el segundo copian información desde una base de datos externa, cuyo esquema difiere del de la base de datos construida en el proyecto. En la práctica, esta tarea permite a los estudiantes detectar posibles errores en las etapas anteriores, ya que las dificultades en la carga de datos pueden evidenciar fallos en el diseño o construcción de la base de datos. Para completar esta tarea, los estudiantes deben conocer y utilizar los operadores de modificación de datos disponibles en el DML (*Data Manipulation Language*) de SQL [2], además de comprender el esquema de la base de datos existente y saber cómo consultarla para extraer la información necesaria. Asimismo, deben demostrar buena capacidad crítica para interpretar las incidencias que puedan surgir durante el proceso de carga y resolverlas de acuerdo con las decisiones tomadas en las tareas previas.

Funcionalidades: consiste en la creación de sentencias SQL para la *consulta* y *modificación* de la base de datos, en función de los requisitos de usuario establecidos en el pliego del proyecto. Esta tarea permite a los estudiantes interactuar con la base de datos y demostrar su capacidad para resolver las necesidades de los usuarios dentro del dominio del proyecto. Al igual que la anterior, esta tarea también cumple un propósito de validación, ya que la dificultad para implementar ciertas funcionalidades o la obtención de resultados inesperados pueden evidenciar defectos en las tareas previas. Para su realización, los estudiantes deben conocer y utilizar los operadores de consulta disponibles en el DML (*Data Manipulation Language*) de SQL [2]. Además, esta tarea exige capacidad crítica para analizar y gestionar las incidencias que puedan surgir durante la implementación, ya sea debido a errores en fases previas o a una construcción incorrecta de las sentencias SQL.

III-B. Evaluación

La naturaleza iterativa del ciclo de vida de las bases de datos requiere mecanismos que valoren tanto el resultado final del proyecto como su evolución. Además, el enfoque a la tarea que plantea este proyecto y su desarrollo en un contexto de aprendizaje realista, nos lleva a optar por una evaluación basada en el desempeño (*performance-based assessment*) [15]. Esta estrategia permite valorar el desempeño de los estudiantes en cada tarea y, por tanto, evaluar su capacidad para diseñar bases de datos, construirlas, cargarlas y utilizarlas de acuerdo con las funcionalidades solicitadas.

Nuestro sistema de evaluación valora regularmente la calidad del producto (evaluación sumativa) y, en base a ello, proporciona retroalimentación continua a los estudiantes a lo largo del proyecto (evaluación formativa). Para ello, se contemplan dos entregas parciales y una final, en las que el alcance se incrementa progresivamente con el ciclo de vida de la base de datos. Este aspecto define nuestro sistema de evaluación, que se presenta en la Tabla I.

Como se observa en la última fila, cada entrega tiene un peso mayor que la anterior, de modo que el producto

TABLA I
SISTEMA DE EVALUACIÓN DEL PROYECTO

Tarea	Resultado	Entrega					
		Parcial #1		Parcial #2		Final	
#1	Modelo ER Diccionario	✓	50 %	↑	30 %	=	20 %
#2	Relacional	✓	30 %	↑	10 %	=	5 %
#3	Tablas Integridad Privilegios	✓	20 %	↑ ✓	20 %	= ↑ ✓	15 %
#4	Creación Copia			✓ ✓	20 %	= ↑	20 %
#5	Consulta Modificación			✓ ✓	20 %	↑ ↑	40 %
		0,15		0,35		0,50	

final representa el 50 % de la calificación, mientras que su evolución a lo largo del proyecto constituye la otra mitad. La primera entrega parcial se centra en las fases iniciales, evaluando el diseño conceptual (#1), el diseño lógico (#2) y la creación de tablas dentro de la tarea de construcción de la base de datos (#3), con ponderaciones del 50 %, 30 % y 20 %, respectivamente. Estos resultados se incluyen también en la segunda entrega, permitiendo a los estudiantes incorporar el *feedback* recibido y abordar un incremento de alcance (↑) debido a la introducción de nuevos requisitos en el proyecto. En esta fase, la ponderación relativa de las tareas iniciales disminuye (por ejemplo, el diseño conceptual pasa del 50 % al 30 %) para dar cabida a nuevos resultados: las restricciones de integridad en la construcción de la base de datos, la carga parcial de datos (#4) y un primer conjunto de funcionalidades básicas (#5). Por último, la entrega final integra todos los resultados del proyecto: se incorporan los privilegios de acceso a la base de datos (✓), se amplía el alcance de los resultados de la segunda entrega (↑) para incluir restricciones avanzadas de integridad, la carga completa de datos y funcionalidades más complejas, y se mantienen (=) las tareas incluidas desde el inicio del proyecto y la creación de datos. Además, las ponderaciones se ajustan nuevamente para reflejar la importancia relativa de cada tarea en esta fase, asegurando así una evaluación equilibrada y alineada con la progresión del proyecto.

La evaluación de cada resultado se concreta mediante una rúbrica que incluye criterios de aceptación definidos para medir objetivamente la calidad de producto [16]. La tabla II muestra la rúbrica utilizada para evaluar las restricciones de integridad, que incluye dos criterios de aceptación: en la parte izquierda, entre corchetes, se indica la ponderación de cada criterio en la evaluación sumativa, mientras que los porcentajes incluidos en su descripción establecen el nivel mínimo de calidad necesario para su aceptación. Por ejemplo, si los *triggers* implementados aseguran el 75 % de las comprobaciones esperadas, se considera que la base de datos satisface el criterio. Si, por el contrario, las verificaciones realizadas mediante *checks* solo alcanzan el 60 % de los casos esperados, el criterio no se cumple. Cabe destacar que los porcentajes establecidos en las rúbricas se incrementan progresivamente a lo largo del proyecto, de modo que cada entrega exige un nivel de calidad superior al de la anterior.

TABLA II
RÚBRICA DE EVALUACIÓN DE LAS RESTRICCIONES DE INTEGRIDAD

[25 %] DDL-3.1. La base de datos implementa correctamente, al menos, el 80 % de las comprobaciones necesarias para garantizar la consistencia básica de los datos (<i>checks</i>)
[75 %] DDL-3.2. La base de datos implementa correctamente, al menos, el 70 % de las comprobaciones necesarias para garantizar la consistencia avanzada de los datos (<i>triggers</i>).

IV. METODOLOGÍA

Para establecer entornos de aprendizaje orientados a competencias es esencial alinear constructivamente los resultados de aprendizaje, las actividades diseñadas para alcanzarlos y los mecanismos utilizados para evaluarlos [17]. Por lo tanto, nuestro proyecto requiere una metodología que movilice los conocimientos, las habilidades y las disposiciones requeridas para que los estudiantes alcancen los resultados esperados con un nivel de calidad que satisfaga el sistema de evaluación.

Scrum [3] se presenta como una solución adecuada, ya que proporciona un marco de trabajo estructurado que favorece el desarrollo iterativo e incremental del producto en ciclos cortos (*sprints*). La interacción continua con el cliente permite incorporar su retroalimentación en cada *sprint*, asegurando con ello que el desarrollo del producto se mantenga alineado con sus expectativas y necesidades. Además, Scrum fomenta una cultura de mejora continua, refinando tanto el producto como las dinámicas de trabajo a partir de la experiencia acumulada (*aprendizaje*), e impulsando un trabajo coordinado con responsabilidades definidas en cada fase del proyecto.

Aunque Scrum proporciona una estructura sólida para la gestión de proyectos, no está diseñado específicamente para entornos educativos. Por ello, proponemos una metodología denominada PROBLAM (*Project-Based Learning with Agile Management*), que adapta el equipo, los eventos y los artefactos de Scrum a un entorno de aprendizaje basado en proyectos.

IV-A. Equipo

Un equipo PROBLAM está formado por 4 o 5 estudiantes y el profesor de la asignatura. Son los propios estudiantes los que deciden la composición del equipo, cuya responsabilidad principal es asegurar el aprendizaje efectivo de todos sus integrantes. Por ello, todos los estudiantes participan en todas las tareas planificadas, asegurando el desarrollo de las competencias necesarias para realizarlas. Al igual que en Scrum, el equipo es autorganizado, pero dentro del marco que define la guía de trabajo, establecida por el profesor. La mejora continua es clave para el desarrollo del equipo y de cada estudiante, que al final de cada *sprint* valora su trabajo y el de sus compañeros mediante un informe de valoración. Esta evaluación, orientada a la autorreflexión, busca que tanto el equipo como cada estudiante, a nivel individual, aprendan de la experiencia. El equipo PROBLAM distingue tres roles: estudiante, profesor y capitán, inspirados en los roles de desarrollador, *Product Owner* y *Scrum Master* de Scrum [3].

Estudiante: se encarga de completar los trabajos asignados dentro del equipo, asegurando que sus resultados cumplan con los niveles de calidad establecidos en el sistema de

evaluación. Asimismo, es responsable de completar el informe de valoración y de utilizar sus resultados para mejorar su desempeño como miembro del equipo y, en definitiva, asumir la responsabilidad de su propio aprendizaje.

Profesor: es responsable de maximizar la calidad del aprendizaje del equipo PROBLAM a través de la retroalimentación de cada entrega del producto. Asimismo, establece la guía de trabajo, proporcionando el *scaffolding* [12] necesario para que el equipo planifique cada *sprint*. Finalmente, colabora con el capitán en la resolución de los impedimentos técnicos y organizativos que este no haya podido solucionar y que puedan afectar el progreso del equipo.

Capitán: es un rol rotatorio, asumido por 1 o 2 estudiantes en cada *sprint*. Su responsabilidad es facilitar la organización y el desarrollo de los trabajos dentro del marco establecido por la guía de trabajo, incluyendo todos los eventos planificados en PROBLAM. Asimismo, debe fomentar un buen ambiente de trabajo y actuar como referente para resolver impedimentos, solicitando ayuda del profesor cuando no lo logre.

IV-B. Eventos

La dinámica de trabajo en PROBLAM se estructura en *sprints* de 4-5 semanas de duración. El objetivo de cada *sprint* es construir un nuevo incremento del producto que satisfaga los requisitos establecidos en el pliego. Además, actúa como contenedor de otros eventos, diseñados para que el equipo mantenga un ritmo de trabajo constante, identifique avances e impedimentos y actúe en consecuencia. El capitán es responsable de organizar los eventos y, a su finalización, completar un acta orientada a asegurar la trazabilidad del trabajo realizado y proporcionar visibilidad sobre la evolución del equipo a lo largo del proyecto. Los eventos de planificación, sincronización, retrospectiva y revisión están inspirados en Scrum [3], mientras que la tutoría es un evento de naturaleza completamente académica.

Planificación: es el primer evento de cada *sprint* y tiene como objetivo distribuir el trabajo entre los miembros del equipo, asignando responsables y fechas de vencimiento a cada uno de los trabajos propuestos en la guía, publicada previamente por el profesor junto con la nueva versión del pliego. La asignación de trabajos se realiza por consenso y, en algunos casos, pueden dividirse para facilitar su ejecución. El resultado de esta reunión es la planificación del *sprint*, que se refleja en el cuaderno de trabajo.

Sincronización: es una reunión semanal breve, en la que cada miembro del equipo expone sus avances y bloqueos desde la última reunión y su plan de trabajo hasta la siguiente. Como resultado, el equipo actualiza su conocimiento sobre el estado del *sprint* y, si es necesario, replanifica el trabajo. Si el equipo detecta un impedimento técnico u organizativo que bloquee su avance, el capitán solicita una tutoría.

Retrospectiva: es el último evento del *sprint* y se celebra inmediatamente antes de la entrega del incremento. Su objetivo es fomentar una reflexión colectiva sobre qué se ha hecho bien, qué aspectos pueden mejorarse y qué cambios se implementarán en el siguiente *sprint*. Como resultado, el

equipo define hasta 3 acciones de mejora, que se registran en el cuaderno de trabajo.

Revisión: es el último evento del proyecto, al que solo pueden acceder los equipos que hayan construido un producto potencialmente aceptable; es decir, que satisfaga los criterios de aceptación establecidos. La revisión se desarrolla en dos fases: (i) el equipo expone su dinámica de trabajo durante el proyecto y, (ii) el profesor formula cuestiones para recabar las evidencias necesarias para evaluar tanto el producto como el proceso seguido por el equipo. Finalizada la revisión, el profesor determina la evaluación final, considerando las valoraciones de los incrementos entregados en cada sprint y las evidencias obtenidas durante este evento.

Tutoría: es un evento que se produce a demanda del equipo, cuyo propósito es resolver los bloqueos que dificultan su avance. Cada equipo dispone de una tutoría por *sprint*, aunque puede solicitar una segunda, justificando su necesidad. En este caso, el profesor decide si procede realizarla o si el equipo debe plantear sus consultas a través de otros canales disponibles en la asignatura. En ningún caso, el profesor ofrece soluciones directas en la tutoría, sino que guía al equipo para que las encuentre por sí mismo.

IV-C. Artefactos

Los artefactos en PROBLAM estructuran el trabajo del equipo, aseguran la trazabilidad del proceso y proporcionan información clave para la evaluación del aprendizaje. Al igual que en Scrum, los artefactos mejoran la transparencia dentro del equipo y son esenciales para su adaptación durante todo el proyecto. En este caso, solo el incremento es comparable a un artefacto Scrum, aunque la guía de trabajo también adapta características de los *backlogs* [3].

Pliego: describe el proyecto en lenguaje natural, proporcionando información suficiente del dominio para que los estudiantes identifiquen los requisitos técnicos y los enmarquen en el contexto de negocio. El pliego se actualiza conforme evoluciona el alcance del proyecto en cada *sprint*, e incluye la descripción del incremento y las rúbricas para su valoración.

Guía de trabajo: proporciona el *scaffolding* necesario para que el equipo aborde el trabajo requerido para construir un nuevo incremento. Para ello, la guía describe todos los trabajos a realizar en el *sprint*, explicando su propósito, detallando sus objetivos y especificando los resultados esperados. El profesor actualiza la guía de trabajo en cada *sprint* para asegurar su alineamiento con la nueva versión del pliego.

Cuaderno de trabajo: es el artefacto principal para la organización del equipo, donde se registra y mantiene el plan de trabajo y se documenta el esfuerzo realizado durante el *sprint*. Todos los miembros del equipo son responsables de mantenerlo actualizado en base a su trabajo, para garantizar la transparencia y proporcionar una visión actualizada del estado del proyecto en todo momento. El cuaderno de trabajo se incluye en todas las actas, para garantizar la trazabilidad del proceso y proporcionar información potencialmente evaluable.

Informe de valoración: busca que los estudiantes analicen, de forma crítica y a nivel individual, aspectos como el compromiso, la iniciativa, la comunicación y el esfuerzo de los

miembros del equipo, incluyéndose a sí mismos. Su propósito es proporcionar *feedback* sobre la dinámica del equipo a partir de valoraciones anónimas realizadas por sus integrantes, que se entregan al equipo de forma agregada. Además, el informe proporciona al profesor información relevante sobre el equipo, ayudándole a identificar posibles conflictos, desigualdades en la carga de trabajo o cualquier otra situación que pueda comprometer el aprendizaje efectivo de sus miembros.

Incremento: es el resultado del *sprint* y refleja el avance realizado por el equipo en la construcción del producto. El incremento incluye los resultados de todos los trabajos planificados en el *sprint*, el cuaderno de trabajo y el informe de valoración. Al igual que en Scrum, los incrementos son aditivos: cada nuevo incremento se añade a los de los *sprints* anteriores, y se evalúan conjuntamente respecto a las rúbricas establecidas en el pliego. Un incremento se acepta si los resultados incluidos cumplen los criterios de aceptación.

V. CASO DE ESTUDIO

El proyecto y la metodología de trabajo fueron evaluados en Sistemas de Bases de Datos (SBD), asignatura de 2º curso del Grado en Ingeniería Informática de Servicios y Aplicaciones de la Universidad de Valladolid. En la asignatura se combinaron clases magistrales teórico-prácticas y actividades regulares de resolución de problemas, para que los estudiantes adquirieran los *conocimientos y habilidades* necesarias para abordar el proyecto. Así, el proyecto cumplió un rol vertebrador, permitiendo a los estudiantes integrar dichos conocimientos y habilidades en un contexto alineado con escenarios reales de la industria del software.

El proyecto comenzó la primera semana del curso y su desarrollo se alineó con el resto de actividades de la asignatura [16]. Las entregas del proyecto se coordinaron con los exámenes parciales, permitiendo a los estudiantes aprovechar las sinergias entre ambas pruebas de evaluación. Así, el proyecto reforzó el aprendizaje adquirido en clase y promovió un enfoque más activo del proceso de aprendizaje, en el que los estudiantes asumieron un rol más autónomo. Esta dinámica mejoró su comprensión de los conceptos y fortaleció su capacidad de adaptación a contextos evolutivos.

Los 62 matriculados en SBD (45 hombres y 17 mujeres) se distribuyeron en 13 equipos de trabajo: 10 equipos de 5 miembros y 3 equipos de 4 miembros, cuya configuración fue decidida por los propios estudiantes. Todos los equipos abordaron PLAYROCK en condiciones comparables, aunque su alcance se ajustó para los equipos de 4 miembros, asegurando una carga de trabajo equitativa para todos los estudiantes.

El apartado V-A presenta brevemente el proyecto PLAYROCK, y a continuación, los apartados V-B y V-C exponen los resultados académicos obtenidos y las valoraciones de los estudiantes.

V-A. PLAYROCK

PLAYROCK es una red social diseñada para fans de la música rock, donde los usuarios pueden seguir a sus artistas favoritos, descubrir nuevas canciones y compartir su pasión por el género. La plataforma permite gestionar colecciones

TABLA III
EVOLUCIÓN DEL ALCANCE DE PLAYROCK

Tarea	Ítem	Incrementos		
		<i>Sprint #1</i>	<i>Sprint #2</i>	<i>Sprint #3</i>
#1	Entidades	10	19	19
	Relaciones	14	24	24
	IS-As	0	2	2
	Atributos	59	84	84
#2	Tablas	18	29	29
	FKs	23	41	41
#3	Tablas	18	29	29
	FKs	23	41	41
	Checks	0	5	23
	Triggers	0	6	11
	Privilegios	0	0	50
#4	Tablas creación	0	4	4
	Tablas copia	0	20	25
#5	Funcionalidades	0	25	50

de discos y *playlists*, reproducir canciones, recibir recomendaciones y, en definitiva, mantenerse al tanto de las novedades del mundo del rock. El diseño de PLAYROCK se inspira en *Spotify* [18], proporcionando a los estudiantes una referencia conocida que les ayude a comprender el proyecto.

Durante el curso se publicaron tres versiones del pliego (semanas 1, 6 y 11), reflejando una evolución progresiva del alcance del proyecto. La tabla III ilustra cómo los diferentes ítems de evaluación evolucionaron *sprint* a *sprint* (las celdas sombreadas indican el *sprint* en el que cada ítem llegó a su alcance máximo). Por ejemplo, el diseño lógico alcanzó su máximo en el *sprint* #2, con 29 tablas y 41 claves foráneas (FKs), de modo que en el *sprint* #3 los estudiantes siguieron trabajando sobre el mismo alcance, mejorando su diseño en un proceso iterativo, pero no incremental.

El resto de los artefactos¹ también se actualizaron en cada *sprint*: la guía de trabajo se presentó en un tablero Scrum, el cuaderno de trabajo como una hoja de cálculo (con un pequeño *dashboard* integrado) y el informe de valoración mediante un formulario. Todos los equipos dispusieron de un espacio de trabajo compartido con chat incorporado, diseñado para facilitar la colaboración y almacenar los resultados consolidados en cada incremento. Finalmente, el desarrollo se llevó a cabo en un sistema MariaDB centralizado, donde cada equipo disponía de su propia base de datos.

El profesor, como miembro de los equipos, tenía acceso a todos sus recursos compartidos, pero solo lo usaba a solicitud del capitán o para propósitos de evaluación.

V-B. Resultados académicos

La mejora continua y la atención a la calidad del producto son principios clave de Scrum, que también se reflejan en PROBLAM. Como se observa en la Figura 2, la evaluación de los incrementos muestra una tendencia creciente en todas las tareas, excepto en la construcción de la base de datos en el *sprint* #2. Eso se debió a que varios equipos no entregaron los resultados de la implementación de *triggers*, al priorizar otras tareas con mayor peso en el segundo examen parcial. La

situación se revirtió en el *sprint* #3, donde se puede observar que la evaluación de todas las tareas situó su mediana en torno a los 8 puntos. Cabe destacar la mejora en la evaluación de las funcionalidades, cuya mediana prácticamente se duplicó en el *sprint* #3, a pesar del incremento en su alcance y complejidad.

El crecimiento observado en las calificaciones refleja una mejora en las habilidades técnicas del estudiantado, pero el aprendizaje en un entorno colaborativo también depende de la organización y la dinámica del equipo. Para comprender esta influencia, hemos analizado los cuadernos de trabajo y los informes de valoración con el objetivo de identificar los factores que impactan en el desempeño de los equipos. Dado que los informes incluyen respuestas en escala de Likert, se ha utilizado la mediana como estimador de agregación [19] y se han aplicado métodos no paramétricos, particularmente el coeficiente de correlación de Spearman. La Figura 3 presenta los mapas de calor obtenidos de los datos recogidos en los *sprints* #1 y #3, mostrando cómo han evolucionado las relaciones entre factores del trabajo en equipo y su impacto en el desempeño de los estudiantes. Los resultados del *sprint* #1 muestran que los equipos con colaboraciones previas reportaron percepciones más positivas en aspectos como el compromiso y la iniciativa de sus miembros. Además, se observa una fuerte correlación entre la percepción del esfuerzo individual, la valoración del compromiso y la satisfacción con los eventos celebrados. Estos eventos parecen jugar un papel clave en cómo los equipos experimentan PROBLAM, influyendo tanto en la valoración de la experiencia de trabajo como en la percepción del ajuste a la metodología. Otro hallazgo relevante es la alta correlación entre la comunicación y la percepción de la experiencia de trabajo, la generación de un buen ambiente y el nivel de compromiso dentro del equipo. Esto sugiere que la comunicación efectiva es fundamental en la dinámica grupal desde las primeras etapas del proyecto. En general, los resultados indican que, en este primer *sprint*, los equipos más organizados y con una percepción positiva de su esfuerzo y compromiso lograron mejores experiencias de trabajo. En el *sprint* #3, las relaciones entre los factores del trabajo en equipo evolucionaron significativamente, consolidando una alta correlación entre aspectos clave como la experiencia de trabajo, el compromiso, la iniciativa, el ambiente de trabajo, el esfuerzo, la comunicación y la comprensión de las propuestas. Esto sugiere que los equipos alcanzaron un mayor grado de madurez, lo que favoreció su cohesión grupal y mejoró el desempeño colectivo. Otro hallazgo relevante es la fuerte correlación negativa entre el tiempo dedicado al *sprint* #3 y la valoración de la dinámica de trabajo, en contraste con el *sprint* #1, donde esta relación era positiva. Esto evidencia que, al inicio del proyecto, los equipos necesitaron más tiempo para familiarizarse con la metodología y las dinámicas de trabajo, pero, una vez asimiladas, optimizaron su gestión del tiempo y mejoraron la eficiencia de su trabajo.

Como resultado final, 9 equipos entregaron productos aptos, 3 no aptos y 1 equipo no entregó el incremento final. Estos resultados son consistentes con la progresión observada en la dinámica de trabajo, y en el caso de los equipos no aptos, con

¹Todos los artefactos utilizados en PLAYROCK están disponibles en <https://github.com/migumar2/PlayRock>

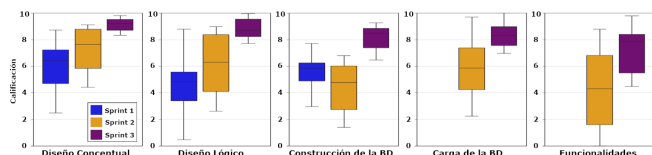


Figura 2. Distribución de las calificaciones por objetivos.

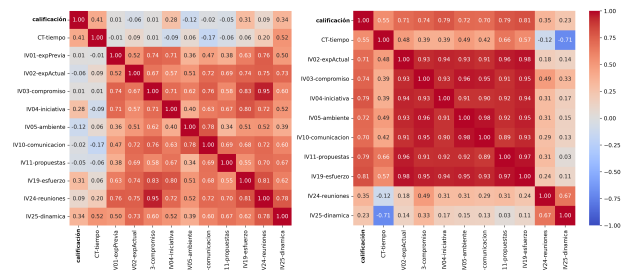


Figura 3. Análisis de correlación: Sprint #1 vs. Sprint #3

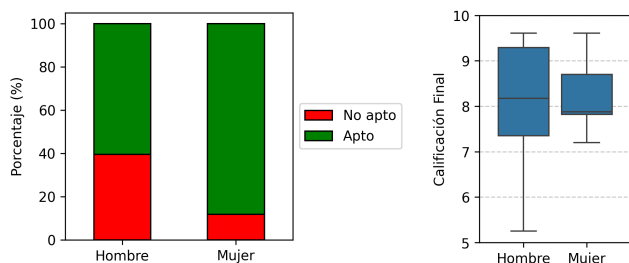


Figura 4. Resultados académicos individuales

las bajas valoraciones en aspectos clave del trabajo en equipo.

Para finalizar, completamos nuestro estudio con un análisis del rendimiento individual de los estudiantes, desagregado por género. La Figura 4 (izq.) muestra que el 60 % de los hombres y el 90 % de las mujeres fueron calificados como aptos, alcanzando una tasa de rendimiento global del 70 %. La Figura 4 (der.) profundiza en estas calificaciones, revelando que las medianas de hombres y mujeres son comparables. Sin embargo, en el caso de las mujeres, las calificaciones tienden a concentrarse en el primer cuartil, mientras que en los hombres presentan una mayor variabilidad, oscilando entre el umbral de apto y el máximo alcanzado en la asignatura, que es el mismo para ambos géneros.

V-C. Valoración de los estudiantes

Al final del proyecto, 38 de los 62 estudiantes participaron en una encuesta voluntaria de valoración. El 92 % consideró que PLAYROCK contribuyó de manera alta (4) o muy alta (5) a la activación de los conocimientos adquiridos en la asignatura, mientras que el 79 % valoró positivamente (4) o muy positivamente (5) su organización en *sprints*. Asimismo, el 95 % de los estudiantes otorgó una valoración alta (4) o muy alta (5) al sistema de evaluación incremental de PROBLAM, y el 82 % destacó la utilidad del *feedback* recibido con la misma puntuación. En definitiva, los resultados reflejan una valoración positiva del proyecto por parte del estudiantado, cuyas respuestas abiertas avalan también los objetivos perseguidos: “PLAYROCK me ha ayudado mucho a la hora de preparar la

asignatura y llevarla al día”, “me servía de aprendizaje para los exámenes”, “hemos aprendido a diseñar y realizar una base de datos en un trabajo de verdad”, “es el trabajo en el que más hemos aprendido cómo se trabajará en la empresa en un futuro”, “ha sido un aprendizaje dinámico y muy gratificante” y, como conclusión, “será difícil para nosotros olvidarnos de PLAYROCK tan rápidamente como lo hacemos con otros trabajos :)”.

Complementariamente, los estudiantes también indicaron en la encuesta que PLAYROCK había contribuido al desarrollo de habilidades transversales: el 90 % de los estudiantes afirmó haber mejorado su responsabilidad, el 82 % su paciencia y capacidad de resolución de problemas, y el 70 % su capacidad de toma de decisiones, entre otras habilidades clave.

VI. CONCLUSIONES Y TRABAJO FUTURO

Las dinámicas de aprendizaje ágiles son un método efectivo para la adquisición de competencias en bases de datos, dada la naturaleza iterativa e incremental que comparten su ciclo de vida y marcos de trabajo ágiles como Scrum. Por ello, hemos diseñado un proyecto genérico que cubre las competencias principales del ciclo de vida de una base de datos y una metodología (PROBLAM) que lo desarrolla siguiendo una dinámica ágil. Los resultados obtenidos muestran una adquisición progresiva de las competencias esperadas en la materia, alcanzando resultados académicos notables al final del proyecto. Los estudiantes valoran la estructura iterativa del proyecto y la retroalimentación continua como aspectos clave para su aprendizaje. Además, se evidencia que la mejora en la cohesión de los equipos y la mayor madurez que alcanzan al final del proyecto no solo contribuyen a un aprendizaje más profundo, sino que fortalecen habilidades esenciales como la organización, la comunicación y la toma de decisiones. Estos hallazgos refuerzan la necesidad de incorporar enfoques de aprendizaje activo, que mejoren el alineamiento de la formación universitaria y las prácticas del sector profesional en el ámbito de las bases de datos, favoreciendo un aprendizaje más auténtico y transferible a entornos reales.

Nuestra primera línea de trabajo futuro se centra en ampliar el proyecto para abordar otras etapas del ciclo de vida de una base de datos, como el diseño físico o la gestión transaccional, e incorporar tareas relacionadas con analítica (OLAP) o ingeniería de datos. Así, el proyecto podría trascender a una segunda asignatura de bases de datos, estructurando la formación de los estudiantes en torno a un proyecto común y reforzando, con ello, la continuidad del aprendizaje. Además, estamos explorando la aplicación de PROBLAM en asignaturas de otras materias que desarrollan actividades de ABP orientadas al sector profesional.

AGRADECIMIENTOS

Este trabajo está vinculado al PID UVAGILE *reloaded: automatización basada en el producto de aprendizaje* (desarrollado en el marco del GID Educación ágil y tecnologías aplicadas a la mejora de los procesos de enseñanza-aprendizaje) subvencionado por el Vicerrectorado de Innovación Docente y Transformación Digital de la Universidad de Valladolid.

REFERENCIAS

- [1] D. Perkins, *Making Learning Whole: How Seven Principles of Teaching Can Transform Education*. San Francisco, CA: Jossey-Bass, 2009.
- [2] T. Connolly y C. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson, 2015.
- [3] K. Schwaber y J. Sutherland, "The Scrum Guide," <https://scrumguides.org>, 2020.
- [4] W. Wijnands y R. van Solingen, "Transforming Education with eduScrum," en *Trends in E-learning*. Springer, 2019, pp. 63–78.
- [5] A. N. Kumar, R. K. Raj, S. G. Aly, M. D. Anderson, B. A. Becker, R. L. Blumenthal, E. Eaton, S. L. Epstein, M. Goldweber, P. Jalote, D. Lea, M. Oudshoorn, M. Pias, S. Reiser, C. Servin, R. Simha, T. Winters, y Q. Xiang, *Computer Science Curricula 2023*. New York, NY, USA: Association for Computing Machinery, 2024.
- [6] V. Ng, C. Lau, y P. Shum, "Multi-Disciplinary Learning through a Database Development Project," *Electronic Journal of e-Learning*, vol. 10, no. 4, pp. 417–427, 2012.
- [7] M. Ishaq, A. Abid, M. S. Farooq, M. F. Manzoor, U. Farooq, K. Abid, y M. A. Helou, "Advances in database systems education: Methods, tools, curricula, and way forward," *Education and Information Technologies*, vol. 28, no. 3, pp. 2681–2725, 2023.
- [8] J. S. Krajcik y N. Shin, "Project-Based Learning," en *The Cambridge Handbook of the Learning Sciences*, R. K. Sawyer, Ed. New York: Cambridge University Press, 2014, pp. 275–297.
- [9] P. Letelier Torres y H. Cornide Reyes, "Evaluación de equipos de trabajo en el marco de Aprendizaje Basado en Proyectos y Métodos Ágiles," *Actas de las JENUI*, vol. 5, pp. 165–172, 2020.
- [10] J. Ma, "Problem-based learning with database systems," *Computers & Education*, vol. 22, no. 3, pp. 257–263, 1994.
- [11] C. Domínguez y A. Jaime, "Database design learning: A project-based approach organized through a course management system," *Computers & Education*, vol. 55, no. 3, pp. 1312–1320, 2010.
- [12] J. Van de Pol, M. Volman, y J. Beishuizen, "Scaffolding in teacher-student interaction: A decade of research," *Educational Psychology Review*, vol. 22, pp. 271–296, 2010.
- [13] P. Salza, P. Musmarra, y F. Ferrucci, "Agile methodologies in education: A review," *Agile and lean concepts for teaching and learning: Bringing methodologies from industry to the classroom*, pp. 25–45, 2019.
- [14] M. Bwye, I. Arrans, J. M. García, C. Müller, y J. A. Parejo, "Aprendiendo metodologías ágiles jugando con scrumprg," en *Actas de las JENUI - Vol. 9 (2024)*, 2024, pp. 101 – 108.
- [15] G. Wiggins, *Educative assessment: Designing assessments to inform and improve student performance*. Jossey-Bass Publishers, 1998.
- [16] M. A. Martínez Prieto, J. Silvestre, y R. Pinedo González, "Visibilizando el aprendizaje a través de la autoevaluación," *Actas de las JENUI*, vol. 8, pp. 51–58, 2024.
- [17] J. Biggs, "Enhancing teaching through constructive alignment," *Higher education*, vol. 32, no. 3, pp. 347–364, 1996.
- [18] A. Dixit, "Database Design for Spotify," <https://medium.com/towards-data-engineering/design-the-database-for-a-system-like-spotify-95ffd1fb5927>, 2023, Último acceso, febrero 2025.
- [19] S. Jamieson, "Likert scales: How to (ab) use them?" *Medical education*, vol. 38, no. 12, pp. 1217–1218, 2004.