



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**Captura y procesamiento de métricas de aprendizaje
en eXtreme Learning**

Alumno: Samuel Vicente Sánchez

**Tutores: Miguel Ángel Martínez Prieto
María Jesús Rodríguez Triana**

Fecha: 10 de julio de 2025

Captura y procesamiento de métricas de aprendizaje en eXtreme Learning

Samuel Vicente Sánchez

10 de julio de 2025

*A mis padres y mi hermana.
Sin ellos no podría haber llegado hasta aquí.*

*El éxito es la suma
de pequeños esfuerzos
repetidos día tras día.*

Robert Collier.

Agradecimientos

Quiero agradecer a todas las personas que me han acompañado, no solo en la realización de este Trabajo Fin de Grado, sino también durante estos cinco años que ha durado la carrera.

Quiero empezar agradeciendo a mis tutores Miguel Ángel Martínez Prieto y María Jesús Rodríguez Triana por ayudarme en todo momento con el desarrollo de este proyecto. Gracias por el constante apoyo, la solución de todos los problemas y dudas, y sobre todo por brindarme la oportunidad de trabajar con ellos en este proyecto en el que he aprendido mucho.

También quiero agradecer a mi familia, en especial a mis padres y mi hermana, que siempre me han animado y apoyado a estudiar esta carrera y han estado en todo momento a mi lado. Siempre estaré agradecido por haberme facilitado el seguir progresando en mi vida, comenzando por ayudarme económicamente para vivir en Valladolid y, posteriormente, en Segovia.

Quería agradecer también a todos los amigos que hice al principio de la carrera y que me han acompañado durante estos cinco años, por compartir este camino tan bonito y hacer que la universidad no haya sido solo estudio y exámenes, sino también diversión y alegría en todo momento. He sido muy feliz con vosotros durante estos años y espero que siga siendo así para siempre. En especial, quiero agradecer a mis compañeros de piso Marcos, David, Pedro e Inés, que me han aguantado día a día durante estos últimos años. Gracias.

No me olvido, obviamente, de toda la gente que he conocido estos dos últimos años en Segovia. Aún viniendo después de que todo el mundo tuviera un grupo formado, se nos acogió de la mejor manera posible, convirtiéndonos en una gran familia. Como dijimos en el discurso de graduación este año, que sirvió para cerrar esta etapa, no esperábamos mucho de esta ciudad, pero nos ha dado uno de los mejores años de nuestras vidas. Entre todos los que han estado a mi lado en Segovia, quiero agradecer a Itziar por ser la mejor amiga que nadie podría pedir nunca, por estar siempre ahí para todo lo que necesito y, sobre todo, por acompañarme en todos los planes que han hecho que estos dos últimos años hayan sido tan especiales, siendo la mejor forma de acabar esta carrera.

A todos vosotros, gracias de corazón.

Resumen

Las metodologías ágiles están cada vez más presentes en el ámbito laboral y, como consecuencia, en el académico. Pero los profesores que aplican este tipo de metodologías, como por ejemplo *eXtreme Learning* o en concreto *Test-Driven Learning*, deben invertir demasiado tiempo procesando y analizando todos los datos generados por los estudiantes.

Por ello, en este proyecto planteamos una forma de automatizar el procesamiento de estos datos que se generan durante el proceso enseñanza-aprendizaje en una asignatura universitaria. Primero, se han adaptado los conceptos de una metodología ágil diseñada para este tipo de docencia de forma que se puedan representar con un modelo conceptual. Además, se ha preparado una aplicación funcional que permite gestionar y procesar toda la información necesaria para el desarrollo normal de la asignatura, así como la consulta de estos datos generados, diseñando para ello una herramienta de *Learning Analytics*.

Hemos conseguido así construir una herramienta que le permite al profesor reducir en gran medida el tiempo que utiliza para el procesamiento y análisis de estos datos, de forma que pueda dedicar más cantidad de tiempo a la personalización del aprendizaje para los alumnos que cursen dicha asignatura.

Palabras claves: *Learning Analytics*, *eXtreme Learning*, *Test-Driven Learning*, procesamiento de datos, metodologías ágiles, enseñanza-aprendizaje.

Abstract

Agile methodologies are increasingly present in the workplace and, as a result, in learning. However, teachers who apply these methodologies, such as *eXtreme Learning* and, in particular *Test-Driven Learning*, often need to invest a considerable amount of time processing and analyzing all the data generated by students.

For this reason, this project proposes a way to automate the processing of the data generated during the teaching and learning process in a university subject. First, the concepts of an agile methodology designed for this type of teaching have been adapted so that they can be represented with a conceptual model. In addition, a functional application has been developed that enables the management and processing of all the information required for the normal development of the subject, as well as the consultation of the generated data, through the design of a *Learning Analytics* tool.

We have thus succeeded in building a tool that allows teachers to significantly reduce the time spent on processing and analyzing this data, so they can devote more time to personalizing the learning experience for the students enrolled in the course.

Keywords: *Learning Analytics*, *eXtreme Learning*, data processing, agile methodologies, teaching-learning.

Índice general

Lista de figuras	v
Lista de tablas	ix
I Descripción del proyecto	1
1. Introducción	3
1.1. Planteamiento del problema	4
1.2. Objetivos	5
1.3. Condicionantes	5
1.3.1. Restricciones	6
1.4. Alcance del proyecto	6
1.5. Estructura de la memoria	7
2. Gestión del proyecto	9
2.1. Metodología	9
2.1.1. Paquetes de trabajo	10
2.1.2. Roles	11
2.1.3. Eventos	11
2.1.4. Artefactos	12
2.1.5. Entorno de trabajo	13
2.2. Riesgos	14
2.3. Planificación temporal	14
2.3.1. <i>Sprint</i> #1	15
2.3.2. <i>Sprint</i> #2	16
2.3.3. <i>Sprint</i> #3	18
2.3.4. <i>Sprint</i> #4	19
2.4. Presupuestos	21
2.4.1. <i>Hardware</i>	21
2.4.2. <i>Software</i>	23
2.4.3. Recursos humanos	23
2.4.4. Presupuesto total	25
2.5. Balance temporal y económico	25
2.5.1. Balance temporal	25
2.5.2. Balance económico	28

3. Antecedentes	33
3.1. Contexto educativo del trabajo	33
3.1.1. eXtreme Learning	33
3.1.2. Test-Driven Learning (TDL)	37
3.1.3. Retos de <i>eXtreme Learning</i>	38
3.2. Fundamentos teóricos	38
3.2.1. Learning Analytics	39
3.2.2. Multimodal Learning Analytics (MMLA)	40
3.2.3. Analytics layers for learning design	43
3.3. Estado del arte	44
3.3.1. Descripción de trabajos relacionados	45
3.3.2. Discusión	48
3.4. Fundamentos técnicos	50
3.4.1. ETL	50
3.4.2. Alternativas para el almacenamiento de datos	52
3.4.3. Backend	53
3.4.4. Frontend	54
 II Desarrollo de la solución	 55
4. Análisis	57
4.1. Actores	57
4.2. Requisitos de usuario	57
4.3. Requisitos funcionales	73
4.4. Requisitos no funcionales	73
4.5. Requisitos de interfaz externa	74
4.6. Diseño conceptual	79
4.6.1. Modelo de dominio	79
4.6.2. Diccionario de datos	80
4.7. Mapeo de los orígenes de datos	96
 5. Diseño	 103
5.1. Arquitectura física	103
5.2. Arquitectura lógica	104
5.3. Estructura lógica del almacenamiento	105
5.4. Interfaz de usuario	108
 6. Implementación	 115
6.1. Herramientas	115
6.2. Preparación de los datos	117
6.3. Servicios programados	118
6.4. Construcción del dashboard	121

7. Pruebas	123
7.1. Pruebas con fechas	123
7.2. Pruebas para la carga de archivos	131
7.3. Pruebas para eliminación de información	135
 III Resultados	 137
8. Aceptación	139
8.1. Objetivo 1: Modelo Conceptual	139
8.2. Objetivo 2: Mecanismo de Gestión Automatizada	140
8.3. Objetivo 3: Solución de <i>Learning Analytics</i>	143
9. Conclusiones y trabajo futuro	147
9.1. Conclusiones	147
9.1.1. Perspectiva del proyecto	147
9.1.2. Perspectiva personal	148
9.2. Trabajo futuro	149
 Bibliografía	 151
 IV Apéndices	 153
A. Manual de Instalación	155
A.1. Requisitos previos	155
A.1.1. PyCharm	155
A.1.2. Python	156
A.2. Preparación del entorno	157
A.2.1. Crear entorno de Python	157
A.2.2. Instalar dependencias	158
A.2.3. Definir configuración de ejecución	158
A.3. Ejecución de la aplicación	159
A.3.1. Despliegue del backend	159
A.3.2. Despliegue del frontend	160
 B. Manual de Usuario	 161
B.1. Configuración inicial	162
B.2. Crear producto y proceso de aprendizaje	166
B.2.1. Gestionar producto de aprendizaje	166
B.2.2. Gestionar acciones	168
B.2.3. Crear plantillas y preguntas	169
B.3. Procesamiento de datos	171
B.4. Gestión avanzada	173

Índice de figuras

1.1.	Diagrama de características de la aplicación.	6
1.2.	Workflow de la parte de Procesamiento de datos.	7
2.1.	Planificación del sprint #1.	17
2.2.	Planificación del sprint #2.	18
2.3.	Planificación del <i>sprint</i> #3.	19
2.4.	Planificación del <i>sprint</i> #4.	21
2.5.	Balance del sprint #1.	26
2.6.	Balance del sprint #2.	27
2.7.	Balance del sprint #3.	28
2.8.	Balance del sprint #4.	29
3.1.	Ejemplo del producto de aprendizaje de la asignatura de <i>Sistemas de Bases de Datos</i> . Dada una visión general de la asignatura, encontramos los 5 objetivos en los que se divide. Además, podemos ver que el objetivo de diseño lógico se subdivide en tres historias de aprendizaje. Una de esas historias (Modelo Relacional) se descompone a su vez en 5 criterios que determinan el alcance de dicha historia. [11].	35
3.2.	Proceso de aprendizaje de Sistemas de Bases de Datos [11].	36
3.3.	<i>Data Value Chain</i> : etapas [13].	41
3.4.	<i>Multimodal Data Value Chain</i> : etapas [21].	41
3.5.	Representación gráfica del <i>framework</i> de A14LD [6].	44
3.6.	Ejemplo de un informe generado por la herramienta GLIMPSE [16].	46
3.7.	Arquitectura de PAMEL [22].	47
3.8.	Interfaz de la aplicación desarrollada en <i>Workplace Learning Analytics</i> [25]. . . .	48
4.1.	Diagrama de casos de uso.	58
4.2.	Estructura de carpetas descargada de <i>Moodle</i>	75
4.3.	Archivos contenidos en una carpeta de una entrega de <i>Moodle</i>	75
4.4.	Estructura de la hoja de cálculo de autoevaluaciones.	76
4.5.	Estructura de la hoja de cálculo de tareas.	77
4.6.	Estructura de la hoja de cálculo de resultados.	77
4.7.	Estructura de la hoja de cálculo de evaluaciones.	78
4.8.	Estructura de la hoja de cálculo de estudiantes.	78
4.9.	Modelo de dominio del proyecto.	79
4.10.	Diccionario de datos de la entidad ACCIÓN.	81
4.11.	Diccionario de datos de la entidad APRENDIZAJE.	81

4.12. Diccionario de datos de la entidad CRITERIO.	82
4.13. Diccionario de datos de la entidad CURSO.	82
4.14. Diccionario de datos de la entidad ESTUDIANTE.	82
4.15. Diccionario de datos de la entidad EVALUACIÓN.	83
4.16. Diccionario de datos de la entidad HISTORIA.	83
4.17. Diccionario de datos de la entidad OBJETIVO.	84
4.18. Diccionario de datos de la entidad PLANTILLA.	84
4.19. Diccionario de datos de la entidad PREGUNTA.	85
4.20. Diccionario de datos de la entidad RESPUESTA.	86
4.21. Diccionario de datos de la entidad RESULTADO.	87
4.22. Diccionario de datos de la entidad SPRINT.	87
4.23. Diccionario de datos de la entidad TAREA.	88
4.24. Diccionario de datos de la relación ASOCIAR.	89
4.25. Diccionario de datos de la relación AUTOEVALUAR.	89
4.26. Diccionario de datos de la relación CONTENER.	90
4.27. Diccionario de datos de la relación CORRESPONDER.	90
4.28. Diccionario de datos de la relación DESARROLLAR.	90
4.29. Diccionario de datos de la relación ENSEÑAR.	90
4.30. Diccionario de datos de la relación EVALUAR.	91
4.31. Diccionario de datos de la relación HACER.	91
4.32. Diccionario de datos de la relación INCLUIR.	91
4.33. Diccionario de datos de la relación INSTANCIAR.	92
4.34. Diccionario de datos de la relación IS_A_ACCIÓN.	92
4.35. Diccionario de datos de la relación MATRICULAR.	92
4.36. Diccionario de datos de la relación MEDIR.	93
4.37. Diccionario de datos de la relación OBTENER.	93
4.38. Diccionario de datos de la relación ORGANIZAR.	93
4.39. Diccionario de datos de la relación PLANIFICAR.	93
4.40. Diccionario de datos de la relación PLANTEAR.	94
4.41. Diccionario de datos de la relación PROGRAMAR.	94
4.42. Diccionario de datos de la relación RESOLVER.	94
4.43. Diccionario de datos de la relación SUPERAR.	94
4.44. Diccionario de datos de la relación VALORAR.	95
4.45. Mapeo lógico de la fuente de datos: AUTOEVALUACIONES.	96
4.46. Mapeo lógico de la fuente de datos: CUADERNOS DE TRABAJO.	96
4.47. Mapeo lógico de la fuente de datos: LISTADO DE ESTUDIANTES.	97
4.48. Mapeo lógico de la fuente de datos: EVALUACIONES.	97
4.49. Mapeo lógico de la fuente de datos: FORMULARIOS.	97
4.50. Mapeo lógico de la fuente de datos: OBJETIVOS DE LA ASIGNATURA.	98
4.51. Mapeo lógico de la fuente de datos: TABLERO DE TRELLO.	99
4.52. Mapeo lógico de los datos insertados MANUALMENTE.	100
4.53. Mapeo lógico de los datos insertados AUTOMÁTICAMENTE.	101
5.1. Arquitectura física de la aplicación.	104
5.2. Diagrama de despliegue de la aplicación.	104
5.3. Arquitectura lógica de la aplicación.	105

5.4. Página principal de la aplicación.	109
5.5. Página de selección previa.	109
5.6. Página de creación de <i>sprints</i>	110
5.7. Página para la planificación de historias en <i>sprints</i>	110
5.8. Página para la carga y procesamiento de archivos.	111
5.9. Página para la visualización de los elementos del producto y proceso de aprendizaje.	112
5.10. Página de visualización de los <i>dashboards</i>	113
7.1. Error obtenido al introducir una fecha de inicio posterior a una de finalización al crear un curso.	124
7.2. Error obtenido al introducir una fecha de inicio posterior a una de finalización al crear un sprint.	125
7.3. Error obtenido al introducir fechas externas a las indicadas por el curso.	127
7.4. Error obtenido al introducir una fecha que pertenece al rango temporal de otro sprint.	129
7.5. Advertencia y limitación al definir las fechas de una acción.	131
7.6. Error obtenido al cargar un archivo de estudiantes erróneo.	132
7.7. Error obtenido al cargar un archivo que no tiene la extensión “.zip”.	134
7.8. Error obtenido al cargar un archivo que no tiene la extensión “.xlsx”.	135
7.9. Error obtenido al tratar de eliminar un elemento con dependencias.	136
8.1. Interfaz del formulario de evaluación para el sistema de gestión.	141
8.2. Sección del formulario dedicada a evaluar el procesamiento automático.	141
8.3. Sección del formulario dedicada a preguntas de respuesta abierta.	142
8.4. Resultados obtenidos en el formulario que evalúa el objetivo 2.	143
8.5. Primera parte del cuestionario EFLA utilizado para la evaluación	144
8.6. Segunda parte del cuestionario EFLA utilizado para la evaluación	145
8.7. Segunda parte del cuestionario EFLA utilizado para la evaluación	146
A.1. Proceso de instalación de PyCharm.	156
A.2. PyCharm instalado y proyecto abierto.	157
A.3. Selección de versión de Python.	158
A.4. Definición del entorno de ejecución.	159
A.5. Texto mostrado en terminal al ejecutar.	160
B.1. Primera parte de la página principal.	161
B.2. Segunda parte de la página principal.	162
B.3. Selección de fechas del curso.	163
B.4. Creación de los sprints.	164
B.5. Selección de historias asignadas para cada <i>sprint</i>	165
B.6. Carga de estudiantes matriculados en un curso.	166
B.7. Página para la gestión de los objetivos.	167
B.8. Interfaz para crear un nuevo objetivo.	167
B.9. Página para la gestión de las acciones de aprendizaje.	168
B.10. Duplicar una acción.	169
B.11. Página para la gestión de plantillas y preguntas.	170
B.12. Criterios evaluados en una plantilla.	170

B.13.Primer página del procesamiento de datos.	171
B.14.Segunda página del procesamiento de datos.	172
B.15.Selección de curso y acción previas.	173
B.16.Selección de estudiante y <i>dashboard</i>	174

Índice de tablas

1.1. Objetivos.	5
1.2. Restricciones Técnicas del Proyecto.	6
2.1. Tabla de evaluación de riesgos	14
2.2. Presupuesto <i>hardware</i>	22
2.3. Presupuesto <i>software</i>	23
2.4. Presupuesto recursos humanos.	24
2.5. Presupuesto <i>hardware</i>	30
4.1. Actores.	57
4.2. Especificación del caso de uso CU-01: Crear curso.	59
4.3. Especificación del caso de uso CU-02: Procesar estudiantes.	60
4.4. Especificación del caso de uso CU-03: Planificar <i>sprint</i> curso.	61
4.5. Especificación del caso de uso CU-04: Gestionar producto aprendizaje.	62
4.6. Especificación del caso de uso CU-05: Gestionar objetivos.	63
4.7. Especificación del caso de uso CU-06: Gestionar historias.	64
4.8. Especificación del caso de uso CU-07: Gestionar criterios.	65
4.9. Especificación del caso de uso CU-08: Gestionar acciones.	65
4.10. Especificación del caso de uso CU-09: Gestionar acciones de evaluación.	66
4.11. Especificación del caso de uso CU-10: Gestionar acciones de aprendizaje.	67
4.12. Especificación del caso de uso CU-11: Gestionar método evaluación.	67
4.13. Especificación del caso de uso CU-11: Gestionar plantillas.	68
4.14. Especificación del caso de uso CU-13: Gestionar preguntas.	69
4.15. Especificación del caso de uso CU-14: Procesar autoevaluaciones.	70
4.16. Especificación del caso de uso CU-15: Procesar evaluaciones.	70
4.17. Especificación del caso de uso CU-16: Procesar tareas.	71
4.18. Especificación del caso de uso CU-17: Procesar resultados.	71
4.19. Especificación del caso de uso CU-18: Consultar información agregada.	72
4.20. Requisitos funcionales.	73
4.21. Requisitos no funcionales.	73
4.22. Requisitos de interfaz externa.	74
7.1. Especificación de la prueba 1.	124
7.2. Especificación de la prueba 2.	126
7.3. Especificación de la prueba 3.	128
7.4. Especificación de la prueba 4.	130

7.5. Especificación de la prueba 5.	132
7.6. Especificación de la prueba 6.	133
7.7. Especificación de la prueba 7.	134
7.8. Especificación de la prueba 8.	136

Parte I

Descripción del proyecto

Capítulo 1

Introducción

Las metodologías ágiles surgieron a principios del siglo XXI [1], como respuesta a la rigidez de los modelos tradicionales en el contexto de la gestión de proyectos. La idea clave de estas metodologías es la adaptación al cambio de forma rápida y eficiente, basándose en principios de flexibilidad, colaboración y mejora continua. La “agilidad” que permiten estas metodologías destaca, principalmente, en el ámbito del desarrollo software, donde los cambios, actualizaciones y nuevas especificaciones son comunes.

En los últimos años, este tipo de metodologías ha aumentado su impacto, trascendiendo en el mundo empresarial. La gran mayoría de las grandes empresas [4] utilizan estas metodologías para conseguir un mejor rendimiento en sus trabajadores, lo que ha hecho que otros sectores, directamente relacionados con el laboral, se vean afectados. Entre ellos, está el de la educación, donde han dado lugar a nuevas formas de enseñanza orientadas a desarrollar habilidades como el trabajo en equipo, la autonomía y la resolución de problemas. En este contexto, la educación ágil busca preparar a los estudiantes para un entorno laboral dinámico e incierto, acercándolos a las metodologías que encontrarán en su futuro profesional [17].

Desarrollar una asignatura con este tipo de metodologías conlleva la generación de un gran número de artefactos por parte del alumnado y sus docentes, donde reportan datos tales como los objetivos de la asignatura, las acciones formativas a realizar y los criterios de satisfacción de las mismas, los resultados de los alumnos, el tiempo invertido por cada uno de ellos en la realización de sus tareas, o la evaluación por parte de los docentes y los propios alumnos. El principal problema es que, para ello, es necesario invertir mucho tiempo del que, en la mayoría de las ocasiones, los profesores no disponen, por lo que prefieren seguir metodologías más tradicionales. Toda esta cantidad de datos puede ser utilizada para realizar diferentes tipos de análisis y poder sacar conclusiones, de forma que se pueda mejorar el proceso de enseñanza-aprendizaje.

En el contexto presentado, este trabajo se enfoca en la automatización de la recogida de los datos, que facilitará su posterior análisis (como ilustraremos también en este proyecto). Así, este Trabajo Fin de Grado (TFG) sentará las bases para que en el futuro se pueda ayudar a los profesores a liberar carga de trabajo, que pueda ser utilizada para actividades que aporten un mayor valor a los alumnos, como recomendaciones o ayudas personalizadas para superar dificultades que se presenten durante el curso. Además de esto, con este proyecto se busca que los profesores puedan ver de forma más visual el avance de sus alumnos a lo largo del curso, obteniendo un *feedback* que permita visualizar cuáles de las acciones formativas planteadas han ayudado a una mejora del rendimiento de los estudiantes.

La asignatura Sistemas de Bases de Datos, impartida en el Grado de Ingeniería Informática

de Servicios y Aplicaciones, es un ejemplo real de este contexto de innovación educativa. Esta asignatura sigue la metodología *Test-Driven Learning* (TDL) [12], realización del *framework eXtreme Learning*. A lo largo de la asignatura, se generan numerosas hojas de cálculo con información de las autoevaluaciones de los estudiantes en cada una de las pruebas realizadas, así como otras tantas con los registros del trabajo de los alumnos en cada uno de los *sprints* en los que está planteada la asignatura. La intención de este trabajo es poder recoger toda esa información de forma automática para almacenarla en una base de datos que nos permita, posteriormente, generar analíticas que informen y ayuden a la toma de decisiones de profesores y alumnos.

Este trabajo se enmarca dentro del campo de las analíticas del aprendizaje (en inglés *Learning Analytics*). Este trata sobre la medición, recopilación, análisis y reporte de datos sobre los alumnos y sus contextos, con el fin de comprender y mejorar el aprendizaje y los entornos en los que se produce [24]. Además, el campo de *Learning Analytics* en general, y en concreto este TFG, contribuyen a los Objetivos de Desarrollo Sostenible ¹, en concreto al 4 (“Educación de calidad”) y al 16 (“Paz, justicia e instituciones sólidas”), desarrollando procesos de enseñanza-aprendizaje eficaces, responsables y transparentes. Gracias a la base planteada en este Trabajo Fin de Grado, se pueden desarrollar también futuros proyectos en los que se realice un análisis más profundo de estos datos o, incluso, realizar predicciones a futuro de los posibles rendimientos de los alumnos, de forma que los estudiantes puedan saber sus puntos más débiles en los que deben mejorar incluso antes de afrontar las pruebas de evaluación.

La introducción planteada en párrafos anteriores es la idea sobre la que he construido este proyecto, con el fin de conseguir lo mencionado previamente. Antes de comenzar con los aspectos más técnicos, es necesario definir los límites del problema que vamos a tratar y fijarnos unos objetivos a cumplir durante el desarrollo del mismo. Además, también analizaremos posibles inconvenientes que puedan surgir a lo largo del desarrollo, con la intención de estar todo lo preparados que podamos ante cualquier adversidad que surja en el camino. Una vez realizada dicha introducción, en este mismo capítulo se podrá consultar una guía de los distintos apartados de la memoria para que sea más sencillo para el lector poder seguir el desarrollo del proyecto.

1.1. Planteamiento del problema (*Problem Statement*)

Este apartado presenta el planteamiento del problema abordado en este proyecto. Este planteamiento proporciona una descripción general del problema a tratar, convirtiendo la idea planteada en la introducción en un problema acotado, bien definido, que nos permita realizar una investigación más focalizada.

Destacaremos en los siguientes puntos las diferencias entre la situación ideal que pretendemos que se pueda conseguir gracias a este Trabajo Fin de Grado y la realidad que tenemos actualmente, destacando las posibles consecuencias en caso de no abordar dicho problema y concluyendo con el planteamiento propuesto para resolverlo.

- IDEAL: En muchas ocasiones, los alumnos no tienen claro que decisiones tomar para mejorar su aprendizaje, y los profesores no tienen capacidad para realizar un seguimiento personalizado de la evolución de todos los estudiantes. Por ello, sería necesario que tanto profesores como alumnos puedan informar sus decisiones con cierta evidencia, que puede ser extraída de la gran cantidad de datos disponibles sobre el aprendizaje y su contexto,

¹Se pueden consultar todos los Objetivos de Desarrollo Sostenible en: <https://sdgs.un.org/es/goals>

mejorando así sus prácticas de enseñanza-aprendizaje. Este proceso de recogida e integración de información debería ser simple y automático, para facilitar la construcción de herramientas de análisis y visualización que se nutran de la misma.

- **REALIDAD:** Actualmente, los docentes tienen que procesar de forma manual los resultados de los estudiantes. Además, el almacenamiento de los datos históricos de los resultados de los alumnos debe hacerse de forma manual, así como las posibles recomendaciones de mejora para cada uno de los alumnos. Por otro lado, los alumnos, al realizar las entregas, solo obtienen una visión limitada de sus resultados, sin una contextualización basada en evidencia que informe sobre el progreso y los puntos en los que se han producido los errores.
- **CONSECUENCIAS:** En esta situación actual, el profesor no dispone del tiempo necesario para realizar un análisis personalizado por alumno, por la carga de trabajo que conlleva procesar y analizar toda la información generada en cada una de las entregas realizadas por los estudiantes. Por su parte, los estudiantes no obtienen una visión clara de sus resultados, lo que puede generar incertidumbre sobre cómo mejorar su desempeño académico.
- **PROPUESTA:** En este proyecto, implementaremos un sistema que sea capaz de automatizar el procesamiento de las evidencias de aprendizaje de una asignatura y organizar esta información en una base de datos relacional. Además, se creará un *dashboard* que ilustre cómo profesores y alumnos podrían beneficiarse de las analíticas del aprendizaje.

1.2. Objetivos

Nuestro proyecto se centra en conseguir un sistema que sea capaz de, automáticamente, procesar recursos generados durante el proceso de enseñanza-aprendizaje de una asignatura. Podemos desglosar esta idea general en varios objetivos, que hemos planteado en la tabla 1.1.

ID	Objetivos
Obj-01	Diseñar un modelo de dominio que represente el funcionamiento integral de una asignatura universitaria de acuerdo con la metodología de <i>Test-Driven Learning</i> .
Obj-02	Diseñar e implementar un mecanismo que facilite la gestión y automatice el procesamiento de los recursos de enseñanza-aprendizaje que se generan a lo largo de una asignatura que implementa la metodología <i>Test-Driven Learning</i> .
Obj-03	Construir una solución de <i>Learning Analytics</i> que ilustre el potencial de la base de conocimiento obtenida en el Obj-02.

Tabla 1.1: Objetivos.

1.3. Condicionantes

Consideramos ahora algunos puntos que pueden limitar el desarrollo del proyecto, ya sean por falta de disponibilidad de algunos recursos o por factores ajenos al proyecto que puedan

afectar a la planificación o desarrollo del mismo.

1.3.1. Restricciones

Las restricciones técnicas son limitaciones que encontramos a la hora de planificar, desarrollar o implementar el proyecto. Pueden hacer referencia a términos de falta de disponibilidad de *hardware*, *software* o herramientas que sean necesarias para el desarrollo, aunque también pueden ser factores externos que no sean competencia directa del equipo de desarrollo.

ID	Restricciones Técnicas
Rest-01	Los datos utilizados en este proyecto se han anonimizado previamente para asegurar el cumplimiento de las normas vigentes en protección de datos.
Rest-02	Para realizar el estudio indicado en los objetivos se utilizarán datos de la asignatura Sistemas de Bases de Datos, del curso 2024-2025.

Tabla 1.2: Restricciones Técnicas del Proyecto.

1.4. Alcance del proyecto

En este trabajo vamos a trabajar en tres vías principalmente: el procesamiento automático de los datos, la gestión de esos datos desde una aplicación, y la construcción de un *dashboard*. Estas tres características principales se muestran en el diagrama de la Figura 1.1

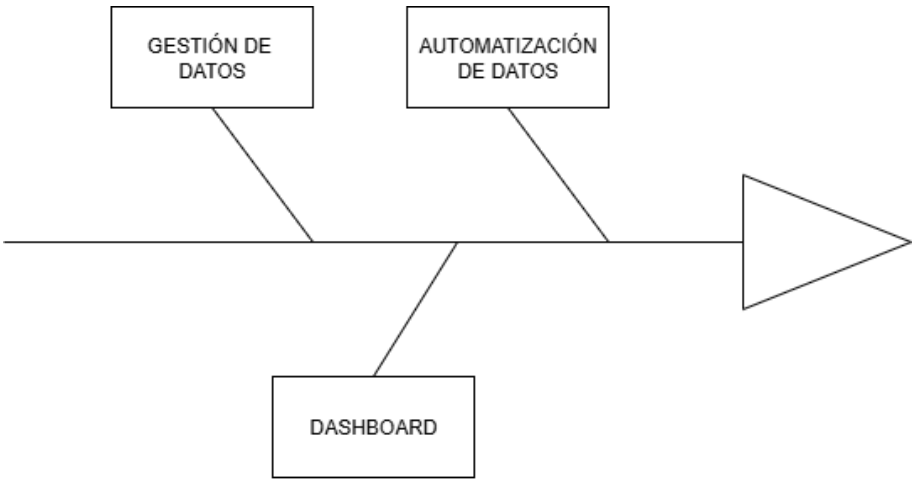


Figura 1.1: Diagrama de características de la aplicación.

De estas características, la principal es el procesamiento de datos. Por ello, mostraremos a continuación el flujo de datos (*workflow*) sobre el que se va a construir esta parte. En nuestro caso, este *workflow* contiene seis etapas, como podemos ver en la Figura 1.2.

Explicaremos brevemente cada una de las etapas de este *workflow* a continuación.

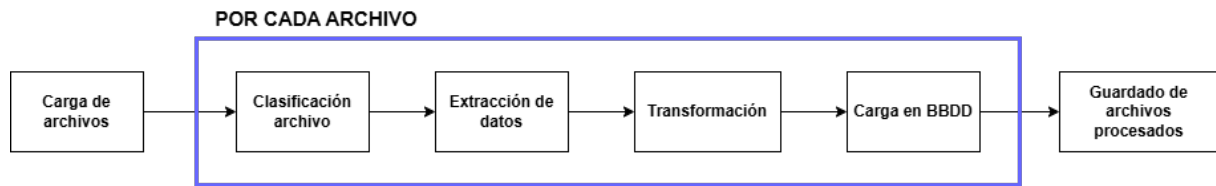


Figura 1.2: Workflow de la parte de Procesamiento de datos.

1. **Carga de archivos:** Las diferentes fuentes de datos que se van a utilizar para obtener y procesar los datos deben de estar almacenadas en diferentes carpetas de forma organizada.
2. **Clasificación del archivo:** Se debe indicar el tipo de archivo que es el cargado, información que podremos obtener del nombre o de la carpeta desde la que se ha cargado. Esto permitirá distinguir las fuentes de datos a la hora de procesar los archivos.
3. **Extracción de datos:** Para cada uno de los archivos cargados, una vez identificada la fuente de datos de la que proviene se debe leer la información que contiene dicho archivo. Esta información se cargará en memoria para poder ser posteriormente utilizada.
4. **Transformación:** Es posible que los datos que hayamos obtenido de nuestras fuentes no estén en el formato en el que queremos almacenarlos posteriormente. Por ello, necesitaremos transformar parte de esta información para que tenga la estructura deseada antes de cargarla.
5. **Carga en BBDD:** Una vez la información tiene la estructura que deseamos que aparezca en la base de datos (BBDD), podremos realizar la carga para poblar las tablas de nuestra base.
6. **Guardado de archivos procesados:** Una vez se hayan procesado todos los archivos disponibles, los almacenaremos en otras carpetas diferentes estos archivos para poder distinguirlos de aquellos que estén pendientes de procesar. De esta forma podremos, además, mantener un histórico de los archivos procesados por si fuera necesario.

La implementación de este trabajo se realizará utilizando el lenguaje de programación *Python*, combinándolo con librerías como *pandas* o *pyspark*.

1.5. Estructura de la memoria

Se ha estructurado la memoria de este proyecto en los siguientes capítulos:

- **Capítulo 1: Introducción.** Ayuda a entender el contexto en el que se desarrolla el trabajo, motivando como es necesario simplificar el trabajo de los profesores y permitir una mayor personalización del aprendizaje.
- **Capítulo 2: Planificación.** Explica la metodología seguida para desarrollar este proyecto, incluyendo la planificación, presupuestos y los respectivos balances temporal y económico.

- **Capítulo 3: Antecedentes.** Aborda todos los conocimientos básicos necesarios para la comprensión del proyecto. Para ello, además de desarrollar todos los conceptos clave, se plantea una revisión del estado del arte relacionado con el trabajo realizado, revisando así completamente el contexto de enseñanza-aprendizaje en el que se ha desarrollado este TFG.
- **Capítulo 4: Análisis.** Describe el análisis del producto software que se va a construir en este proyecto. Para ello, se ha realizado un análisis de todos los requisitos importantes para el desarrollo del mismo, así como el planteamiento del diseño conceptual que permite poner en la tierra todas las ideas planteadas en capítulos anteriores.
- **Capítulo 5: Diseño.** Presenta el diseño del producto, para lo que se aborda la estructura lógica del almacenamiento necesario, así como las arquitecturas y los bocetos de la interfaz de usuario de la aplicación.
- **Capítulo 6: Implementación.** Explica como se ha realizado la implementación de la aplicación, herramientas y tecnologías utilizadas y como estas se han aplicado en la aplicación desarrollada.
- **Capítulo 7: Pruebas.** Describe todas las pruebas realizadas para evaluar el correcto funcionamiento de la aplicación. Además, también se detallarán las comprobaciones realizadas a los datos cargados, asegurándonos que toda la información sea consistente y tenga sentido.
- **Capítulo 8: Aceptación.** Explica el proceso que se ha realizado para asegurarse de que el proyecto cumple con los objetivos establecidos. Para ello se detallan una serie de métricas que permitirán evaluar el éxito del proyecto.
- **Capítulo 9: Conclusiones.** Expone las conclusiones obtenidas una vez finalizado el desarrollo del proyecto, describiendo además futuras líneas de trabajo abiertas para sacar un mayor partido al proyecto presentado.

Capítulo 2

Gestión del proyecto

Toda la gestión que se ha realizado a lo largo del proyecto, junto con la metodología ASAP (*Agile Student Academic Projects*) [10], que hemos elegido para el desarrollo de este proyecto, se presenta en este capítulo. Gracias a esta metodología hemos gestionado este proyecto desde un punto de vista “ágil”, planificando las distintas tareas, agrupadas en actividades, a lo largo de los cuatro *sprints* en los que hemos dividido este Trabajo Fin de Grado (TFG).

Comenzaremos exponiendo en el apartado 2.1 todos los detalles de la metodología utilizada, de forma que podamos comprender mejor los siguientes apartados y capítulos de este trabajo. En este mismo capítulo, tenemos la planificación temporal (2.3), los presupuestos (2.4), el balance temporal y económico (2.5) y los riesgos (2.2).

2.1. Metodología

Como se ha mencionado anteriormente, hemos elegido una metodología ágil para este proyecto, en concreto ASAP. La elección de esta metodología está marcada por su diseño, ya que está planteada para ayudar al alumno a realizar el Trabajo Fin de Grado (TFG) de forma efectiva y profesional [10].

ASAP nos permite familiarizarnos con la forma de trabajo de *frameworks* comúnmente utilizados en el ámbito profesional, como pueden ser *Scrum* [19] o *eXtreme Programming* [7], planteando una metodología basada en estos *frameworks* que se adapte perfectamente al ámbito pedagógico y, concretamente, al desarrollo de un TFG. Esta metodología nos permite alcanzar los objetivos de nuestro proyecto, proporcionando una estructura de trabajo clara y aportando un *feedback* constante. Para ello, el entorno que presenta ASAP presta una ayuda esencial para la planificación, desarrollo, aceptación y comunicación del proyecto.

La metodología ASAP se organiza principalmente en tres componentes clave: roles, eventos y artefactos, inspirados en los conceptos del mismo nombre en *Scrum*. En nuestro caso, antes de explicar dichos componentes, presentaremos los cinco paquetes de trabajo que plantea ASAP. Con esta base, podremos entender perfectamente el contexto en el que se desarrollan los roles, eventos y artefactos que define la metodología. Para completar esta sección, presentaremos las herramientas utilizadas, que han facilitado el seguimiento del proyecto y la comunicación entre los tutores y el alumno.

2.1.1. Paquetes de trabajo

ASAP describe un TFG en torno a cinco paquetes de trabajo que permiten desarrollar completamente los objetivos de aprendizaje de este. Cada paquete de trabajo comprende la realización de un conjunto de actividades, cuyos resultados se evalúan de acuerdo con un conjunto de criterios de aceptación, establecidos en ASAP. En la planificación temporal desarrollada en el apartado 2.3, veremos cómo hemos realizado nuestra planificación en torno a estas actividades.

Cada uno de los paquetes de trabajo presentados a continuación permite alcanzar los hitos principales en los que se divide el TFG. En este caso, tenemos cinco paquetes de trabajo.

- **Proyecto:** La gestión de un proyecto de este estilo conlleva, como primer punto, una identificación clara del problema que se va a tratar en el mismo. Para ello, en este paquete de trabajo, se quiere forjar la base sobre la que posteriormente se elaborará el proyecto, tratando aspectos como la planificación, metodología o recursos utilizados. Toda esta construcción se realizará en torno al planteamiento iterativo de la metodología ASAP, revisando, en cada uno de los *sprints*, el avance de estas historias para conseguir un producto de alta calidad.
- **Antecedentes:** Una vez tenemos planteada la idea general del problema que queremos tratar en el proyecto, podemos realizar un análisis de la situación en la que se enmarca el TFG. Para ello, compararemos este trabajo con otras soluciones similares, de forma que podamos aprender de los éxitos logrados en proyectos previamente desarrollados, pero también entender los errores cometidos en dichos trabajos. Estudiaremos en este apartado el contexto de negocio en el que se desarrolla el proyecto, analizando, entre otras cosas, el estado del arte o los fundamentos técnicos y teóricos.
- **Desarrollo:** Este proyecto irá creciendo a lo largo de los *sprints* de forma iterativa e incremental. Del control de esta evolución se encarga este paquete de trabajo, en el que partiremos desde un análisis y primer diseño de la solución, con el objetivo de llegar a la construcción, evaluación y prueba de nuestro producto final.
- **Aceptación:** Los objetivos planteados en el apartado 1.2 de este proyecto deben cumplirse para asegurar el éxito del trabajo realizado, y, para ello, completaremos un conjunto de actividades destinadas a medir la calidad del producto final. Este será el principal propósito de este paquete de trabajo, estableciendo métricas que se deberán cumplir para evaluar el éxito del proyecto.
- **Comunicación:** Un proyecto desarrollado como parte de un TFG comprende dos formas de comunicar el trabajo realizado. La primera de ellas es a través de la memoria técnica del proyecto, que se irá completando durante todos los *sprints* de forma incremental, hasta conseguir un producto que represente el trabajo realizado al completo. La otra parte es la presentación del mismo, donde se expondrán los puntos más importantes de la extensa explicación dada en la memoria.

Como hemos mencionado anteriormente, cada uno de estos paquetes de trabajo se divide en actividades, que han sido las utilizadas para realizar la planificación del trabajo a lo largo de los *sprints*. Podremos consultar las actividades consideradas en cada paquete de trabajo, así como el reparto de estas en los cuatro *sprints* programados, en el apartado de planificación (2.3).

2.1.2. Roles

La metodología ASAP define 4 roles, con los que se representan a los implicados directamente en el desarrollo del TFG. Cada uno de ellos lo hace de una forma particular, con responsabilidades y contribuciones específicas, detalladas a continuación.

- **Estudiante:** El rol principal en un TFG es el estudiante, por lo que debe ser el más activo en el proyecto. Este es el encargado de planificar las tareas y desarrollar la implementación del proyecto, con el objetivo de alcanzar la calidad de producto deseada. Es importante que esté continuamente en contacto con los otros roles, especialmente con el tutor y la comunidad, con la finalidad de que todos trabajen en la misma dirección.
- **Tutor:** Para este TFG, el rol de tutor lo tendrán dos personas, ya que ASAP contempla la posibilidad de que sean uno o varios tutores los que colaboren en el proyecto. Su presencia en el proyecto es clave, aportando *feedback* de forma constante, y prestando su ayuda tantas veces como se necesite. El principal objetivo de los tutores es supervisar el desarrollo del trabajo, siendo responsabilidad suya la calidad del producto desarrollado, por lo que se deben encargar de ayudar al estudiante a conseguir dicha calidad. Para ello, deben proporcionar el material necesario para el desarrollo del proyecto y asegurarse que se cumplen los plazos indicados en la planificación.
- **Comunidad:** Paralelamente al desarrollo de este TFG, otros alumnos siguen esta misma metodología para sus trabajos. Estos estudiantes, además de sus tutores y algunos expertos que colaboran en el desarrollo de los proyectos, serán los que tomen el rol de comunidad. Estos se encargarán de dar un punto de vista diferente del trabajo, aportando soluciones o ideas que puedan mejorar la calidad del mismo.
- **Tribunal:** Este es el único de los roles ajeno al desarrollo del proyecto, por lo que solo conocerá el producto final presentado. El tribunal es responsable de la evaluación del TFG, valorando si los resultados satisfacen los objetivos de aprendizaje planteados.

2.1.3. Eventos

El evento principal que determina la metodología ASAP es el *sprint*. Este TFG, tal como indica la metodología elegida, se abordará en *sprints* de aproximadamente 1 mes de duración. Cada uno de estos *sprints* se puede definir como un período de trabajo en el que se planifican las tareas necesarias para alcanzar el “objetivo del *sprint*”. Este objetivo vendrá determinado por tres aspectos: los resultados consolidados hasta el *sprint* anterior (en caso de que haya alguno), el trabajo a realizar y las condiciones particulares en las que se desarrolla dicho *sprint*.

Durante el mes que dura cada uno de los *sprints*, se llevan a cabo una serie de eventos, definidos por la metodología elegida. Cada uno de estos eventos tiene un objetivo concreto y se desarrolla en un momento diferente del *sprint*, como explicaremos a continuación.

- **Planificación:** Esta tiene lugar al inicio de cada uno de los *sprints*, estableciendo los objetivos que se deben cumplir en ese periodo de tiempo. Para ello, se estudiarán brevemente las actividades propuestas en cada paquete de trabajo, detallando un plan de trabajo para cada una de ellas. Esto permitirá controlar a lo largo de todo el *sprint* que se están cumpliendo las fechas marcadas, para asegurar la entrega de un incremento completo al final del mismo.

- **Sincronización:** Una vez a la semana se realizará una reunión de sincronización en la que, al menos, participarán el estudiante y el tutor, con la posibilidad de que esté presente algún otro miembro de la comunidad, si fuera necesario. En esta reunión el estudiante comentará los avances realizados durante la pasada semana, obteniendo *feedback* por parte del tutor. También se solucionarán posibles bloqueos encontrados desde la última sincronización.
- **Comunicación de progresos:** Al finalizar cada uno de los *sprints*, tendrá lugar una reunión en la que deberá estar presente toda la comunidad. En esta reunión, parte de los estudiantes que estén desarrollando su TFG con la metodología ASAP presentarán los avances realizados durante el *sprint*, con el objetivo de obtener un *feedback* con distintos puntos de vista, y así mejorar de cara a la futura presentación del trabajo.
- **Retrospectiva:** Se realizará al finalizar el *sprint*, con la finalidad de que cada persona reflexione sobre los puntos fuertes y los aspectos a mejorar de su trabajo durante el mes anterior. Todos los miembros de la comunidad deben participar en este evento, consensuando así líneas de mejora para todos los estudiantes que estén desarrollando su TFG.

2.1.4. Artefactos

La metodología ASAP utiliza tres artefactos, que nos serán de utilidad para realizar el seguimiento del proyecto, permitiendo documentar y evaluar el trabajo realizado.

- **Incremento:** Como hemos mencionado anteriormente, al finalizar cada *sprint* los estudiantes deben presentar el trabajo realizado durante ese mes. Esta entrega es lo que se conoce como incremento, mostrando el progreso realizado en el proyecto y el grado de satisfacción de los distintos objetivos abordados. Contendrá el producto del TFG en su estado actual, incluyendo la memoria, el cuaderno de trabajo y cualquier otro contenido que se requiera en el proyecto.
- **Retroalimentación:** Este artefacto va de la mano de la comunicación de progresos, mencionada en el apartado anterior (2.1.3). En la retroalimentación se incluirán todos los comentarios y sugerencias recibidos al finalizar el *sprint* por parte de los distintos miembros de la comunidad. Con esta información, se deben plantear una serie de cambios tanto en la planificación como en el desarrollo del siguiente *sprint*, para evitar los errores cometidos.
- **Cuaderno de trabajo:** Es el elemento principal que se utiliza para la planificación y seguimiento de las actividades que se deben abordar en cada *sprint*. El cuaderno de trabajo se entrega como una hoja de cálculo, con cuatro pestañas, que se explican a continuación:
 - **Planificación:** Esta pestaña organiza la información necesaria para llevar a cabo la planificación del proyecto, pero también para realizar una pequeña autoevaluación al terminar el *sprint*. En esta pestaña, al inicio de cada *sprint*, se indican las fechas de inicio y fin planificadas para cada una de las historias incluidas en el alcance de dicho *sprint*. Además, se tiene una columna para ir monitorizando a lo largo del *sprint* el estado de las historias, pudiendo controlar si están sin iniciar, en progreso, bloqueadas o finalizadas. Por último, esta hoja también registrará las fechas de inicio y finalización reales para cada historia, lo que se hará de forma automática con los datos de la pestaña “Trabajo”, lo que nos permitirá realizar una comparación entre la planificación y la realidad al finalizar el *sprint*.

- **Cronograma:** La segunda pestaña de la hoja de cálculo contiene la información temporal del *sprint*, organizada por historias (en las filas), y por días y semanas en las columnas. Esta hoja tiene también una columna “Total Horas” que agrega la cantidad de esfuerzo invertido por historia y objetivo. Toda esta información se completa automáticamente con los datos de las pestañas “Planificación” y “Trabajo”, mostrando, además, con color verde o rojo los días en los que se ha trabajado dentro o fuera de la planificación respectivamente.
- **Trabajo:** Esta pestaña actúa a modo de bitácora del TFG, donde el alumno mantiene un registro diario de las tareas realizadas para completar el trabajo. De cada tarea se registra su fecha y la historia con la se vincula, indicando el tiempo (en minutos) invertido en ella, y, opcionalmente, añadiendo un comentario por cada tarea realizada. La información de esta pestaña es clave para mantener actualizado el cronograma presente en la pestaña anterior. Esto nos permite, además, guardar un registro sobre el tiempo dedicado a cada tarea, de forma que, en caso de necesitarlo, podamos consultar a que tareas le hemos dedicado más tiempo en el *sprint*.
- **Dashboard:** La última pestaña contiene un pequeño cuadro de mandos con visualizaciones que permiten al alumno ser más consciente del estado de su TFG. Se presentan algunas visualizaciones que permiten ver más claramente la planificación, el avance de las historias o los resultados de cada una de ellas. Además de eso, se podrá ver un gráfico adicional con la carga de trabajo semanal a lo largo del *sprint*, ya que la dinámica de ASAP busca que el ritmo de trabajo en el TFG se mantenga durante todo el proyecto, por lo que lo ideal sería mantener una altura similar en todas las barras de este gráfico.

2.1.5. Entorno de trabajo

Para poder trabajar de forma cómoda con ASAP, el tutor del TFG ha facilitado una herramienta, que forma parte del entorno de la metodología, y que ha sido de gran utilidad para simplificar la comunicación y planificación del proyecto. Esta herramienta es el espacio de trabajo compartido, que se basa principalmente en un canal en *Teams*, en el que estarán todos los miembros que tengan una participación activa en el proyecto. Dentro del canal, cada estudiante tendrá su espacio de trabajo personal, en el que se almacenará toda la información necesaria para el desarrollo del proyecto. El espacio compartido permitirá, además, manejar y tener un registro activo de la comunicación entre todos los participantes en el proyecto. Todo ello ayudará, tanto al alumno como a los tutores, a mantener un *feedback* de forma activa durante el desarrollo del proyecto, con la finalidad de llevar a cabo un proceso de mejora continua que maximice la calidad del producto final.

Además de estos elementos, también se ha utilizado un tablero de *Trello* personal, en el que se han ido organizando las historias a realizar. Estas tareas se han repartido en varias columnas: *Backlog*, *To Do*, *Doing*, *Blocked* y *Done*. Al inicio de cada *sprint*, una vez realizada la planificación, se han ido creando las historias en el tablero con las fechas de inicio y finalización programadas, de forma que ha sido más sencillo seguir el método de trabajo y organizar las tareas a realizar.

2.2. Riesgos

Para continuar con este apartado, se han analizado los posibles problemas que podrían surgir durante el proyecto, con el fin de estar preparados para cualquier inconveniente y poder reaccionar ante estos con soluciones rápidas y eficientes.

Riesgo	Posibilidad de Ocurrencia	Mitigación
Disponibilidad de pocos datos para obtener conclusiones definitivas.	Baja	Generalizar conclusiones o realizar pruebas con ejemplos de datos.
Sobrecarga de datos que dificulte el tratamiento de los mismos.	Alta	Crear subgrupos de estudiantes, pudiendo consultar para ellos toda la información que hayan generado estos durante el curso.

Tabla 2.1: Tabla de evaluación de riesgos

2.3. Planificación temporal

Como ya hemos mencionado en la sección anterior, la metodología ASAP divide el proyecto en 4 *sprints* inicialmente, con la posibilidad de un quinto *sprint* de “desbordamiento”, que se utilizará en caso de no haber completado el alcance del proyecto en el cuarto *sprint*. Al ser el TFG un trabajo de 12 créditos ECTS, está estimado en unas 300-360 horas de trabajo; por tanto, para cada *sprint* se deberían invertir alrededor de 75 horas de trabajo.

La planificación temporal de los *sprints* es la siguiente:

- *Sprint* #1: 24/02/2025 - 21/03/2025
- *Sprint* #2: 24/03/2025 - 25/04/2025
- *Sprint* #3: 28/04/2025 - 23/05/2025
- *Sprint* #4: 26/05/2025 - 20/06/2025

Al inicio de cada uno de los *sprints*, realizaremos una planificación en la que se establecen las fechas esperadas de inicio y finalización de cada una de las actividades a realizar en ese *sprint*. El tutor propone las actividades a realizar en el *sprint*, de acuerdo con los paquetes de trabajo activos en él, y se mostrarán en los sucesivos apartados. Para realizar cada una de las actividades, el alumno se encarga de subdividirlas en tareas, que se organizarán en el tablero de *Trello* previamente mencionado, y que se referenciarán en el cuaderno de trabajo cuando se trabaje en ellas. Con esta división en tareas, a las que se les asignan unas fechas de inicio y fin en función de la planificación realizada, el alumno puede identificar fácilmente en qué tareas tiene que trabajar para cumplir las fechas planificadas.

Cabe destacar que, para realizar la planificación, se utiliza el cuaderno de trabajo previamente explicado en el apartado 2.1.5. Esta hoja de cálculo permitirá mantener el registro de las tareas

en las que se ha trabajado cada día, referenciando cada una de ellas con una de las historias definidas por el tutor. Gracias a esta herramienta, todos los participantes del proyecto tienen la información del estado del avance del proyecto siempre actualizada y pueden consultar, entre otras cosas, las historias en las que se está trabajando o el tiempo dedicado a las historias ya completadas.

Mostraremos a continuación las historias realizadas en cada *sprint*, así como un resumen de las horas invertidas en cada una de ellas y un diagrama de *Gantt* por *sprint*, que nos ayudará a visualizar las tareas en las que se ha trabajado cada día.

2.3.1. *Sprint* #1

En este *sprint* se han desarrollado las siguientes actividades, organizadas en función de los cinco paquetes de trabajo principales en los que divide las actividades ASAP.

1. Proyecto

Para este primer paquete de trabajo, tenemos las siguientes actividades que, en este caso, conforman todas las incluidas en el paquete de trabajo, todas ellas abordadas en este primer *sprint*.

- **Proyecto-1 (Planteamiento del problema):** Construir el *problem statement* del proyecto para asegurar la identificación y comunicación precisa del problema a resolver.
- **Proyecto-2 (Objetivos):** Establecer los objetivos del proyecto para guiar el trabajo y evaluar el éxito de sus resultados de manera efectiva.
- **Proyecto-3 (Condicionantes):** Establecer las condiciones que pueden afectar al desarrollo del proyecto en términos de negocio y técnicos.
- **Proyecto-4 (Metodología):** Establecer las condiciones que pueden afectar al desarrollo del proyecto en términos de negocio y técnicos.
- **Proyecto-5 (Planificación):** Desarrollar un plan detallado de tareas y actividades que optimice el uso eficiente de los recursos disponibles en el proyecto y cumplir con los plazos establecidos.
- **Proyecto-6 (Balance):** Hacer el balance entre los recursos utilizados y los resultados obtenidos, analizando las desviaciones que se han producido respecto a la planificación original, para asegurar la viabilidad, la eficiencia y el aprendizaje continuo.
- **Proyecto-7 (Interacción):** Mantener una interacción efectiva y continua durante todo el proyecto para asegurar la transparencia, la colaboración y la mejora continua.

2. Antecedentes

Para este segundo paquete de trabajo, se abordarán todas las actividades en este primer *sprint*. Estas actividades son las siguientes:

- **Antecedentes-1 (Entorno de negocio):** Conocer el entorno de negocio en el que se desarrolla el proyecto para construir un producto que se alinee con las necesidades y expectativas de sus *stakeholders*.

- **Antecedentes-2 (Estado del arte):** Completar una revisión exhaustiva del estado del arte relacionado con el proyecto para identificar soluciones y enfoques existentes y, en base a ello, determinar las oportunidades de mejora y diferenciación del producto a desarrollar en el TFG.
- **Antecedentes-3 (Fundamentos teóricos):** Comprender los fundamentos teóricos del proyecto para poder elegir las teorías o principios más adecuados para abordar el diseño de la solución.
- **Antecedentes-4 (Fundamentos técnicos):** Comprender los fundamentos técnicos del proyecto para poder elegir objetivamente las herramientas necesarias para construir el producto.

3. Desarrollo

Este tercer paquete de trabajo se compone de cinco actividades, de las cuales solo una de ellas se tratará en este *sprint*. Cabe destacar que, además, solo se abordará de forma parcial.

- **Desarrollo-1 (Entorno de negocio):** Analizar el problema para determinar los diferentes tipos de requisitos (y los *stakeholders* relacionados) que describen el producto a construir.

4. Aceptación

En este *sprint* no tenemos ninguna historia de aprendizaje asociada a este objetivo en la que trabajar.

5. Comunicación

Comprende dos grandes actividades, una de ellas relacionada con la documentación técnica que comenzaremos a desarrollar en este *sprint*, y otra, tratada más adelante, que estará enfocada en la presentación del trabajo.

- **Comunicación-1 (Documentación técnica):** Esta actividad comprende varios grupos de criterios, enfocados entre otras cosas en la estructura, organización, redacción o la introducción de referencias y uso de un formato correcto en la memoria del proyecto.

En la imagen 2.1 encontramos el diagrama de Gantt con la planificación del primer *sprint*. Cabe destacar que, aunque el *sprint* comenzó el día 24 de febrero, la planificación no se realiza hasta el 26 por dificultades de disponibilidad por parte de los tutores en los primeros días de *sprint*. También podemos observar que la entrega del incremento del *sprint* se realizará el día 23 de marzo, siendo la comunicación de procesos el día 25, lo que explica la finalización de gran parte de las historias el día 23, a excepción de la interacción.

2.3.2. *Sprint* #2

En el segundo *sprint* se ha trabajado, en las mismas actividades que en el primer *sprint*, y, además, en otras 3 actividades del paquete de trabajo de “Desarrollo” y una actividad más en el paquete de trabajo de “Comunicación”. Se exponen a continuación todas las actividades en las que se ha trabajado.

	Semana #1							Semana #2							Semana #3							Semana #4						
Proyecto	2402	2502	2602	2702	2802	0103	0203	0303	0403	0503	0603	0703	0803	0903	1003	1103	1203	1303	1403	1503	1603	1703	1803	1903	2003	2103	2203	2303
Planteamiento del problema																												
Objetivos																												
Condicionantes																												
Metodología																												
Planificación																												
Balance																												
Interacción																												
Antecedentes	2402	2502	2602	2702	2802	0103	0203	0303	0403	0503	0603	0703	0803	0903	1003	1103	1203	1303	1403	1503	1603	1703	1803	1903	2003	2103	2203	2303
Entorno de negocio																												
Estado del arte																												
Fundamentos teóricos																												
Fundamentos técnicos																												
Desarrollo	2402	2502	2602	2702	2802	0103	0203	0303	0403	0503	0603	0703	0803	0903	1003	1103	1203	1303	1403	1503	1603	1703	1803	1903	2003	2103	2203	2303
Análisis del problema																												
Comunicación	2402	2502	2602	2702	2802	0103	0203	0303	0403	0503	0603	0703	0803	0903	1003	1103	1203	1303	1403	1503	1603	1703	1803	1903	2003	2103	2203	2303
Documentación técnica																												

Figura 2.1: Planificación del sprint #1.

1. Proyecto

Las actividades abordadas en este segundo *sprint* son las mismas que se trataron en el primero.

2. Antecedentes

Al igual que con el primer paquete de trabajo, en este caso trabajaremos de nuevo sobre todas las actividades con el fin de refinarlas y poder completarlas.

3. Desarrollo

En el tercero de los paquetes de trabajo, tenemos más diferencias con respecto al primer *sprint*. En este caso trabajaremos sobre 4 de las 5 actividades que componen este objetivo, aunque dos de ellas se abordarán de forma parcial. En concreto, las actividades “Desarrollo-3” y “Desarrollo-4” se trabajarán al completo, mientras que las dos primeras se hará un trabajo parcial. Las nuevas actividades no presentadas antes sobre las que se trabajará son:

- **Desarrollo-2 (Diseño de la solución):** Establecer el diseño de la solución para poder construir un producto que alcance los objetivos del proyecto.
- **Desarrollo-3 (Construcción del producto):** Construir un producto acorde al diseño planteado para alcanzar los objetivos del proyecto.
- **Desarrollo-4 (Evaluación del producto):** Probar el producto para asegurar que cumple con todos sus requisitos.

4. Aceptación

No planificaremos este paquete de trabajo en el segundo *sprint*

5. Comunicación

En este *sprint*, se trabajará en ambas actividades, de cara a preparar la presentación para la comunicación de procesos al final del *sprint*. La nueva actividad es:

- **Comunicación-2 (Presentación):** Esta actividad comprende varios grupos de criterios, enfocados en el contenido, asegurándose que este ofrece una visión completa,

precisa y consistente del trabajo realizado; y la exposición, que buscan asegurar la calidad de la exposición realizada.

En la imagen 2.2 encontramos el diagrama de Gantt con la planificación del segundo *sprint*.

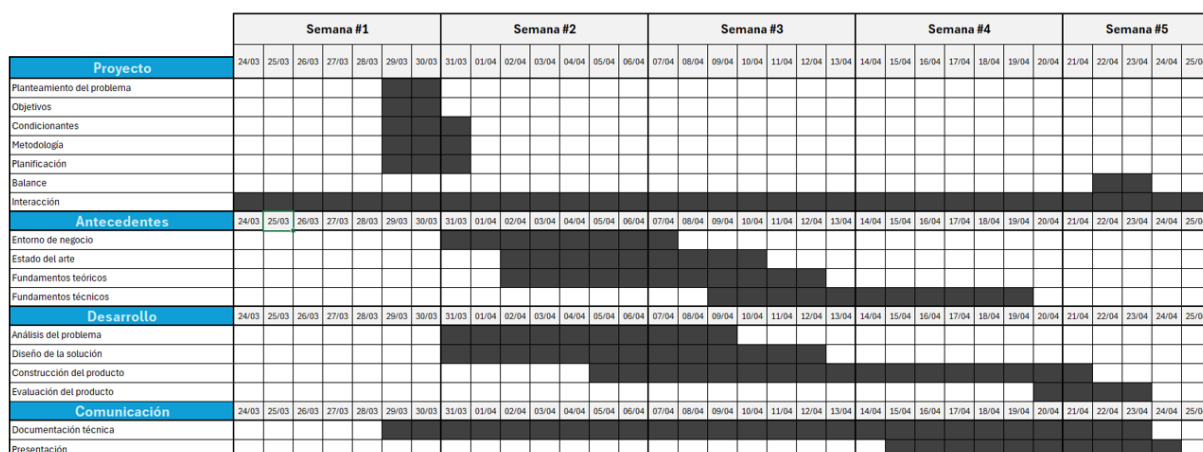


Figura 2.2: Planificación del sprint #2.

2.3.3. Sprint #3

El tercer *sprint* ha consistido en el refinamiento de algunas de las actividades planteadas en el sprint anterior. Por ello, las actividades abordadas en este *sprint* ya fueron planteadas en el anterior en su totalidad.

1. Proyecto

Las historias de aprendizaje en las que hemos vuelto a trabajar son las que deben estar presentes en todos los *sprints*:

- Proyecto-5 (Planificación).
- Proyecto-6 (Balance).
- Proyecto-7 (Interacción).

2. Antecedentes

En este caso solo trataremos una de las actividades, en la que tendremos que trabajar de nuevo porque no estaba finalizada al terminar el *sprint* anterior.

- Antecedentes-4 (Fundamentos técnicos).

3. Desarrollo

Trabajaremos de forma completa este paquete de trabajo, abordando todas las actividades y todos los criterios de las mismas. Trabajaremos sobre

- Desarrollo-1 (Entorno de negocio).

- Desarrollo-2 (Diseño de la solución).
- Desarrollo-3 (Construcción del producto).
- Desarrollo-4 (Evaluación del producto).

4. Aceptación

No planificaremos este paquete de trabajo en el tercer *sprint*.

5. Comunicación

En este *sprint*, volveremos a trabajar únicamente en la documentación técnica del proyecto, posponiendo la presentación del mismo para el último *sprint*.

En la imagen 2.3 encontramos el diagrama de Gantt con la planificación del tercer *sprint*.

	Semana #1							Semana #2							Semana #3							Semana #4				
	28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05
Proyecto																										
Planificación																										
Balance																										
Interacción																										
Antecedentes																										
Fundamentos técnicos																										
Desarrollo																										
Análisis del problema																										
Diseño de la solución																										
Construcción del producto																										
Evaluación del producto																										
Comunicación																										
Documentación técnica																										

Figura 2.3: Planificación del *sprint* #3.

2.3.4. *Sprint* #4

Este cuarto *sprint* ha sido en el que hemos afianzado las actividades abordadas en el anterior, y además hemos completado la parte de aceptación del proyecto, acercándonos así a la finalización del proyecto.

1. Proyecto

Volveremos a trabajar sobre las actividades que deben estar presentes en cada uno de los *sprints*:

- Proyecto-5 (Planificación).
- Proyecto-6 (Balance).
- Proyecto-7 (Interacción).

2. Antecedentes

Refinaremos y completaremos la actividad pendiente de este apartado, en este caso los fundamentos técnicos (Antecedentes-4).

3. Desarrollo

Planificaremos todas las actividades de este paquete de trabajo en este *sprint*. Son las mismas que en *sprints* anteriores, con la incorporación además de:

- **Desarrollo-5 (Puesta en producción):** Caracterizar el proceso de puesta en producción para que los *stakeholders* puedan utilizar el producto construido de acuerdo con sus necesidades.

4. Aceptación

Planificaremos por primera vez este paquete de trabajo en este *sprint*, abordando todas sus actividades.

- **Aceptación-1 (Métricas de éxito):** Definir las métricas de éxito para evaluar el cumplimiento de los objetivos del proyecto.
- **Aceptación-2 (Plan de aceptación):** Diseñar el plan de aceptación para asegurar una evaluación estructurada y rigurosa del producto.
- **Aceptación-3 (Análisis de resultados):** Seleccionar y aplicar la metodología adecuada para asegurar un enfoque estructurado y sistemático en el desarrollo del proyecto.

5. Comunicación

Trabajaremos, de nuevo, las dos actividades de este objetivo, con la idea de preparar la presentación para la defensa final.

En la imagen 2.4 encontramos el diagrama de Gantt con la planificación del cuarto *sprint*.

	Semana #1							Semana #2							Semana #3							Semana #4						
Proyecto	26/05	27/05	28/05	29/05	30/05	31/05	01/06	02/06	03/06	04/06	05/06	06/06	07/06	08/06	09/06	10/06	11/06	12/06	13/06	14/06	15/06	16/06	17/06	18/06	19/06	20/06	21/06	22/06
Planificación																												
Balance																												
Interacción																												
Antecedentes																												
Fundamentos técnicos																												
Desarrollo																												
Análisis del problema																												
Diseño de la solución																												
Construcción del producto																												
Evaluación del producto																												
Puesta en producción																												
Aceptación																												
Métricas de éxito																												
Plan de aceptación																												
Análisis de resultados																												
Comunicación																												
Documentación técnica																												
Presentación																												

Figura 2.4: Planificación del *sprint* #4.

2.4. Presupuestos

Presentaremos en esta sección el coste estimado requerido para llevar a cabo este proyecto. Para ello, hemos considerado por separado el coste que tendría el *hardware*, *software* y los recursos humanos que serían necesarios para un proyecto de este tipo. Por ello, presentaremos a continuación el coste estimado de cada uno de estos elementos, así como un resumen final en el que se muestra el presupuesto estimado de este proyecto.

2.4.1. Hardware

Para el desarrollo al completo de este proyecto, se ha utilizado un ordenador portátil de gama media-alta, en concreto, con las siguientes especificaciones:

- Procesador: 13th Gen Intel(R) Core(TM) i7-13620H - 2.40 GHz
- Memoria RAM: 16 GB
- Almacenamiento interno: 1 TB
- Sistema operativo: Windows 11 Home, 64 bits

Además, hemos requerido de una conexión Wi-Fi estable para poder realizar la investigación pertinente, así como hacer la conexión con la base de datos, desplegada en un servidor de la universidad y realizar todas las peticiones necesarias a la misma.

Para este TFG se han utilizado dos máquinas virtuales de la Universidad de Valladolid, una de ellas en la que está desplegada la base de datos, y otra que contendrá el despliegue de la

aplicación. En nuestro caso, ambas han sido proporcionadas por la Universidad, pero en caso de no serlo deberíamos considerarlas para el presupuesto estimado de nuestro proyecto.

En resumen, podemos construir la siguiente tabla con los datos necesarios para el posterior cálculo del presupuesto:

Herramienta	Coste total (euros)	Vida útil (meses)	Porcentaje uso (%)	Uso en meses
Ordenador portátil	799	$4 \times 12 = 48$	75	4
Wi-Fi	15 (al mes)	-	20	4
Máquina virtual BBDD	20 (al mes)	-	50	3
Máquina virtual despliegue	30 (al mes)	-	20	1

Tabla 2.2: Presupuesto *hardware*.

Una vez enumerados todos los elementos de *hardware* que vamos a utilizar, procedemos a calcular el coste que conlleva el uso de cada uno de ellos. Para ello, utilizaremos la siguiente fórmula, que nos permite realizar un cómputo ponderado del coste de los elementos de *hardware* utilizados:

$$\text{Coste (euros)} = \frac{\text{Coste total (euros)}}{\text{Vida útil (meses)}} \times \text{Porcentaje de uso} \times \text{Tiempo de uso (meses)} \quad (2.1)$$

Al aplicar la fórmula (2.1) a los datos de la tabla 2.2, podremos obtener el coste total del *hardware* para este proyecto.

- Ordenador portátil:

$$\text{Coste (€)} = \frac{799}{48} \times 0,75 \times 4 = 49,94 \text{ €}$$

- Wi-Fi:

$$\text{Coste (€)} = 15 \times 0,20 \times 4 = 12 \text{ €}$$

- Máquina virtual BBDD:

$$\text{Coste (€)} = 20 \times 0,50 \times 3 = 30 \text{ €}$$

- Máquina virtual despliegue:

$$\text{Coste (€)} = 30 \times 0,20 \times 1 = 6 \text{ €}$$

Por tanto, en total, el presupuesto del *hardware* sería el siguiente:

$$\begin{aligned} \text{Coste hardware (€)} &= \text{Coste (portátil)} + \text{Coste (Wi-Fi)} + \text{Coste (BBDD)} + \text{Coste (despliegue)} = \\ &= 97,94 \text{ €} \end{aligned}$$

2.4.2. Software

Para el software, el presupuesto será reducido, ya que la mayoría de las herramientas empleadas en el desarrollo de este proyecto tienen licencia gratuita. En la siguiente tabla, vemos las herramientas que se utilizarán para llevar a cabo este proyecto.

Herramienta	Coste por mes	Porcentaje uso	Uso en meses
Teams	0	100	4
Trello	0	100	4
Overleaf	14,90	20	4
Visual studio code	0	100	4
PyCharm	9,90	100	3

Tabla 2.3: Presupuesto *software*.

De forma análoga al caso del *hardware*, utilizaremos una fórmula para realizar los cálculos ponderados. En este caso, el coste total del software coincide con el coste del único software utilizado que no tiene una licencia gratuita.

$$\begin{aligned}
 \text{Coste } \textit{software} (\text{€}) &= \text{Coste (PyCharm)} + \text{Coste (Overleaf)} \\
 &= 9,90 \times 1,00 \times 3 + 14,90 \times 0,20 \times 4 = 41,62 \text{ €}
 \end{aligned}$$

2.4.3. Recursos humanos

Para completar el presupuesto para este proyecto, nos falta estimar los recursos humanos necesarios para completar el proyecto. En este proyecto, el estudiante ha desempeñado varios roles, que han sido los siguientes:

- **Analista funcional:** se encarga de realizar el análisis, además de estudiar el contexto al completo del proyecto. En relación con la metodología utilizada se encargaría de parte de los objetivos de Proyecto, Antecedentes y parte del Desarrollo (Análisis y Diseño).
- **Desarrollador *frontend*:** responsable del desarrollo de la parte visual de la aplicación. Encargado de parte del Desarrollo (Construcción y Evaluación).
- **Desarrollador *backend*:** análogo al anterior, trabajará en el desarrollo de lógica que soporta el desarrollo de la aplicación.
- **Tester de *software*:** encargado de *testear* la aplicación, garantizando su correcto funcionamiento. Parte del objetivo de Aceptación debería ser desarrollado por este rol.

Nótese que no se ha requerido de ningún especialista para la documentación del proyecto, suponiendo que todos los profesionales que trabajan en el mismo se encargarían de documentar la parte del proyecto desarrollada por ellos.

Para realizar el cálculo, hemos estimado el porcentaje del proyecto en el que se requerirá la colaboración de cada uno de los roles. Además, hemos supuesto que en total el proyecto se

desarrollará en unas 330 horas (ya que un TFG tiene una extensión de entre 300 y 360 horas). Para cada uno de los roles, se ha considerado el salario medio bruto anual, que se puede consultar en la tabla 2.4, y además hemos considerado que la empresa para la que trabajarían está obligada a dar de alta en la seguridad social a sus trabajadores, teniendo que pagar entre un 30-35 % más de cotización por cada uno de ellos (tomaremos una estimación del 32,5 % extra por trabajador).

En la tabla siguiente se presenta un resumen de los datos necesarios para obtener el presupuesto total para los recursos humanos:

Rol	Salario anual (euros)	Porcentaje participación
Analista funcional	34.000	45
Desarrollador <i>frontend</i>	36.000	15
Desarrollador <i>backend</i>	38.550	35
Tester de <i>software</i>	26.300	5

Tabla 2.4: Presupuesto recursos humanos.

Con estos datos, podemos calcular el presupuesto para cada uno de los roles, utilizando la siguiente fórmula, en la que hemos supuesto que de media una persona trabaja unas 1.824 horas al año:

$$\text{Salario rol} = \frac{\text{Salario anual}}{\text{Horas al año}} \times \text{Horas totales proyecto} \times \text{Porcentaje participación} \quad (2.2)$$

Para calcular el coste final que supondría cada rol para la empresa deberíamos sumar, a este salario, la cotización que paga la empresa a la seguridad social por el trabajador, que hemos considerado que sea del 32,5 %. Con ello, nos quedaría que el coste total es:

$$\text{Coste rol} = \text{Salario rol} + 32,5 \% \times \text{Salario rol} = 1,325 \cdot \text{Salario rol} \quad (2.3)$$

Utilizando las ecuaciones (2.2) y (2.3), podemos calcular cuánto costaría el recurso de cada uno de los roles:

■ **Analista funcional:**

$$\text{Salario analista} = 1,325 \cdot \left(\frac{34000}{1824} \cdot 330 \cdot 0,45 \right) = 3667,62 \text{ €}$$

■ **Desarrollador *frontend*:**

$$\text{Salario frontend} = 1,325 \cdot \left(\frac{36000}{1824} \cdot 330 \cdot 0,15 \right) = 1294,49 \text{ €}$$

■ **Desarrollador *backend*:**

$$\text{Salario backend} = 1,325 \cdot \left(\frac{38550}{1824} \cdot 330 \cdot 0,35 \right) = 3234,43 \text{ €}$$

■ **Tester de *software*:**

$$\text{Salario tester} = 1,325 \cdot \left(\frac{26300}{1824} \cdot 330 \cdot 0,05 \right) = 315,23 \text{ €}$$

Sumando todos, calculamos el presupuesto total estimado para recursos humanos, que sería el siguiente:

$$\begin{aligned}\text{Coste recursos humanos (€)} &= \text{Coste (analista)} + \text{Coste (frontend)} \\ &+ \text{Coste (backend)} + \text{Coste (tester)} = \\ &= 8511,47 \text{ €}\end{aligned}$$

2.4.4. Presupuesto total

El presupuesto total estimado del proyecto lo podemos calcular sumando los tres anteriores, por lo que nos quedaría un total de:

$$\begin{aligned}\text{Presupuesto total (€)} &= \text{Coste (hardware)} + \text{Coste (software)} + \text{Coste (recursos humanos)} = \\ &= 97,94 + 41,62 + 8511,47 = 8651,03 \text{ €}\end{aligned}$$

2.5. Balance temporal y económico

En este apartado presentamos un análisis en el que estudiamos las posibles diferencias entre la planificación temporal y económica con el tiempo y dinero real invertido en el proyecto.

2.5.1. Balance temporal

Como ya mencionamos en el apartado 2.3, la planificación temporal se ha hecho en torno a la división del trabajo en 4 *sprints*, que se extiende hasta una fecha cercana a la entrega y presentación del TFG.

En la planificación indicamos también que el tiempo dedicado al trabajo en cada uno de los *sprints* debería ser similar, al haber seguido la metodología ASAP para organizarnos. A continuación, mostramos el tiempo dedicado en cada *sprint* al proyecto:

- *Sprint* #1: 65 horas
- *Sprint* #2: 73 horas
- *Sprint* #3: 85 horas
- *Sprint* #4: 125 horas

En los siguientes subapartados, veremos un análisis comparativo más detallado entre la planificación realizada en cada *sprint* y el tiempo dedicado a las distintas tareas en ese *sprint*. Para ello, nos basaremos en el diagrama de Gantt obtenido en el cuaderno de trabajo al finalizar cada *sprint*. En este, encontraremos marcado en negro la planificación explicada en el apartado anterior, y en tonos rojos y verdes los días que se ha trabajado fuera o dentro de la planificación de las diferentes actividades, respectivamente.

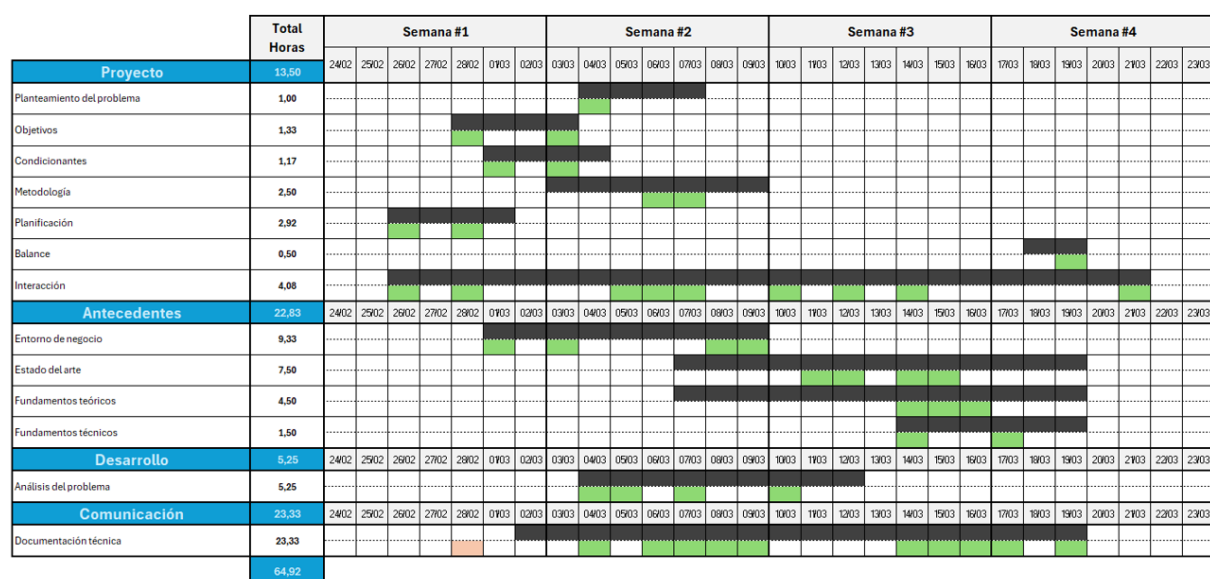


Figura 2.5: Balance del sprint #1.

Sprint 1

Observando la Figura 2.5, vemos una pequeña desviación en la planificación por una causa menor, que fue la disponibilidad de la plantilla para comenzar a redactar la memoria.

En cuanto a los días planificados en los que no hemos trabajado, observando de nuevo la Figura 2.5, la razón principal es el reparto de trabajo entre actividades o la falta de disponibilidad. Cabe destacar que el retraso en el comienzo de las actividades de estado del arte y fundamentos teóricos estuvo marcado por la aparición de ciertas dudas sobre el planteamiento de dichos apartados en la memoria. En cuanto al resto de actividades, salvando pequeñas diferencias, la planificación se ha ajustado a la realidad en mayor medida, por lo que concluimos que la planificación en este *sprint* ha sido satisfactoria.

Sprint 2

Nos centraremos ahora en la Figura 2.6, en la que encontramos bastantes más desviaciones respecto a la planificación en comparación con el *sprint* anterior. Las desviaciones se han producido principalmente por retrasos a la hora de plantear la corrección de algunas actividades del *sprint* anterior y por iteraciones necesarias para alcanzar la versión final del modelo de dominio y el diseño de la base de datos.

Además, en la figura 2.6 vemos que no hemos trabajado en algunas actividades que habíamos planificado, como la de evaluación. Esto es porque, al haber hecho muchas iteraciones sobre el modelo de dominio, se ha decidido no empezar con la programación de la aplicación, de forma que evitemos tener que modificar varias veces el código de la misma. Este retraso también se ha causado por la falta de disponibilidad, ya que este *sprint*, aunque ha tenido una semana más que el anterior, ha estado marcado por días de vacaciones en los que no se ha dispuesto de tanto tiempo para trabajar en el proyecto. Por ello, concluimos que, aunque ha habido un desvío sobre la planificación, se ha conseguido el objetivo de tener una base clara sobre la que comenzar a

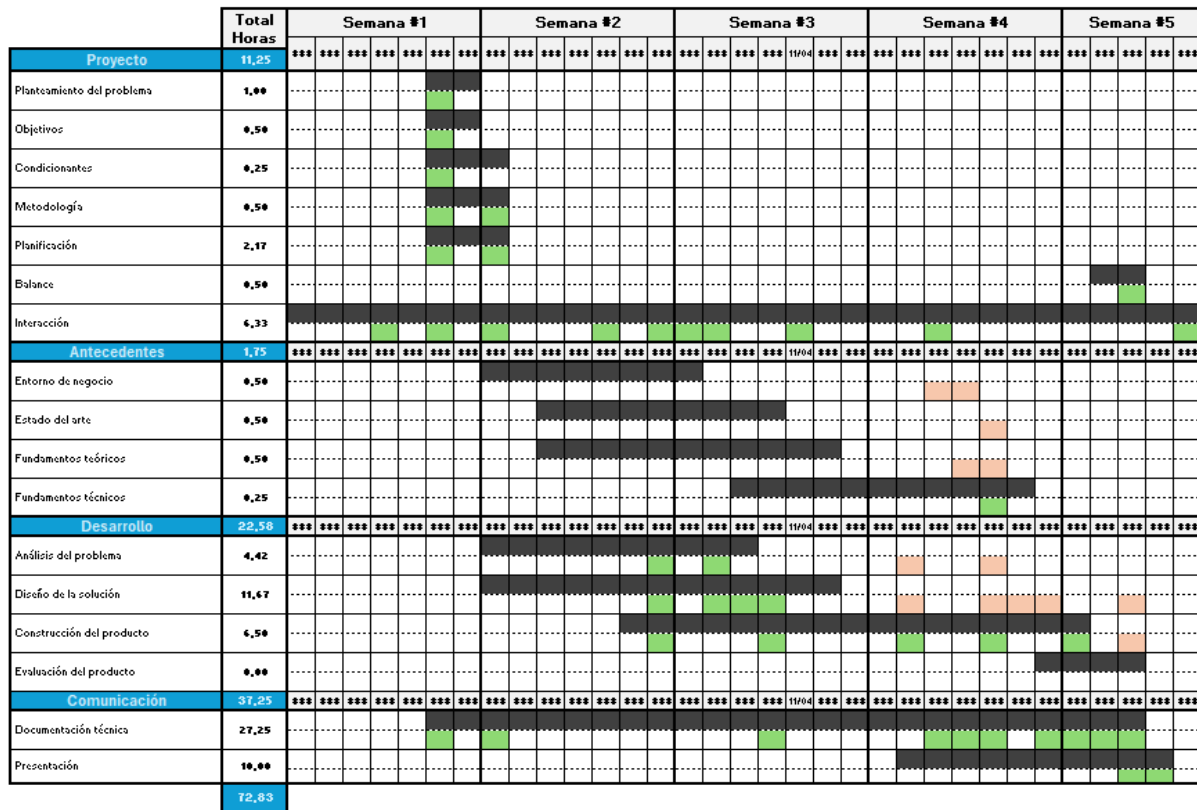


Figura 2.6: Balance del sprint #2.

construir el proyecto, por lo que no debería ser una preocupación no haber cumplido lo indicado en el cronograma de planificación.

Sprint 3

El tercer *sprint* ha sido similar al primero en cuanto al acierto de la planificación, teniendo pocas desviaciones y muy puntuales, ya que casi todo el desarrollo realizado se ha centrado en la parte de la construcción del producto. Como podemos ver en la imagen 2.7, esa es la historia a la que le hemos dedicado casi todo el tiempo en este *sprint*, desarrollando así la aplicación en la que se centra este proyecto.

En este *sprint*, hemos trabajado en todas las actividades planificadas, y las desviaciones que hemos mencionado antes han sido causadas por investigaciones que se han tenido que realizar en puntos muy concretos por el desarrollo de la aplicación que se estaba realizando. Además, el alcance claro de esta aplicación se ha terminado definiendo cerca del final del *sprint*, lo que ha causado cambios hacia el final en la parte de análisis y diseño, que no estaban pensados pero que no suponen mayor problema. Podemos concluir así que la planificación de este tercer *sprint* ha sido correcta y adecuada al desarrollo realizado.

	Total Horas	Semana #1							Semana #2							Semana #3							Semana #4						
		28/04	29/04	30/04	01/05	02/05	03/05	04/05	05/05	06/05	07/05	08/05	09/05	10/05	11/05	12/05	13/05	14/05	15/05	16/05	17/05	18/05	19/05	20/05	21/05	22/05	23/05		
Proyecto	6,42																												
Planificación	0,50																												
Balance	0,25																												
Interacción	5,67																												
Antecedentes	2,50																												
Fundamentos técnicos	2,50																												
Desarrollo	63,75																												
Análisis del problema	5,50																												
Diseño de la solución	1,67																												
Construcción del producto	54,50																												
Evaluación del producto	2,08																												
Comunicación	11,67																												
Documentación técnica	11,67																												
	84,33																												

Figura 2.7: Balance del sprint #3.

Sprint 4

El ajuste que hemos hecho respecto a la planificación en este *sprint* ha sido casi perfecto en cuanto a actividades trabajadas en los periodos de tiempo indicados en la planificación. En la imagen 2.8 podemos ver que la mayoría del tiempo en este *sprint* lo hemos dedicado a la construcción y evaluación del producto, ya que hemos conseguido completar casi el 100 % del desarrollo de la aplicación asociada al proyecto.

En este *sprint*, hemos trabajado en todas las historias planificadas, con pocas desviaciones destacables, y se han dedicado más horas en general que en otros *sprints* por la cercanía con la entrega del proyecto.

2.5.2. Balance económico

En este apartado vamos a presentar el coste real del proyecto, comparándolo con lo que habíamos estimado en los presupuestos presentados en el apartado 2.4. Para ello, haremos una comparativa por cada una de las partes analizadas: *hardware*, *software* y recursos humanos.

Comenzando con el *hardware*, todos los elementos que supusimos que se iban a utilizar finalmente se han utilizado y no ha habido ninguna herramienta extra que haya sido necesaria en este caso. Aún así, planificamos que el proyecto se iba a completar en 4 meses, pero finalmente, por diversos motivos, se ha tenido que aumentar en medio mes más, por lo que algunos costes se han incrementado por esa razón. Por otro lado, el servidor de despliegue finalmente solo se ha utilizado en la parte final del proyecto, en concreto en el último medio mes extra, lo que supone una reducción del coste en esa parte.

Por tanto, el coste real del *hardware* ha sido el siguiente, donde se ha utilizado para calcular cada uno de los costes individuales la ecuación (2.1) expuesta en el apartado del presupuesto:

- Ordenador portátil:

$$\text{Coste (€)} = \frac{799}{48} \times 0,75 \times 4,5 = 56,18 \text{ €}$$

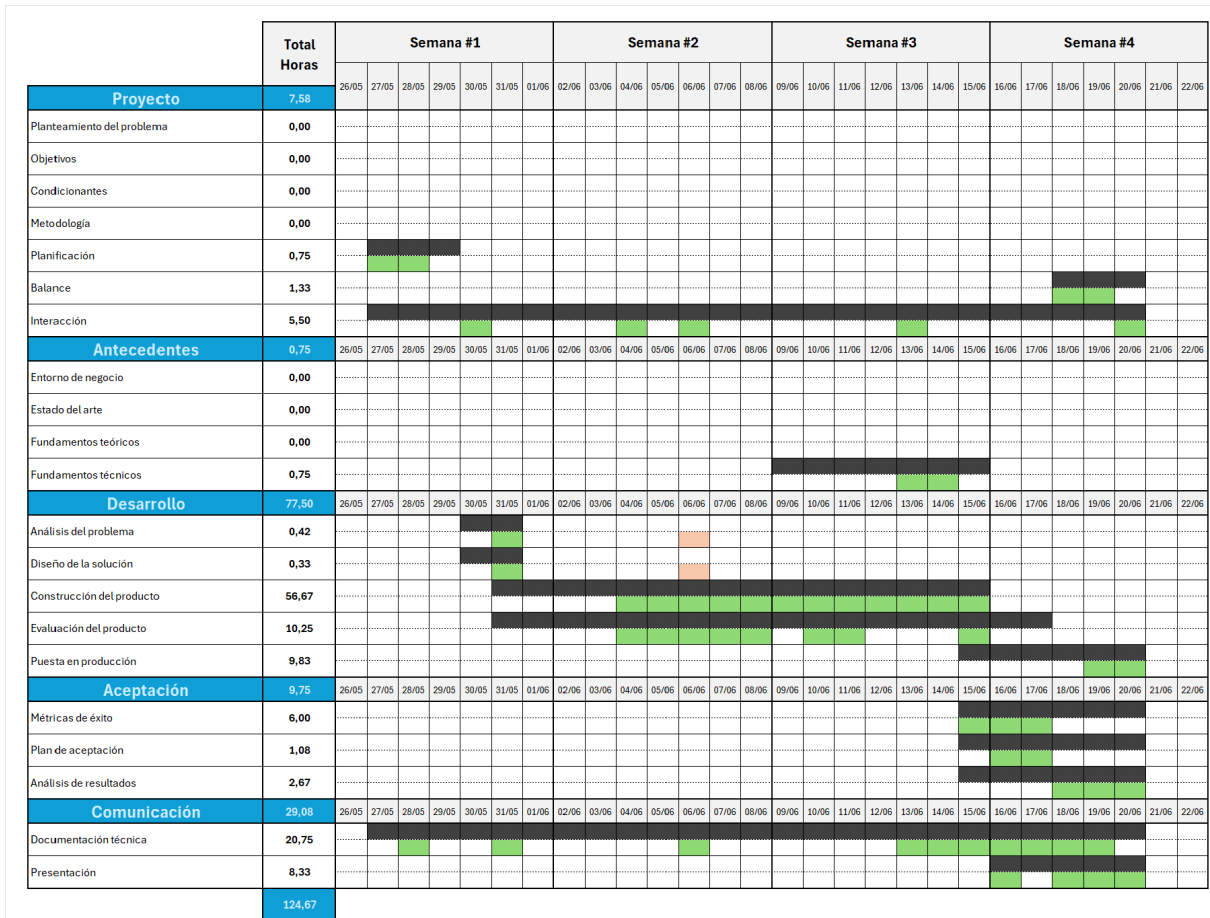


Figura 2.8: Balance del sprint #4.

- Wi-Fi:

$$\text{Coste (€)} = 15 \times 0,20 \times 4,5 = 13,5 \text{ €}$$

- Máquina virtual BBDD:

$$\text{Coste (€)} = 20 \times 0,50 \times 3 = 30 \text{ €}$$

- Máquina virtual despliegue:

$$\text{Coste (€)} = 30 \times 0,20 \times 0,5 = 3 \text{ €}$$

Y calculando el total obtenemos:

$$\begin{aligned} \text{Coste hardware (€)} &= \text{Coste (portátil)} + \text{Coste (Wi-Fi)} + \text{Coste (BBDD)} + \text{Coste (despliegue)} = \\ &= 102,68 \text{ €} \end{aligned}$$

Si realizamos la comparativa con el presupuesto que habíamos diseñado para el *hardware*, que era de 121,94 €, vemos que el coste real ha sido ligeramente inferior, en concreto:

$$\text{Presupuesto} - \text{Coste} = 121,94 - 114,68 = 7,26 \text{ € inferior}$$

Herramienta	Coste total (euros)	Vida útil (meses)	Porcentaje uso (%)	Uso en meses
Ordenador portátil	799	$4 \times 12 = 48$	75	4,5
Wi-Fi	15 (al mes)	-	20	4,5
Máquina virtual BBDD	20 (al mes)	-	50	3
Máquina virtual despliegue	30 (al mes)	-	20	0,5

Tabla 2.5: Presupuesto *hardware*.

Para el software, no hemos tenido ninguna variación si comparamos el presupuesto planteado con los costes que hemos tenido finalmente. Esto es porque los únicos *softwares* que no tenían licencia libre, aunque finalmente tardáramos más en empezar a utilizarlos de lo que teníamos pensado, se ha compensado con el tiempo que hemos tenido que alargar el proyecto. Esto hace que el coste del *software* haya sido:

$$\begin{aligned}\text{Coste software (€)} &= \text{Coste (PyCharm)} + \text{Coste (Overleaf)} \\ &= 9,90 \times 1,00 \times 3 + 14,90 \times 0,20 \times 4 = 41,62 \text{ €}\end{aligned}$$

Y en consecuencia, en este caso la diferencia es de 0 € entre el presupuesto y el coste para el *software*

Por último, realizaremos la comparativa entre los presupuestos planteados para los recursos humanos necesarios y los costes reales que han supuesto. En cuanto a los roles requeridos, no se han necesitado otros perfiles que no hubiéramos considerado, aunque sí que se han visto modificadas las horas que nos ha llevado el proyecto y, en consecuencia, el número de horas que ha participado cada uno de ellos. Por ello, cabe destacar que habíamos planificado unas 330 horas de duración del proyecto, que finalmente han sido un total de 348 horas como mencionamos en el apartado 2.5.1.

Con estos datos, podemos calcular el coste que ha supuesto cada uno de los roles, utilizando las fórmulas (2.2) y (2.3), explicadas en el apartado 2.4.3.

■ **Analista funcional:**

$$\text{Salario analista} = 1,325 \cdot \left(\frac{34000}{1824} \cdot 348 \cdot 0,45 \right) = 3867,78 \text{ €}$$

■ **Desarrollador *frontend*:**

$$\text{Salario frontend} = 1,325 \cdot \left(\frac{36000}{1824} \cdot 348 \cdot 0,15 \right) = 1365,09 \text{ €}$$

■ **Desarrollador *backend*:**

$$\text{Salario backend} = 1,325 \cdot \left(\frac{38550}{1824} \cdot 348 \cdot 0,35 \right) = 3410,85 \text{ €}$$

■ **Tester de *software*:**

$$\text{Salario tester} = 1,325 \cdot \left(\frac{26300}{1824} \cdot 348 \cdot 0,05 \right) = 332,43 \text{ €}$$

Y calculando el coste total:

$$\begin{aligned} \text{Coste recursos humanos (€)} &= \text{Coste (analista)} + \text{Coste (frontend)} \\ &\quad + \text{Coste (backend)} + \text{Coste (tester)} = \\ &= 8976,15 \text{ €} \end{aligned}$$

Por tanto, sabiendo que el presupuesto inicial que habíamos planteado para los recursos humanos necesarios era de 8511,47 €, vemos que ha habido un pequeño sobrecoste respecto a los presupuestos planificados. La razón principal de esto es el exceso de horas empleadas en el proyecto en comparación con las que habíamos planificado.

Una vez vista la comparativa para cada uno de los puntos considerados, podemos hacer una comparativa general del presupuesto respecto al coste final que ha tenido el proyecto. Para ello, recordemos que en el presupuesto que habíamos realizado teníamos que el coste estimado del proyecto sería de 8651,03 €. Calculando el coste total que ha tenido el proyecto:

$$\begin{aligned} \text{Presupuesto total (€)} &= \text{Coste (hardware)} + \text{Coste (software)} + \text{Coste (recursos humanos)} = \\ &= 114,68 + 29,70 + 8976,15 = 9120,45 \text{ €} \end{aligned}$$

Podemos concluir que las horas extra respecto a las planificadas han hecho que se incremente un poco el coste, tanto en la parte del *hardware* como en los recursos humanos. En este caso, el incremento de coste es mínimo, solo un 1 % respecto al coste planificado, por tanto podemos concluir que los presupuestos realizados al inicio del proyecto son correctos y se adaptan a lo desarrollado posteriormente.

Capítulo 3

Antecedentes

Este tercer capítulo del TFG contiene todos los conocimientos básicos necesarios para entender el contexto en el que se desarrolla el proyecto, además de algunas comparativas con sistemas que podrían ser similares al producto final que se pretende obtener. Realizaremos una introducción al contexto educativo en el que se enfoca este trabajo (*eXtreme Learningy Test-Driven Learning*) y al campo de *Learning Analytics*, además de comentar algunos fundamentos generales tanto teóricos como técnicos.

3.1. Contexto educativo del trabajo

Antes de entrar en detalle de los conceptos y metodologías en los que se basa el proyecto, daremos un contexto sobre el ámbito universitario en el que nos encontramos en el punto de realizar este trabajo. Este Trabajo Fin de Grado (TFG) se centra en la asignatura *Sistemas de Bases de Datos*, del Grado en Ingeniería Informática de Servicios y Aplicaciones, impartida en el campus de Segovia de la Universidad de Valladolid (UVa).

En esta asignatura se sigue una metodología “ágil”, que combina elementos de distintas metodologías como *eXtreme Programming* o *Scrum*, y que permite a los estudiantes disfrutar de una enseñanza mucho más personalizada, además de poder tener un *feedback* constante de su desempeño en las distintas actividades de evaluación que se llevan a cabo. En esta asignatura se ofrece, además, la posibilidad de realizar solo un examen y entrega de proyecto final, de forma que cualquier alumno pueda elegir la metodología que le sea más adecuada para su situación personal. Esta dinámica de aprendizaje se conoce como *eXtreme Learning* y esta asignatura, en particular, la materializa mediante la metodología *Test-Driven Learning* (TDL) [12]. Por lo tanto, TDL establece la base sobre la que construiremos este proyecto, ya que utilizaremos datos sobre el proceso de aprendizaje de los estudiantes en el último año que se ha impartido la asignatura.

3.1.1. eXtreme Learning

eXtreme Learning aborda la construcción de un producto de aprendizaje en el ámbito de una asignatura. Para desarrollar este producto de aprendizaje, se adaptan diferentes prácticas ágiles, muy utilizadas en ámbitos productivos, con las que un estudiante puede aprender y mejorar de forma incremental, en base a diferentes tipos de “acciones” que se llevan a cabo durante el curso. Para la especificación del producto, se proponen cuatro niveles de granularidad: visión, objetivos

de aprendizaje, historias de aprendizaje y criterios de aceptación, explicados en las siguientes secciones.

3.1.1.1. Producto de aprendizaje

Como hemos mencionado antes, explicaremos en este apartado los objetivos de aprendizaje, las historias de aprendizaje y los criterios de aceptación, centrándonos además en su aplicación en particular a la asignatura de *Sistemas de Bases de Datos*. Antes de ello, daremos una definición básica de la visión, base sobre la que se construyen el resto de elementos del producto de aprendizaje.

Visión

La visión, tal y como se define en [11], ofrece una descripción general del producto de aprendizaje, alineada con el objetivo global de la asignatura. Esta descripción debe ser breve, clara y estar orientada desde la perspectiva del cliente, entendido como la sociedad. De este modo, la visión se fundamenta en la justificación de la asignatura dentro del plan de estudios y proporciona al estudiante una referencia que le permita valorar su aprendizaje en un contexto de desarrollo profesional.

Objetivos de aprendizaje

Los objetivos de aprendizaje [11] refinan la visión y establecen los hitos principales del producto. Estos permiten desglosar el alcance de un proyecto, de forma que, en vez de abordarlo “como un todo”, se trate parte por parte. En cada uno de los objetivos se debe tener una descripción clara de manera que el estudiante comprenda fácilmente cómo el objetivo se integra en el producto de aprendizaje final y lo que aporta al proyecto completo. Como ya hemos mencionado en la introducción, los objetivos de aprendizaje se pueden desglosar en historias de aprendizaje, siendo los objetivos un “contenedor” de historias.

Aplicando esto al caso particular que nos afecta, en la asignatura de *Sistemas de Bases de Datos*, los objetivos se asemejan a los bloques temáticos definidos en las guías docentes de otras asignaturas. En este caso, tendremos 5 objetivos principales que representan los cinco grandes bloques de contenidos que se tratan en la asignatura. Para ver el ejemplo concreto de la asignatura mencionada, podemos observar la imagen 3.1, en la que se puede encontrar la visión de la asignatura, así como los 5 objetivos principales y un ejemplo de la forma en que se divide el objetivo del diseño lógico.

Historias de aprendizaje

En *eXtreme Learning* [11] también se conocen como subobjetivos, ya que establecen divisiones específicas dentro del objetivo de aprendizaje. En el contexto de los proyectos llevados a cabo con metodologías ágiles, cada una de estas historias representaría “una característica funcional” del producto, por lo que tienen un alcance acotado, normalmente definido por una descripción completa que permite al alumno conocer con exactitud qué debe conseguir para completar dicha historia.

Aplicándolo al ejemplo que nos afecta para este trabajo, dentro de cada uno de los cinco objetivos que habíamos marcado previamente, que eran como los “temas” de la asignatura, tendríamos unos objetivos a cumplir en cada uno de los temas. Estos objetivos, para *eXtreme Learning*, se consideran historias de aprendizaje, pudiendo encontrar un ejemplo en la imagen 3.1,

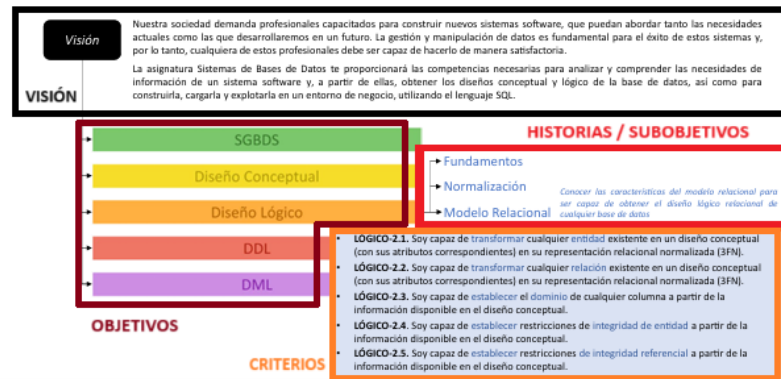


Figura 3.1: Ejemplo del producto de aprendizaje de la asignatura de *Sistemas de Bases de Datos*. Dada una visión general de la asignatura, encontramos los 5 objetivos en los que se divide. Además, podemos ver que el objetivo de diseño lógico se subdivide en tres historias de aprendizaje. Una de esas historias (Modelo Relacional) se descompone a su vez en 5 criterios que determinan el alcance de dicha historia. [11].

en la que vemos cómo el objetivo de diseño lógico está dividido en tres historias de aprendizaje: Fundamentos, Normalización y Modelo Relacional.

Criterios de aceptación

Las historias de aprendizaje comentadas en el apartado anterior también se pueden considerar unos “contenedores” de otra división menor, que, en este caso, serían los criterios de aceptación. Estos se pueden tratar como las “piezas” del aprendizaje que se tienen que ensamblar para construir una historia [11]. Cada uno de estos criterios debe estar especificado de forma que se pueda medir fácilmente, de manera que un estudiante pueda saber qué criterios satisface y cuáles de ellos no cumple, y así poder entender los puntos en los que debe mejorar. Esta forma de construir los criterios permite que se genere un *feedback* para cada alumno sobre la evaluación del producto de aprendizaje que haya entregado. Por último, cabe destacar que estos criterios, por la forma en la que se han construido, ya son indivisibles, es decir, son el nivel más bajo al que se desglosa el producto de aprendizaje y, en nuestro caso, la asignatura.

3.1.1.2. Proceso de aprendizaje

En el apartado 3.1.1.1 vimos cómo desglosar un producto de aprendizaje, y ahora nos centraremos en el proceso hasta conseguir una versión final de dicho producto. La metodología seguida en la asignatura divide el proceso de aprendizaje en dos etapas: aprendizaje y revisión del proyecto. Estas dos etapas son secuenciales, produciéndose la segunda una vez se finalice la primera, y tienen objetivos diferentes. La etapa de aprendizaje tiene como objetivo que los alumnos adquieran todos los conocimientos planteados en la asignatura, mientras que la etapa de revisión consistirá en la validación del producto, observando si satisface o no la visión del proyecto planteado.

eXtreme Learning divide la etapa de aprendizaje en varios *sprints* de corta duración (alrededor de 5 semanas). Cada uno de estos *sprints* se divide en 4 etapas: Inicio, Trabajo, Evaluación y Retrospectiva. A lo largo de todas estas etapas, y principalmente durante las de Trabajo y Evaluación, se desarrollan diversas “acciones formativas” manejadas por los profesores de la asignatura.



Figura 3.2: Proceso de aprendizaje de Sistemas de Bases de Datos [11].

natura y cuyo objetivo es que los estudiantes alcancen el objetivo del *sprint* correspondiente. Estas acciones serán o bien de aprendizaje, como puede ser cualquier clase desarrollada a lo largo de la asignatura; y también de evaluación, que se ejecutarán durante el *sprint* para afianzar los conocimientos adquiridos durante las acciones de aprendizaje. Estas acciones de evaluación pueden ser resoluciones de problemas o evaluaciones a nivel de objetivos que se realizarán una vez por *sprint*.

En la imagen 3.2 podemos ver un ejemplo de cómo se podría estructurar la división en *sprints* en una metodología basada en *eXtreme Learning*.

3.1.1.3. Entorno de aprendizaje

Para que un proyecto se desarrolle con éxito, es necesario que los participantes del mismo tengan un entorno de trabajo con todos los recursos necesarios para completar sus tareas satisfactoriamente. En el caso de *eXtreme Learning*, esto se adapta al contexto del aprendizaje, habilitando ciertas herramientas que facilitan la interacción y el trabajo colaborativo de los estudiantes y con el profesorado.

Este entorno, que llamaremos entorno de aprendizaje [11], tiene cuatro puntos clave que son:

- Asegura que todos los estudiantes tiene una visión compartida y completamente actualizada del estado del proyecto de aprendizaje.
- Facilita el acceso a todos los recursos de aprendizaje necesarios para el desarrollo del proyecto
- Fomenta la comunicación colectiva entre los participantes del proyecto.
- Proporciona una visión global del esfuerzo de los estudiantes y su distribución temporal.

Para conseguir un entorno de aprendizaje como este, es necesario disponer de varios recursos que faciliten cumplir las condiciones anteriores. Para ello, en *eXtreme Learning*, se aprovechan herramientas gratuitas y otras licencias proporcionadas por la universidad, para disponer de:

- **Tablero de trabajo:** se utiliza habitualmente la herramienta gratuita *Trello* para especificar las tareas a realizar durante el *sprint*. Para ello se utiliza la distribución habitual de los tableros *Kanban*, con las columnas *Backlog*, *To Do*, *Doing*, *Blocked* y *Done* para organizar las tareas e historias que se van planteando durante el *sprint*.

- **Cuaderno de trabajo:** en este elemento, habitualmente dispuesto en una hoja de cálculo, los estudiantes pueden ir registrando el tiempo que dedican a las diferentes tareas a realizar en la asignatura. Con esta información se puede llevar un seguimiento de las tareas que más tiempo llevan al estudiante y mantener un registro de las tareas realizadas por cada uno de los alumnos en los diferentes días del *sprint*.
- **Espacio de trabajo compartido:** en la asignatura se tiende a utilizar la herramienta *Teams* que permite la compartición de los recursos de los estudiantes con los profesores. En estos espacios de trabajo los alumnos pueden realizar sus entregas a los profesores de forma sencilla, y los profesores pueden aportar un *feedback* de manera dinámica.
- **Canales de comunicación:** utilizando la misma herramienta *Teams* los estudiantes y profesores podrán mantener una comunicación activa y rápida. Se utilizarán los canales disponibles para cualquier aviso, de forma que todo esté centralizado en la misma aplicación.

Para completar este apartado, cabe destacar que todos los elementos explicados desde el comienzo de 3.1.1 cobrarán todavía más sentido en los próximos apartados cuando expliquemos la base de datos utilizada para el desarrollo del proyecto. Elementos como los objetivos de aprendizaje, historias de aprendizaje, criterios de aceptación, las acciones o el cuaderno de trabajo serán claves, formando la base de la información que utilizaremos para desarrollar el proyecto.

3.1.2. Test-Driven Learning (TDL)

Test-Driven Learning es la materialización en forma de metodología que surge a partir del marco conceptual definido por *eXtreme Learning*, incorporando un enfoque sistemático para la evaluación y retroalimentación del aprendizaje. Mientras que *eXtreme Learning* proporciona la estructura base del producto de aprendizaje y los *sprints*, *Test-Driven Learning* complementa este marco con un sistema de pruebas continuas, autoevaluación y reflexión crítica. Esta metodología busca que los estudiantes adquieran una conciencia plena de su aprendizaje, activando procesos de pensamiento que les permitan analizar y mejorar su desempeño de forma autónoma.

La primera de las novedades que propone *Test-Driven Learning* son tres tipos de pruebas, integradas en los *sprints* del proceso de aprendizaje:

- **Pruebas básicas:** Evalúan historias de aprendizaje individuales mediante actividades prácticas, como resolución de problemas. Se realizan tras las sesiones de laboratorio y permiten a los estudiantes identificar errores tempranos. Incluyen autoevaluación guiada por plantillas (rúbricas).
- **Pruebas de control:** Actúan como exámenes parciales, evaluando uno o varios objetivos de aprendizaje. Su diseño fomenta la profundización en los conceptos y la conexión entre ellos.
- **Pruebas finales:** Cubren la totalidad de un objetivo de aprendizaje y se realizan al finalizar el *sprint* correspondiente. Su evaluación es sumativa, determinando el nivel de dominio alcanzado.

En cada prueba planteada en *Test-Driven Learning* tenemos tres partes claves que son novedosas también de esta metodología. Estas son las siguientes:

- **Tarea:** Actividad alineada con los criterios de aceptación del producto de aprendizaje.
- **Autoevaluación:** Los estudiantes comparan sus soluciones con una referencia proporcionada por el profesor, utilizando hojas de corrección basadas en rúbricas. Esto les permite identificar aciertos y errores de forma estructurada.
- **Análisis crítico:** Mediante rutinas de pensamiento como *Compara-contrasta* o *KWL*, los estudiantes reflexionan sobre su aprendizaje, identifican áreas de mejora y establecen metas futuras.

A diferencia de *eXtreme Learning*, que se centra en la estructura del aprendizaje, *Test-Driven Learning* proporciona las herramientas para visibilizarlo y mejorarlo de forma continua, cumpliendo así la promesa de ofrecer a los estudiantes una conciencia plena de su proceso de aprendizaje. Esta metodología se ha aplicado a un caso particular en la Universidad de Valladolid, que es la asignatura de *Sistemas de Bases de Datos*, impartida en el Grado en Ingeniería Informática de Servicios y Aplicaciones en el campus de Segovia. Los profesores que han aplicado esta metodología han concluido que los estudiantes desarrollan una autopercepción precisa de su aprendizaje; que se fomentan movimientos de pensamiento crítico, como razonar con evidencias o captar lo esencial del aprendizaje; y que se mejora la calidad del producto final en los trabajos realizados por los alumnos.

3.1.3. Retos de *eXtreme Learning*

En el transcurso de la asignatura, se genera un gran volumen de información, y el tratamiento de esta es lo que supone una mayor inversión de tiempo para los profesores de la asignatura. En concreto, del entorno de aprendizaje mencionado antes, procesaremos en nuestro caso, por ejemplo, los cuadernos de trabajo. Actualmente, cada estudiante genera tres cuadernos de trabajo por curso, en cada uno de los cuales hay decenas de líneas de información con las tareas realizadas por cada uno de los estudiantes. Considerando que hay unos 50 estudiantes por curso, y que cada uno de ellos realiza unas 35 tareas que indica en el cuaderno, se generarían más de 5.000 registros de tareas de los estudiantes por curso.

Teniendo en cuenta que al cuaderno de trabajo tenemos que añadirle otras fuentes de datos, como pueden ser las evaluaciones de los profesores, las autoevaluaciones realizadas por los estudiantes o la información de las distintas acciones de formación, la cantidad de información a analizar y su distribución se convierte en un aspecto muy difícil de abordar. Esto supone un reto para los docentes de la asignatura, ya que es muy difícil poder hacer un seguimiento exhaustivo de lo que ha hecho cada estudiante. Además, tener una visión general del estado a nivel de aula también se antoja complicado, dificultando la toma de decisiones basadas en evidencia que permitan mejorar el diseño de la asignatura o intervenir en caso de necesidad.

3.2. Fundamentos teóricos

Este trabajo se motiva en el interés por parte de los docentes en entender y mejorar los procesos de enseñanza-aprendizaje partiendo de los datos recopilados durante el curso. Esta motivación se alinea con los intereses perseguidos en el campo de las analíticas de aprendizaje (en inglés *Learning Analytics*). Según la asociación encargada de la investigación en el campo de *Learning Analytics*, SoLAR (*Society for Learning Analytics Research*), el propósito de *Learning*

Analytics es “entender u optimizar el aprendizaje y los entornos en los que ocurre” [24]. *Learning Analytics* tal como lo definieron en 2011 por primera vez [24], abarca todo el proceso de medición, recopilación, análisis y presentación de los datos tanto de los estudiantes como del contexto de estudio del que disponen, utilizando posteriormente estos datos para comprender y optimizar el aprendizaje dado a los alumnos, además de mejorar los entornos en los que los estudiantes desarrollan su aprendizaje.

3.2.1. Learning Analytics

Learning Analytics ha sido históricamente utilizado para la predicción del rendimiento académico, en particular para identificar a estudiantes en riesgo de fracaso o abandono escolar. No obstante, sus aplicaciones van mucho más allá. Entre los principales usos de *Learning Analytics* se encuentran: el fomento de estrategias de aprendizaje autónomo y continuo mediante el análisis del proceso de aprendizaje seguido por el alumno y los artefactos generados; la provisión de *feedback* personalizado, que permite a docentes y estudiantes ajustar sus estrategias de enseñanza-aprendizaje; la mayor comprensión y guiado de tareas que promuevan habilidades clave como la colaboración, el pensamiento crítico, la creatividad y la comunicación; la promoción de la autorreflexión mediante el análisis de datos propios para identificar fortalezas y áreas de mejora; y el incremento de la calidad educativa gracias a la reflexión sobre la práctica docente y el currículo, y su posterior rediseño. Estas aplicaciones, cuando se implementan adecuadamente, no solo benefician al alumnado, sino también al profesorado y a las instituciones educativas, al permitir la toma de decisiones informadas [23].

3.2.1.1. Métodos de análisis

Una vez los datos han sido recopilados, en *Learning Analytics* se utilizan diferentes métodos de análisis en función de los datos disponibles y el propósito del análisis.

- **Análisis descriptivo:** Esta metodología está centrada en el estudio de los datos pasados, realizando una retrospectiva en la que se analizan datos recogidos durante un periodo de tiempo fijado.
- **Análisis diagnóstico:** En este caso la metodología tiene como objetivo analizar el porqué de los resultados obtenidos.
- **Análisis predictivo:** Para poder realizar predicciones es necesario utilizar modelos estadísticos y algoritmos para prever comportamientos futuros, como identificar a estudiantes en riesgo de no completar un curso.
- **Análisis prescriptivo:** Si disponemos de datos históricos y predicciones futuras, podemos realizar una comparativa entre ambos para encontrar los puntos gracias a los cuales se ha logrado el éxito, o aquellos aspectos que es necesario mejorar.

En el caso concreto de este proyecto, utilizaremos analíticas descriptivas que nos ofrezcan una visión global de lo ocurrido durante el curso, así como la relación entre las diferentes variables de estudio.

3.2.1.2. Impacto en la sociedad actual

Learning Analytics implica a diversos *stakeholders*. Los actores clave son los estudiantes, los docentes y las instituciones académicas, aunque también hay otras personas que intervienen como los desarrolladores de las herramientas, las familias de los menores que las usan o los responsables de la privacidad de los datos en cada institución. Cada uno de ellos se ve afectado de manera diferente por el uso de *Learning Analytics*: los estudiantes reciben información y recomendaciones personalizadas que pueden ayudarles a tomar decisiones sobre su trayectoria académica; el profesorado obtiene herramientas para mejorar la enseñanza y comprender mejor las necesidades del alumnado; y las instituciones educativas utilizan los datos para rediseñar planes de estudio y mejorar sus políticas educativas, como ocurre en enfoques específicos como el *Curriculum Analytics* [3].

En este trabajo, los *stakeholders* principales serán los docentes de la asignatura, quienes se beneficiarán mediante la automatización de tareas repetitivas y el acceso a análisis que les permitirán adaptar su enseñanza de forma más eficiente y basada en evidencias.

3.2.2. Multimodal Learning Analytics (MMLA)

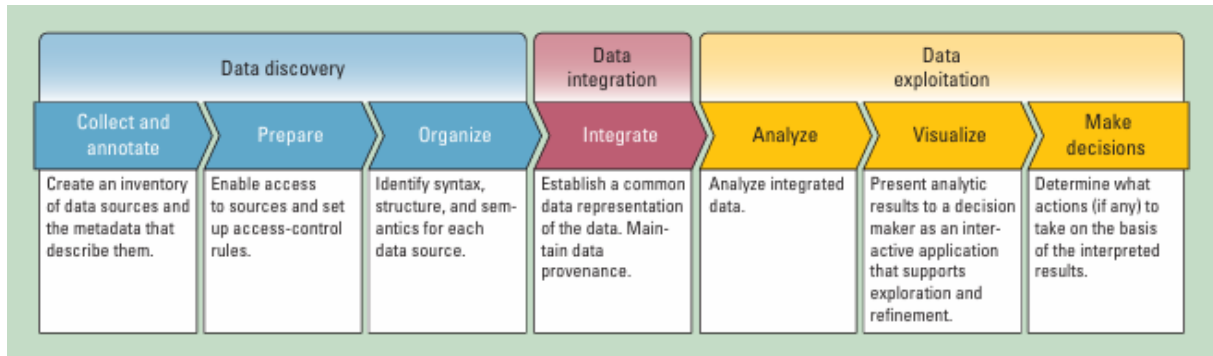
Dentro del campo de *Learning Analytics*, existe una corriente conocida como *Multimodal Learning Analytics* (MMLA) [21], cuyo objetivo es presentar una visión más holística de los fenómenos estudiados a través de la recogida, procesamiento y análisis de múltiples fuentes de datos que aportan información complementaria sobre el proceso de aprendizaje. Estos datos pueden ser recogidos de muchas formas, como espacios digitales con los que interactúan tanto alumnos como docentes, gracias a sensores físicos situados en los distintos entornos de aprendizaje, o incluso de manera autoreportada por los participantes. Estas diferentes fuentes de datos permiten dar una imagen más completa de cómo se están desarrollando los procesos de enseñanza-aprendizaje, ya que recopilan evidencia de los distintos modos y medios en los que estos ocurren.

3.2.2.1. Multimodal Data Value Chain (M-DVC)

Para organizar el complejo proceso del procesamiento de datos, se ha definido lo que se denomina como *Data Value Chain* (DVC), que en el contexto del *data mining* y el *Big Data*, se divide en siete actividades [13], que se pueden ver también en la Figura 3.3.

1. Recogida y anotación (*Collect and Annotate*)
2. Preparación (*Preparation*)
3. Organización (*Organization*)
4. Integración o Fusión (*Integration*)
5. Análisis (*Analysis*)
6. Visualización (*Visualization*)
7. Toma de decisiones (*Decision making*)

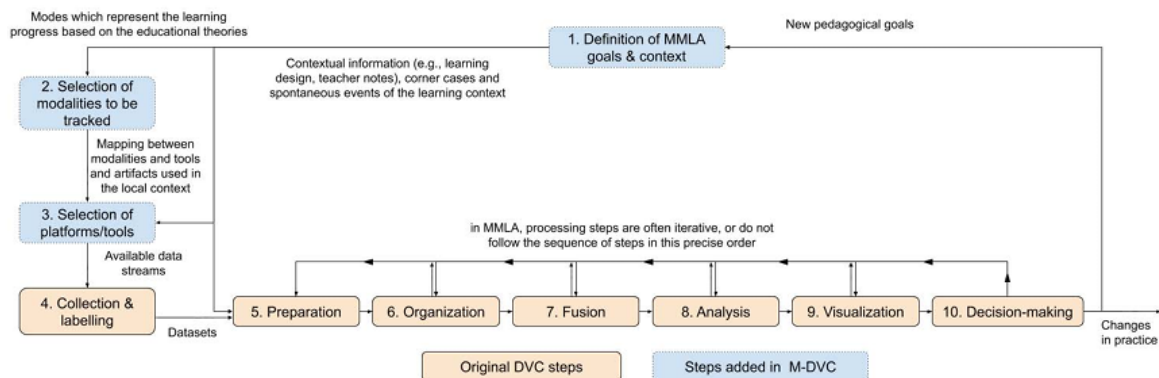
Los nombres entre paréntesis son la nomenclatura original en inglés, presentada para que no haya ninguna confusión con los diagramas que mostraremos posteriormente.

Figura 3.3: *Data Value Chain*: etapas [13].

El concepto de DVC fue introducido a finales del siglo XX por Michael E. Porter, describiéndolo como “un conjunto de actividades que una empresa relacionada con una industria específica lleva a cabo para ofrecer un producto valioso en el mercado” [15]. En el contexto del análisis de datos, la *Data Value Chain* es un marco que describe el ciclo de vida de los datos desde su recolección hasta su utilización en la toma de decisiones. Este enfoque permite estructurar y optimizar el uso de los datos, asegurando que la información, que proviene de datos proporcionados por varios *stakeholders*, está optimizada para su uso en la resolución de problemas y decisiones de calidad.

En este trabajo, nos enfocaremos principalmente en las primeras etapas del DVC, llegando a la integración como contribución principal del TFG, y viendo solo de forma ilustrativa la parte de análisis, visualización y toma de decisiones.

Cuando trabajamos con MMLA buscamos ampliar el análisis del aprendizaje realizado aumentando las fuentes de datos de las que obtenemos la información. Para gestionar esto, en los últimos años, se han desarrollado varias investigaciones [21] que han identificado que la DVC tradicional presenta algunas limitaciones cuando se aplica a escenarios de MMLA (en concreto, falta de información contextual, desafíos en la sistematización de procesos, dificultad en la comunicación entre los stakeholders, procesamiento lineal), lo que ha llevado a la propuesta de un modelo específico para los casos multimodales: el *Multimodal Data Value Chain* (M-DVC).

Figura 3.4: *Multimodal Data Value Chain*: etapas [21].

Este enfoque busca ampliar el análisis del aprendizaje integrando diversas fuentes de datos y considerando el contexto educativo en el que se producen. En la Figura 3.4 se ilustran las fases del M-DVC, diferenciando las heredadas del DVC (en amarillo) y las añadidas específicamente para entornos multimodales (en azul).

A continuación, se describen brevemente las fases del M-DVC:

- 1. Definición de objetivos y contexto de MMLA.** Esta fase inicial, específica del M-DVC, permite establecer los objetivos educativos del análisis y recopilar información contextual relevante del entorno de aprendizaje (diseño instruccional, dinámica del aula, tecnologías disponibles, etc.). Esta información es clave para interpretar adecuadamente los datos recogidos y orientar las decisiones pedagógicas.
- 2. Selección de modalidades.** En esta fase se determina qué tipos de datos se van a analizar (por ejemplo, voz, escritura, interacción digital), en función del contexto definido y los objetivos de análisis. Esta elección condicionará las herramientas y plataformas utilizadas.
- 3. Selección de plataformas y herramientas.** A partir de las modalidades seleccionadas, se escogen las plataformas tecnológicas más adecuadas para capturar y gestionar los datos (sensores, software educativo, cámaras, etc.). Los desarrolladores de software tienen un rol clave en esta etapa. Esta es la última de las etapas novedosas del M-DVC respecto al DVC tradicional.
- 4. Recogida y anotación.** Esta sería la primera etapa del DVC tradicional y consiste en crear un inventario de las fuentes de datos disponibles, añadiendo además los metadatos necesarios para poder verificar la calidad de esas fuentes. Se conoce como recogida a la obtención de datos de las diferentes fuentes, que pueden ser sensores, plataformas digitales, encuestas, etc. Normalmente, estos datos no tienen contexto y suelen ser difíciles de interpretar, por lo que es necesario llevar a cabo la anotación, que es la adición de los metadatos necesarios para describir, entre otros, la procedencia, calidad y formato de los datos. Gracias a estos metadatos podemos facilitar su posterior uso, garantizando que sean comprensibles y trazables. A modo de resumen, en esta etapa se consigue transformar datos no estructurados en datos estructurados con unos metadatos válidos.
- 5. Preparación.** Una vez hemos recolectado los datos necesitamos hacerlos disponibles para que las personas que los requieran puedan utilizarlos para realizar los estudios pertinentes. Estos datos no deben ser accesibles para todos los usuarios, por lo que se deberán establecer controles de acceso a los mismos, de forma que solo las personas con los permisos necesarios puedan visualizarlos y obtener *datasets* de ejemplo de estos datos.

Además, durante la preparación de los datos deberíamos filtrar aquellos que tengan errores muy claros, datos que puedan haberse duplicado o que estén incompletos. Con ello, realizaremos una primera limpieza de los datos para garantizar su calidad y coherencia, y que sean más fáciles de utilizar en posteriores etapas.
- 6. Organización.** El desarrollador que está trabajando con los datos es el encargado de tomar las decisiones de la sintaxis, estructura y semántica de los datos. Para ello, se suelen utilizar repositorios con metadatos y modelos base sobre los que construir la estructura de los datos. Esta organización seleccionada debe plasmarse donde se vayan a estructurar los datos, ya sea una base de datos, un almacén de datos (*data warehouse*) o un *data*

lake. Cuando la estructura esté creada, el siguiente paso de integración será más sencillo, evitando problemas de redundancia o pérdida de información.

7. **Fusión.** La fusión de los datos consiste en integrar todos los datos de las diferentes fuentes que teníamos en el mismo espacio de almacenamiento que hemos organizado en la etapa anterior. En esta etapa, debemos definir como los datos obtenidos de las distintas fuentes se van a relacionar con la representación común. Para ello, debemos diseñar un proceso de transformación que realice agrupaciones y filtrado de los datos, de forma que luego se puedan ingestar en las tablas o espacios reservados para ese tipo de datos.
8. **Análisis.** Esta fase consiste principalmente en que los datos, ya limpios y disponibles en su formato final, son analizados de diferentes formas dependiendo el objetivo final del análisis. Entre otras cosas se pueden buscar patrones o tendencias, aplicar algoritmos de *Machine Learning* o indetificar estructuras ocultas en grandes cantidades de datos (*data mining*). En nuestro TFG, esta será la primera de las etapas que solo mostraremos de forma ilustrativa, ya que el grueso del trabajo se centrará en las etapas anteriores.
9. **Visualización.** La etapa de visualización se encarga principalmente de mostrar los resultados del análisis realizado en la etapa anterior. Estos resultados se podrían consultar en informes estáticos de los datos analizados o, por el contrario, en aplicaciones interactivas que permitan la exploración y refinamiento de los resultados obtenidos. Para el caso concreto de nuestro proyecto, entraremos en esta etapa solo para conseguir mostrar un ejemplo de lo que podríamos obtener con los datos cargados gracias a nuestra aplicación.
10. **Toma de decisiones.** La última etapa consiste en utilizar la información obtenida para guiar la toma de decisiones informadas. Los datos procesados y visualizados permiten optimizar procesos, diseñar estrategias y mejorar la toma de decisiones en distintos ámbitos. El TFG que estamos desarrollando podría servir como punto de partida para futuros trabajos en los que se profundice más en este tema, para lo que se podría utilizar inteligencia artificial o *Machine Learning* con el objetivo de generar predicciones que ayuden a esta toma de decisiones.

El M-DVC introduce varios elementos novedosos respecto al modelo original. Por un lado, las tres primeras etapas (1 a 3) son nuevas y responden a las necesidades específicas de los entornos educativos multimodales. Por otro, se rompe la secuencialidad rígida del DVC tradicional, permitiendo saltos entre fases mediante conexiones recursivas, lo que refleja mejor la naturaleza iterativa del análisis en MMLA. También se conecta directamente la etapa 10 con la etapa 1, cerrando el ciclo y permitiendo adaptar los objetivos según los resultados obtenidos.

Previamente, al hablar del *Data Value Chain* original, mencionamos que en este trabajo nos centraríamos en las etapas de recogida, preparación, organización y fusión de los datos. Adaptando esa afirmación al caso de M-DVC, en este caso nos centraríamos en las etapas de la 1 a la 7 de la Figura 3.4, tocando también las fases 8 y 9 para ilustrar lo que podemos lograr con la base de este TFG en futuras investigaciones.

3.2.3. Analytics layers for learning design

En el ámbito de las analíticas educativas, especialmente aquellas orientadas a informar y mejorar el diseño del aprendizaje, uno de los enfoques más destacados es el *framework Analytics layers for learning design* (AL4LD) [6]. Este modelo organiza las analíticas en tres capas

que permiten entender y apoyar el diseño desde diferentes perspectivas, considerando tanto el propósito como el contexto del análisis y las fuentes de datos utilizadas.

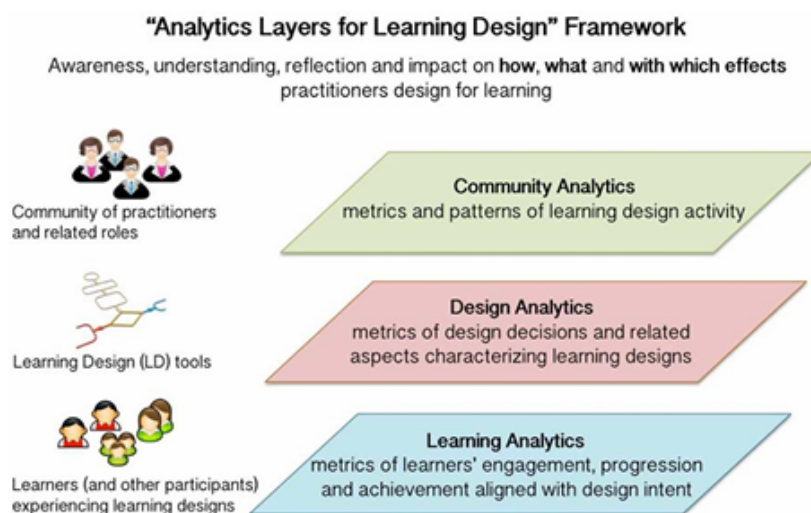


Figura 3.5: Representación gráfica del *framework* de A14LD [6].

La primera capa, *Learning Analytics*, se centra en analizar métricas como la participación, el progreso, el rendimiento y la satisfacción de los estudiantes. Los datos utilizados provienen principalmente del entorno virtual o físico donde tiene lugar el aprendizaje, como los registros de interacción de los estudiantes o sensores en espacios educativos, y se complementan con información institucional como perfiles académicos o encuestas. La segunda capa, *Design Analytics*, se enfoca en examinar el diseño antes de ser implementado, permitiendo reflexionar sobre su estructura pedagógica, la alineación con los objetivos educativos y la carga de trabajo para el profesorado. Las herramientas de diseño digital juegan un papel importante aquí, ya que permiten capturar información sobre los objetivos, las actividades, su secuencia y duración estimada. Por último, *Community Analytics*, la capa superior en la Figura 3.5, busca encontrar patrones de diseño en las actividades académicas compartidas por los docentes en plataformas colaborativas, con el fin de mejorar colectivamente las prácticas pedagógicas mediante el análisis de cómo esos patrones impactan en la comunidad educativa.

Este Trabajo de Fin de Grado se sitúa principalmente en las capa de *Design Analytics*, ya que se centra en analizar los datos generados por los estudiantes durante el desarrollo de una asignatura con el fin mejorar el diseño de aprendizaje. Estos datos incluyen tareas entregadas, evaluaciones y autoevaluaciones realizadas, lo que permite a los docentes automatizar procesos, obtener información relevante y apoyar la toma de decisiones sobre el diseño instruccional a partir de evidencias reales.

3.3. Estado del arte

Dentro del área de las analíticas, en este TFG nos centramos en las analíticas relacionadas con el diseño de aprendizaje [6]. Esto implica que las analíticas que realicemos sobre el aprendizaje

estarán contextualizadas en las decisiones de diseño definidas por los docentes, contribuyendo a que profesores y alumnos entiendan mejor su significado y a una toma de decisiones alineada con los aspectos pedagógicos [16]. En esta sección revisaremos el estado del arte sobre las analíticas del diseño de aprendizaje, explorando sus enfoques, herramientas y aplicaciones en distintos contextos educativos.

Además, dado que en nuestro trabajo integraremos múltiples tipos y fuentes de datos, revisaremos el trabajo realizado en el campo de las analíticas multimodales (es decir, aquellas que integran diferentes orígenes de datos o utilizan diversas formas de análisis [14]).

3.3.1. Descripción de trabajos relacionados

En este apartado, presentaremos algunos trabajos que están directamente relacionados con nuestro proyecto. Extraeremos de ellos tanto los puntos fuertes, en los que nos basaremos para desarrollar el TFG, como las debilidades, que intentaremos evitar y mejorar en el trabajo que estamos realizando.

GLIMPSE

Algunas herramientas, como GLUE!-CAS y GLIMPSE (*Group Learning Interaction Monitor for Pedagogical Scripting Environments*) [16], han sido propuestas en el contexto de entornos de aprendizaje mediados por tecnología, con el objetivo de apoyar tanto a diseñadores instruccionales como a docentes en la toma de decisiones durante el desarrollo de actividades de aprendizaje colaborativo. Estas soluciones permiten cerrar el ciclo entre diseño y análisis, proporcionando información útil en tiempo real que facilita la monitorización y adaptación del diseño pedagógico. Sin embargo, aunque estas propuestas ofrecen un gran valor en escenarios con guiones pedagógicos bien definidos, todavía no se han planteado soluciones adaptadas específicamente a metodologías más flexibles y dinámicas, como es el caso de los proyectos de aprendizaje ágiles.

Un caso representativo de cómo estas herramientas integran el análisis con el diseño lo encontramos en la mencionada herramienta GLIMPSE, donde se muestra cómo esta, junto con GLUE!-CAS, también comentada anteriormente, se conecta con el entorno de aprendizaje para establecer interacciones bidireccionales entre el diseño pedagógico y las capas de *Learning Analytics* presentadas anteriormente [6]. De las dos herramientas, GLUE!-CAS es la encargada de recoger e integrar la evidencia obtenida de diferentes fuentes de datos, como pueden ser entornos de aprendizaje virtuales, registros de participación en las clases o las respuestas de algunos cuestionarios planteados a los alumnos.

Por otra parte, GLIMPSE es una herramienta desarrollada en Java que automatiza el proceso de monitorización basado en guiones pedagógicos. Su funcionamiento se basa en la fusión de dos archivos XML: uno que contiene el guion de aprendizaje generado por GLUE!-PS y otro con la configuración de monitorización, que puede ser creado manualmente o mediante EdiT2++, otra de las herramientas desarrollada junto a las anteriores, pero de menor relevancia en nuestro contexto [16]. A partir de esta información, GLIMPSE aplica heurísticas para filtrar las acciones de los participantes, determina el estado actual de la situación de aprendizaje y lo compara con el estado deseado según el diseño del docente. Finalmente, genera un informe de monitorización en formato HTML, proporcionando alertas sobre desviaciones respecto al diseño previsto, facilitando así la toma de decisiones en tiempo real.

Activity: Actividad 2.1 - Propuesta de proyecto de investigación-acción en grupo pequeño

- Beginning: Tue May 07 16:00:00 CEST 2013
- End: Wed May 08 13:00:00 CEST 2013
- Participation: mandatory-individuals
- Presentiality: blended
- Social level: group

Groups	Participants	Informe del trabajo en grupo pequeño		Propuesta Individual (Estudiante 1)		Propuesta Individual (Estudiante 2)		Propuesta Individual (Estudiante 3)		Propuesta de grupo pequeño (Grupo pequeño 1)		Descripción de la Actividad: Propuesta de investigación-acción en grupo pequeño	Control de asistencia y participación				Warnings	
		link	access	link	access	link	access	link	access	link	access		link	attendance	submission	comment		
Grupo pequeño 1	Alberto		3		1		1		2		5			2		1	*** There is no evidence of Celina using Propuesta Individual (Estudiante 2). This resource must be used by each group member. *** There is no evidence of Alvaro using Propuesta Individual (Estudiante 1). This resource must be used by each group member. *** There is no evidence of Celina using Propuesta Individual (Estudiante 1). This resource must be used by each group member.	
	Alvaro		3		0		1		1		6	5		2		1		1
	Celina		3		0		0		1		5			1		1		

Figura 3.6: Ejemplo de un informe generado por la herramienta GLIMPSE [16].

En la Figura 3.6 encontramos un ejemplo de un informe que podría generar la herramienta GLIMPSE. Estos informes han sido diseñados de forma cooperativa con distintos profesores involucrados en los estudios exploratorios en los que se basa el desarrollo de la herramienta. Esto hace que estén adaptados directamente para el uso sencillo por parte de los docentes, ya que su visualización y uso están pensados para satisfacer las necesidades de estos *stakeholders* [16].

PAMEL

Otra línea de investigación relevante en el campo de las analíticas del aprendizaje es la de las analíticas multimodales, especialmente en contextos en los que intervienen múltiples fuentes de datos para comprender mejor el proceso de aprendizaje. En este sentido, PAMEL (*Processing Architecture for Multimodal Evidence of Learning*) [22] se presenta como una arquitectura diseñada para integrar, estructurar y procesar información contextual procedente de diversas fuentes, facilitando así la toma de decisiones informadas en entornos educativos. Este tipo de propuestas busca dar respuesta a escenarios en los que la riqueza de datos supera las posibilidades de análisis tradicionales, lo que resulta especialmente útil cuando se pretende obtener una visión más completa del aprendizaje más allá de los registros digitales.

PAMEL es una arquitectura orientada a servicios, por lo que tiene dos ventajas respecto a otro tipo de arquitecturas: la primera de ellas es que es extensible para agregar datos de distintas fuentes, ya sean físicas o digitales; y la segunda ventaja es que facilita la reutilización en distintas situaciones de aprendizaje a un coste reducido. Implementando un sistema con arquitectura PAMEL, podemos llevar a cabo la parte del procesamiento de datos del *Data Value Chain* (DVC), como son la preparación, organización y fusión de datos.

Para adaptarse a las soluciones de MMLA, la arquitectura propuesta (ver Figura 3.7) se compone de tres conjuntos de adaptadores, un *PAMEL Manager* y un fichero de configuración. Los diversos adaptadores se utilizan para obtener los datos de las diferentes fuentes de datos que se pueden utilizar, como por ejemplo los *Digital Tool Adapters* para procesar los registros obtenidos de las herramientas digitales de aprendizaje o los *IoT Adapters* para obtener los datos necesarios que se miden con sensores. Estos adaptadores se encargan, además, de transformar dichos datos a un formato que sea sencillo de leer posteriormente, normalmente en forma de tabla, como los CSV. Una vez acaba la acción de estos adaptadores, toda la información se pasa

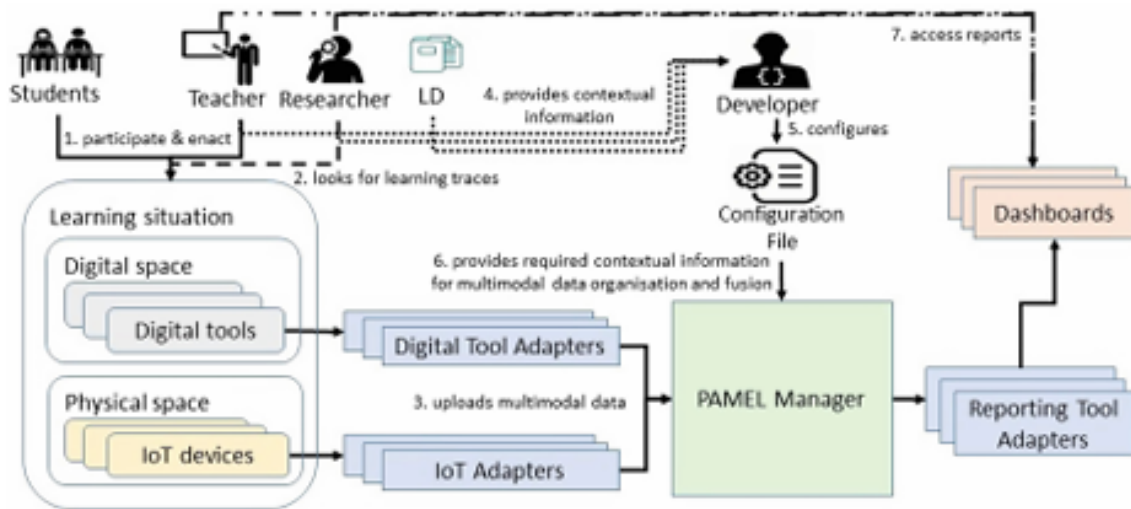


Figura 3.7: Arquitectura de PAMEL [22].

al *PAMEL Manager* que se encarga de fusionar y organizar los datos disponibles. Además de eso, el *PAMEL Manager* es el responsable de distribuir dichos datos hacia el último grupo de adaptadores, llamados *Reporting Tool Adapters*, cuya función es proporcionar las evidencias del aprendizaje obtenidas a los *dashboards*, con los que se planteará una visualización de los datos para su posterior análisis.

Workplace Learning Analytics

El último de los trabajos relacionados con este proyecto plantea una herramienta similar a la que se quiere desarrollar en este proyecto. La herramienta desarrollada en *Workplace Learning Analytics* [25] tiene como objetivo explorar un diseño que proporcione un análisis del aprendizaje en el lugar de trabajo, de forma que pueda ser utilizado principalmente por los gestores de planes de estudio de educación superior en ingeniería. Con estos *dashboard*, el autor pretendía desarrollar una implementación de prueba de concepto para mostrar de qué forma esta tecnología podría ser útil para que los gestores de planes de estudio puedan tomar decisiones sobre posibles modificaciones en los mismos, basándose en la información generada por los estudiantes.

En relación a dicha información, en el ejemplo concreto de la aplicación presentada en este artículo, las fuentes de datos mencionadas son datos generados por los usuarios almacenados en una base de datos, y registros de sistemas que se utilizan para el aprendizaje. Con esta información, se plantea el desarrollo de una aplicación que permita, de forma dinámica, la consulta de ciertos datos mostrando resúmenes de los mismos utilizando gráficos de barras o circulares, entre otros.

En el caso particular de este trabajo, el uso de la aplicación se hace mediante consultas SQL lanzadas directamente desde la interfaz diseñada, aunque se menciona que en futuras versiones se implementará una interfaz más dinámica, y más apta a los *stakeholders* a los que está dirigida la aplicación creada. Esto lo podemos observar en la Figura 3.8, en la que podemos observar en la parte de abajo cómo un usuario podría generarse su propio *dashboard* utilizando directamente

una consulta (*query*) de SQL.

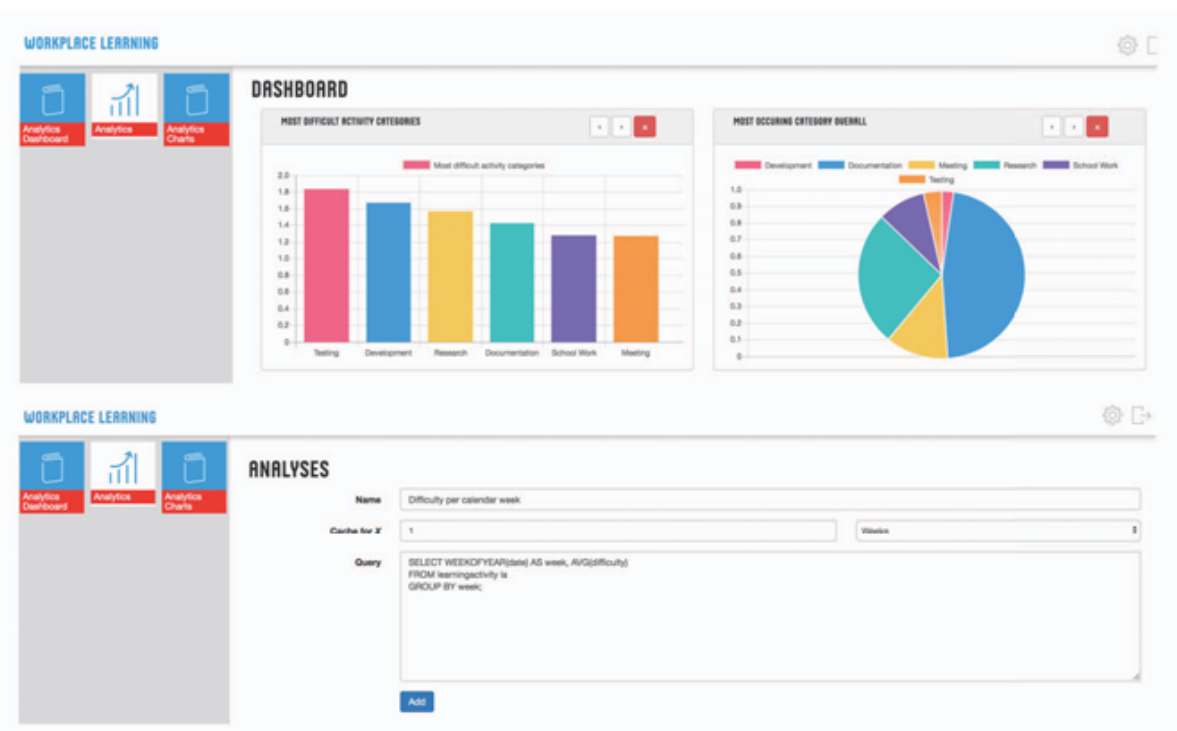


Figura 3.8: Interfaz de la aplicación desarrollada en *Workplace Learning Analytics* [25].

En cuanto a las conclusiones extraídas, después de mostrar esta forma de disponer los datos a los usuarios finales, se obtuvo un *feedback* positivo. Entre los beneficios de este tipo de tecnologías de visualización de datos, se destaca la medición del compromiso, permitiendo ver cómo los estudiantes interaccionan con las distintas herramientas digitales y recursos físicos; toma de decisiones informada, ya que los docentes y gerentes pueden tomar decisiones sobre cambios en planes de estudios con una base sobre la que argumentar dichos cambios; y la personalización del aprendizaje para diferentes tipos de alumnos, consiguiendo así minimizar el porcentaje de abandono de los estudiantes. Por tanto, se concluye en el artículo que el uso de herramientas como esta para realizar analíticas de aprendizaje es muy positivo, mostrando cómo pueden enriquecer la comprensión del proceso de aprendizaje y mejorar la experiencia educativa de los estudiantes.

3.3.2. Discusión

Este apartado está destinado a realizar un análisis comparativo entre los ejemplos mostrados en el apartado anterior y nuestra propuesta, evaluando distintos puntos clave en los que nuestro trabajo puede mejorar y ser innovador respecto a las propuestas ya planteadas por otros autores. A continuación, presentaremos algunos de los problemas que hemos encontrado en otros proyectos, y que pretendemos mejorar en nuestro planteamiento.

Observando los trabajos presentados anteriormente, concluimos que los *dashboards* utilizados en algunos de ellos se han construido siempre partiendo de una sola fuente de datos, por lo que no se obtiene una visión completa, sino parcial, del proceso de aprendizaje [20]. En nuestro proyecto queremos plantear unos *dashboard* que, aunque sean básicos, por no ser el objetivo principal del

proyecto, reciban la información de varias fuentes de datos (hojas de cálculo, *Trello*, *Moodle*, etc.), y sirvan como punto de partida para futuras investigaciones en esta vía.

Continuando en la línea del párrafo anterior, además de variedad en las fuentes de datos, en este proyecto se plantea una innovadora forma de obtener información importante tanto para alumnos como para los docentes, utilizando las rutinas de pensamiento de los alumnos en referencia a las distintas actividades llevadas a cabo en la asignatura. En algunos de los artículos consultados, se hace referencia a encuestas de satisfacción realizadas a los alumnos o al uso de sus resultados en ciertas actividades de aprendizaje, pero en ningún caso referenciando autoevaluaciones realizadas por el propio alumno y de las que se puede obtener información muy valiosa.

Aunque herramientas como GLIMPSE permiten una monitorización con datos multimodales, su especialización a contextos con diseños colaborativos estructurados dificulta su aplicación al caso de las metodologías de aprendizaje ágiles. En este sentido, el presente TFG busca proporcionar a los docentes herramientas para analizar de forma automatizada el desarrollo del aprendizaje con metodologías ágiles, a partir de los datos generados por los alumnos durante la realización de tareas, evaluaciones y autoevaluaciones. De este modo, se pretende facilitar la toma de decisiones docentes, adaptándose mejor al contexto educativo de la asignatura de Bases de Datos en la que se centra este trabajo.

Por otro lado, PAMEL está orientada a permitir una integración avanzada de fuentes de datos físicas y digitales, su foco está más centrado en contextos donde es posible o deseable desplegar infraestructuras tecnológicas complejas, como sensores o sistemas IoT. En contraste, este TFG se enfoca en contextos más cotidianos del entorno universitario, donde se dispone principalmente de los datos digitales generados por los estudiantes al realizar tareas, evaluaciones o autoevaluaciones en plataformas educativas. El objetivo es proponer una solución más ligera y automatizada que permita a los docentes extraer valor de estos datos sin necesidad de infraestructuras adicionales, facilitando la personalización y optimización de la enseñanza, especialmente en metodologías ágiles.

Además, como ya hemos mencionado, la asignatura sobre la que se desarrolla este trabajo sigue una metodología de aprendizaje basada en las metodologías ágiles. Si recordamos los ejemplos mencionados, no se comenta en ningún momento que se utilicen las analíticas de aprendizaje para apoyar metodologías ágiles como estrategia pedagógica de aprendizaje. Es más, en revisiones sistemáticas de la literatura sobre *Learning Analytics* y diseño del aprendizaje, como por ejemplo [8], no se detectan casos de aplicación en contextos de aprendizaje que implementen metodologías ágiles. Por tanto, este es otro punto en el que innovaremos en este TFG, intentando combinar el análisis del aprendizaje con la novedosa metodología que se sigue en *eXtreme Learning*.

En este mismo contexto, a pesar de la importancia de alinear las analíticas y las decisiones de diseño, como ilustran revisiones existentes de la literatura [8], la mayoría de los estudios no tienen en cuenta las decisiones pedagógicas del curso ni la información sobre el contexto. Además, tampoco encontramos en los artículos investigados ninguna referencia en la que se relacionen los datos obtenidos de los alumnos, como las autoevaluaciones o evaluaciones de los docentes, con los objetivos pedagógicos de la asignatura. Por ello, cabe destacar la importancia de este trabajo en modelar no solo la actividad y los datos de los alumnos, sino también el diseño del aprendizaje de la asignatura abordada y plantear la conexión que existe entre ambos.

A la vista de lo cual, en este trabajo propondremos un modelo de datos que nos permite incluir múltiples fuentes y tipos de datos (con evidencia tanto de la actividad como de la evaluación de los alumnos), y los conectaremos tanto con los objetivos de aprendizaje como con los criterios

de evaluación resultantes de aplicar metodologías ágiles al diseño de la asignatura de *Sistemas de Bases de Datos*.

3.4. Fundamentos técnicos

Como ya hemos mencionado, el presente proyecto tiene como objetivo el desarrollo de un sistema para la gestión automatizada de datos educativos, algo que abordaremos mediante procesos ETL y su posterior visualización a través de una interfaz web. Este sistema puede implementarse mediante múltiples combinaciones tecnológicas, cuyas principales alternativas disponibles se presentarán a continuación, con la finalidad de realizar una comparativa que permita elegir la mejor de las opciones a la hora de realizar la implementación del proyecto.

3.4.1. ETL

Como hemos mencionado, el proceso de extracción, transformación y carga de datos, comúnmente conocido como ETL por sus siglas en inglés (Extract, Transform, Load), es el uno de los objetivos principales de este TFG. Esta es una técnica ampliamente utilizada en sistemas de procesamiento de datos, que permite integrar datos desde múltiples fuentes, estandarizarlos según criterios definidos y almacenarlos de forma eficiente en un sistema centralizado, como una base de datos.

Como bien indica el nombre, consta de tres fases diferenciadas. En la fase de extracción se recuperan los datos desde su origen, que puede ser un archivo, una hoja de cálculo, una API o una base de datos externa, entre otros. Posteriormente, durante la transformación, estos datos se limpian, normalizan y adaptan al formato requerido por el sistema destino, aplicando validaciones y reglas de negocio. Finalmente, en la fase de carga, los datos transformados se insertan en la base de datos de destino, listos para ser utilizados por aplicaciones posteriores o para su análisis.

De esta forma no solo se consigue automatizar el manejo de datos repetitivos y propensos a errores humanos, sino que también garantiza la coherencia y la trazabilidad en entornos donde se requiere fiabilidad y mantenimiento continuo.

Estos sistemas se pueden implementar utilizando varios tipos de herramientas, presentadas a continuación:

3.4.1.1. Herramientas ETL especializadas

La primera de las opciones presentadas son unas herramientas que están diseñadas explícitamente para realizar ETL. Entre ellas, se han seleccionado las siguientes opciones:

- **Informatica PowerCenter:** La primera opción presentada es una solución empresarial completa que ofrece capacidades avanzadas de transformación de datos, conectividad con múltiples fuentes y gestión de metadatos. Esta herramienta es ideal para entornos corporativos en los que se manejan grandes volúmenes de datos.
- **Talend Open Studio:** Otra herramienta muy utilizada, y que además es *open-source*, es Talend Open Studio, que incorpora una interfaz gráfica para diseño de procesos ETL, lo que permite abordar el proceso de ETL sin tener un gran conocimiento de programación general. Con esta herramienta, podemos generar código *Java* e integrarlo con tecnologías que

explotan el *Big Data*, como por ejemplo *Spark*, ampliamente utilizada en muchos ámbitos tecnológicos.

- **Microsoft SSIS:** La última opción presentada es un componente de *SQL Server* utilizado para la integración de datos. Gracias a SSIS, podemos tener una integración completa con el ecosistema Microsoft, que puede ser clave si trabajamos en este entorno. Además, garantiza un aprendizaje relativamente sencillo al permitir realizar todas las acciones disponibles utilizando la interfaz visual.

3.4.1.2. Bibliotecas de programación

Si, en cambio, preferimos utilizar un lenguaje de programación tradicional y más general, tenemos varias opciones de librerías contenidas en diferentes lenguajes que permiten gestionar las ETL sin requerir un mayor conocimiento que el dominio de dicho lenguaje y la librería asociada. Entre las opciones más utilizadas hemos seleccionado las siguientes:

- **Python con Pandas:** Esta es la combinación más utilizada en los últimos años, ya que ofrece una gran sencillez para la manipulación de datos en forma de tabla, muy útil, especialmente, si trabajamos con bases de datos relacionales. La librería de *Pandas* nos ofrece estructuras como *DataFrames* o *Series*, con operaciones vectorizadas que permiten un alto rendimiento con grandes cantidades de datos.
- **R con dplyr:** Si estamos trabajando con datos estadísticos, la mejor opción es utilizar el lenguaje R, especializado en estadística pero capacidades de programar ETL a través de algunos paquetes descargables como *dplyr* o *tidyr*. Este es ideal cuando el proceso de ETL incluye transformaciones estadísticas complejas.
- **Java con Apache Camel:** Por último, daremos una opción más cercana al ámbito empresarial. Este *framework*, permite la integración de varios sistemas, incluye algunos componentes relacionados con el proceso de ETL. Es adecuado, especialmente, para entornos que requieran robustez y escalabilidad.

3.4.1.3. Soluciones en la nube

Por otro lado, si queremos desplegar nuestra aplicación en la nube y almacenar todos los datos en servidores *cloud*, podemos seleccionar alguna de las opciones presentadas a continuación. A diferencia de las otras, estas tienen un mayor coste, pero ofrecen algunas características que las anteriores no pueden proporcionar. Para estas opciones hemos decidido acudir a tres de los grandes proveedores de servicios en la nube: AWS, Google y Azure.

- **AWS Glue:** Dentro de AWS, gestionada por Amazon, Glue es un servicio totalmente auto-suficiente que permite crear y ejecutar procesos ETL. En Glue se incluyen varios catálogos de datos, además de la integración con el resto de servicios AWS, punto muy importante en caso de desarrollar nuestra aplicación al completo en entornos de Amazon.
- **Google Dataflow:** De forma casi análoga al caso de AWS, Google ofrece *Dataflow*, que permite la implementación totalmente gestionada del modelo de programación *Apache Beam*, utilizado para procesamiento por lotes y streaming, aplicables al proceso de extracción, transformación y carga de datos que estamos planteando.

- **Azure Data Factory:** Por último, estudiando las opciones que nos brinda Azure, descubrimos *Data Factory*, un servicio de integración de datos basado en la nube que permite crear flujos de trabajo orientados a datos. Con ello podríamos gestionar al completo el proceso de ETL que deseamos.

3.4.2. Alternativas para el almacenamiento de datos

Como hemos mencionado en la explicación de la ETL, el objetivo final de estas es la preparación de los datos para su futuro almacenamiento en una base de datos. Por ello, tenemos que considerar todas las opciones de almacenamiento de datos que existen, para poder posteriormente elegir la que mejor se adapte a las condiciones en las que vamos a desarrollar este proyecto.

Respecto a la elección de la base de datos, lo primero que debemos hacer es elegir qué tipo de base de datos queremos, entre dos opciones: relacional y NoSQL. Mientras las relacionales organizan la información en tablas estructuradas con relaciones definidas, garantizando consistencia mediante ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), las NoSQL adoptan modelos flexibles (documentos, clave-valor, grafos o columnas anchas) optimizados para escalabilidad horizontal y manejo de datos semiestructurados o no estructurados. La elección entre ambos modelos depende fundamentalmente de los requisitos de consistencia, escalabilidad y estructura de los datos a gestionar.

Presentaremos a continuación opciones para los dos tipos de bases de datos, comenzando por las relacionales:

- **MySQL/MariaDB:** La opción más habitual para una base de datos relacional es utilizar las basadas en MySQL (o MariaDB). Estos sistemas, además de ser *open-source*, ofrecen un buen equilibrio entre funcionalidades y rendimiento, soporte transaccional ACID y una amplia variedad de interfaces para utilizarlas.
- **PostgreSQL:** Otra opción bastante utilizada, y muy similar a la anterior es PostgreSQL. Este sistema es ligeramente más avanzado, con soporte para tipos de datos complejos y funciones almacenadas en múltiples lenguajes.
- **Microsoft SQL Server:** Por último, daremos una opción que integra perfectamente con el resto de opciones presentadas de Microsoft. SQL Server es una solución empresarial que, además de bases de datos, ofrece herramientas avanzadas de visualización para la creación de *dashboards* o resúmenes de los datos almacenados.

En cuanto a las bases de datos NoSQL, después de la investigación realizada, hemos seleccionado las siguientes opciones que son las más utilizadas actualmente:

- **MongoDB:** La primera opción permite almacenar datos en formato JSON-like (BSON). Este formato permite un esquema flexible, además de un alto rendimiento, especialmente para las operaciones de lectura.
- **Cassandra:** Si queremos almacenar nuestros datos en múltiples servidores, podemos elegir esta base de datos. Es una base de datos distribuida, diseñada y optimizada para manejar grandes cantidades de datos en múltiples servidores.

- **Redis:** La última opción presentada es Redis. Esta opción es ligeramente diferente a las anteriores, ya que es un almacén de estructuras de datos, que permite almacenar datos en memoria. Habitualmente Redis se utiliza como caché o broker de mensajes, por lo que sus usos son más específicos que las opciones presentadas anteriormente.

3.4.3. Backend

El *backend* constituye la capa fundamental del sistema, responsable de la lógica de negocio, procesamiento de datos y comunicación entre la base de datos y la interfaz de usuario. La selección de tecnologías para esta capa debe considerar factores como rendimiento, escalabilidad, facilidad de desarrollo y mantenimiento, así como la integración con otros componentes del sistema.

En el contexto de aplicaciones web modernas, existen múltiples enfoques para implementar el *backend*, que van desde *frameworks* minimalistas hasta plataformas empresariales completas. A continuación, se presentan las principales alternativas disponibles, clasificadas según los lenguajes de programación subyacentes y sus ecosistemas asociados:

- **Python con Flask/Django:** En el contexto de aplicaciones web modernas, *Python* emerge como una opción destacada mediante dos enfoques principales. Por un lado, Flask representa un *microframework* minimalista ideal para el desarrollo de APIs REST y aplicaciones de pequeña y mediana escala, destacando por su simplicidad y flexibilidad. Por otro lado, *Django* ofrece un *framework full-stack* más completo que incluye ORM integrado y sigue el patrón arquitectónico MVC, proporcionando una solución más integrada pero con mayor *overhead*. Ambos comparten el ecosistema Python, conocido por su sintaxis clara y amplia colección de paquetes especializados.
- **Node.js con Express:** Node.js con Express presenta un enfoque alternativo basado en *JavaScript*, permitiendo un desarrollo *full-stack* con un único lenguaje. Este *stack* es particularmente eficiente para aplicaciones con operaciones I/O intensivas gracias a su modelo de *event loop* no bloqueante. Su ecosistema *npm*, el mayor repositorio de paquetes existente, ofrece soluciones para prácticamente cualquier necesidad, aunque con variaciones en calidad y mantenimiento.
- **Java con Spring Boot:** Para entornos empresariales que requieren robustez y tipado fuerte, Java con Spring Boot se configura como una opción profesional. *Spring* implementa patrones como la inyección de dependencias y ofrece un soporte excepcional para arquitecturas de microservicios, aunque con una curva de aprendizaje más pronunciada y mayor verbosidad en el código. Es especialmente relevante en sistemas que deben integrarse con infraestructuras corporativas existentes.
- **PHP con Laravel:** La última opción propuesta es PHP junto con Laravel, que mantiene su relevancia en el desarrollo web tradicional, ofreciendo un equilibrio entre funcionalidad completa y accesibilidad. Laravel simplifica operaciones comunes como el manejo de rutas, autenticación e interacción con bases de datos relacionales, siendo una opción válida para equipos con experiencia en el ecosistema PHP aunque con limitaciones en rendimiento comparado con alternativas más modernas.

3.4.4. Frontend

La capa de *frontend* es la interfaz entre el sistema y los usuarios finales, determinando en gran medida la experiencia de usuario y la usabilidad de la aplicación. Las tecnologías *frontend* han evolucionado significativamente en los últimos años, pasando de simples páginas HTML estáticas a complejas aplicaciones interactivas que se ejecutan directamente en el navegador.

La elección de tecnologías frontend debe considerar aspectos como:

- Complejidad de la interfaz requerida
- Necesidad de interactividad en tiempo real
- Compatibilidad con distintos dispositivos y navegadores
- Rendimiento en el lado del cliente
- Facilidad de mantenimiento y actualización

Actualmente, los frameworks JavaScript modernos dominan el desarrollo frontend, aunque existen alternativas para casos específicos. Las principales opciones se detallan a continuación:

- **React:** React ha ganado amplia adopción como biblioteca (no *framework* completo) para construir interfaces de usuario basadas en componentes. Su implementación del *Virtual DOM* optimiza el rendimiento al minimizar las operaciones directas sobre el DOM real, mientras que su enfoque declarativo simplifica la creación de interfaces dinámicas. La arquitectura unidireccional de flujo de datos y el uso de *JSX* (mezcla de *JavaScript* y HTML) son características distintivas que ofrecen potencia a costa de cierta complejidad inicial.
- **Angular:** Angular representa un *framework* completo que incluye todas las herramientas necesarias para el desarrollo *frontend* empresarial. Su sistema de *two-way data binding* sincroniza automáticamente la vista con el modelo, aunque con posibles impactos en rendimiento para aplicaciones muy complejas. La inyección de dependencias y el fuerte tipado con *TypeScript* lo hacen especialmente adecuado para equipos grandes que valoran la consistencia y la capacidad de *testing*, aunque su curva de aprendizaje es considerablemente más empinada que alternativas más ligeras.
- **Vue.js:** Vue.js se posiciona como una opción intermedia, combinando aproximaciones de React y Angular pero con una filosofía más progresiva. Permite ser adoptado gradualmente, desde pequeñas mejoras en páginas existentes hasta aplicaciones complejas de una sola página (SPA). Su sintaxis clara y documentación excelente lo hacen accesible para desarrolladores noveles, mientras que características como el sistema reactivo y la composición de componentes satisfacen necesidades avanzadas.

Aunque los *frameworks* modernos ofrecen potentes funcionalidades y patrones de desarrollo estructurados, también introducen una complejidad significativa en la configuración, curva de aprendizaje y tamaño final de la aplicación. Para proyectos con requisitos simples o cuando se prioriza la simplicidad y el rendimiento ligero, sigue siendo válido el enfoque tradicional usando únicamente *HTML* para la estructura, *CSS* para el diseño y *JavaScript* para la interactividad básica. Esta combinación clásica, aunque carece de las abstracciones avanzadas de los *frameworks*, permite un control total sobre el código y evita las dependencias externas, siendo suficiente para interfaces con requerimientos básicos de interactividad.

Parte II

Desarrollo de la solución

Capítulo 4

Análisis

En este capítulo presentaremos la etapa de análisis del software, basada en identificar, definir y documentar los requisitos del sistema. El objetivo de este análisis es conseguir identificar las necesidades de los usuarios, para posteriormente traducir estas en requisitos. Una vez identifiquemos los requisitos, se podrá empezar a diseñar y desarrollar el sistema a partir de los mismos, por lo que son importantes, al ser la base del software que se va a construir.

La definición de estos requisitos será un proceso iterativo durante el proyecto, necesario para poder ajustar los posibles cambios que surjan durante el desarrollo del mismo. Presentaremos a continuación la especificación y modelado de los diferentes tipos de requisitos utilizados en este trabajo.

4.1. Actores

Los actores son los usuarios que podrán utilizar la aplicación que vamos a desarrollar. Para este proyecto, tendremos únicamente un actor, el docente, que será el que podrá utilizar la aplicación, ya sea cargando y procesando datos o visualizando los resúmenes de la información disponible. En la tabla 4.1 se puede consultar la información detallada de este actor:

ID	Actor	Descripción
A-01	Profesor	Será el profesor de cualquier asignatura que siga la metodología <i>Test-Driven Learning</i> , que podrá utilizar la aplicación para simplificar el procesamiento de los datos que se generen en dicha asignatura. Además, podrá consultar información obtenida directamente de la base de datos donde se almacena, como por ejemplo resúmenes personalizados con estadísticas del desempeño de alumnos en particular o de la clase en general.

Tabla 4.1: Actores.

4.2. Requisitos de usuario

En este apartado se recogen los requisitos de usuario del sistema, entendidos como las funcionalidades clave que debe ofrecer desde la perspectiva del usuario final. Para facilitar su análisis

y representación, estos requisitos se modelarán mediante casos de uso, una técnica ampliamente utilizada en ingeniería del software que permite describir de forma estructurada las interacciones entre los usuarios y el sistema.

En la imagen 4.1 se muestra el diagrama de casos de uso del sistema que se va a construir. El profesor podrá interactuar con la aplicación con varias finalidades. La primera a desatacar es la gestión de la asignatura en general, permitiéndole la creación de un curso, gestión del producto de aprendizaje, de las acciones o del método de evaluación seguido. Además de esto, puede ejecutar el procesamiento de la información de la asignatura (evaluaciones, autoevaluaciones, tareas, datos de los estudiantes, formularios). Finalmente, el profesor tendrá la opción de consultar la información agregada de la base de datos, pudiendo visualizar algunos *dashboards* simples agregados por estudiante o generales de toda la clase.

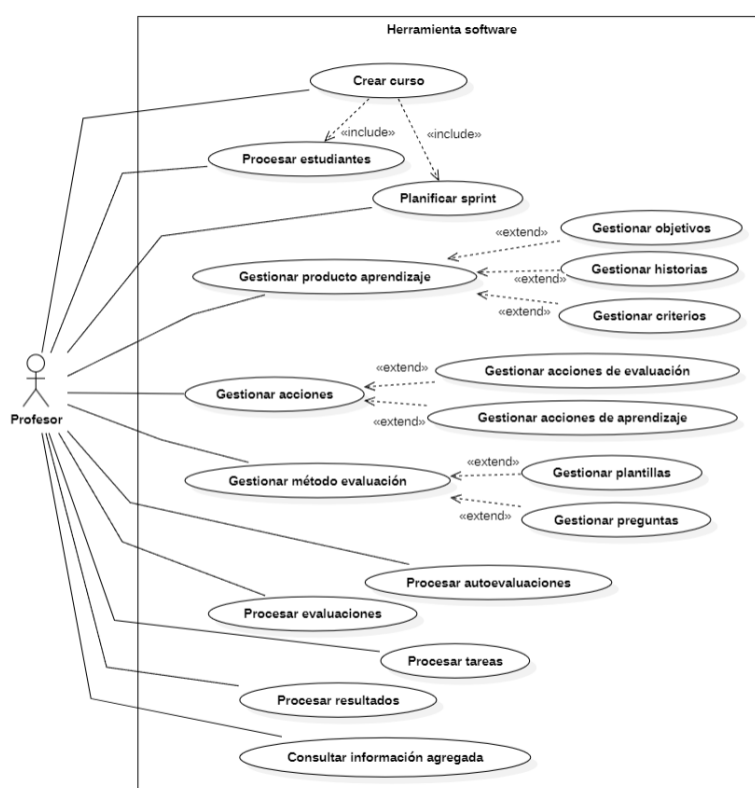


Figura 4.1: Diagrama de casos de uso.

A continuación, se especifican todos los casos de uso que se muestran en la imagen 4.1. Para cada uno de ellos se ha incluido una tabla con todos los apartados necesarios para detallar cada uno de los casos de uso identificados.

CU-01	Crear curso.	
Descripción	Describe el proceso mediante el cual el profesor puede crear un curso y los <i>sprints</i> asociados al mismo, para la asignatura para la que está diseñada la aplicación.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita crear un curso.
	Paso 2	El sistema solicita que se completen los años del curso y las fechas de inicio y finalización del mismo.
	Paso 3	El profesor proporciona la información requerida.
	Paso 4	El sistema solicita que se proporcione la información de cada uno de los <i>sprints</i> asociados al curso.
	Paso 5	El profesor completa la información de cada curso, añadiendo el número de <i>sprints</i> que sea necesario.
	Paso 6	Se realiza el caso de uso CU-03: Planificar <i>sprint</i> (Paso 4 en adelante).
	Paso 7	Se realiza el caso de uso CU-02: Procesar estudiantes (Paso 4 en adelante).
	Paso 8	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.13) o la fecha de finalización del curso es previa a la de inicio.
	Ex 2	El caso de uso vuelve al paso 3 si no se han introducido todos los datos obligatorios (ver Figura 4.13) o alguna fecha de finalización de un <i>sprint</i> es previa a la de inicio, o si alguna de las fechas de inicio o finalización es externa al rango comprendido entre las fechas indicadas previamente de inicio y fin de curso.

Tabla 4.2: Especificación del caso de uso CU-01: Crear curso.

CU-02	Procesar estudiantes.	
Descripción	Describe el proceso mediante el cual el profesor puede procesar nuevos estudiantes que se hayan matriculado posterior al inicio del curso de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita procesar nuevos estudiantes.
	Paso 2	El sistema requiere que se seleccione un curso de los disponibles.
	Paso 3	El profesor selecciona uno de los cursos.
	Paso 4	El sistema solicita el listado de estudiantes matriculados en el curso.
	Paso 5	El profesor proporciona los datos de los estudiantes.
	Paso 6	El sistema procesa los datos e informa de los resultados del proceso.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 4 si no se proporciona un archivo que tenga el formato y nombre válidos.

Tabla 4.3: Especificación del caso de uso CU-02: Procesar estudiantes.

CU-03	Planificar <i>sprint</i>.	
Descripción	Describe el proceso mediante el cual el profesor tiene la opción de realizar la planificación de un <i>sprint</i> posterior a la creación de un curso en la asignatura considerada.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita planificar un <i>sprint</i> .
	Paso 2	El sistema requiere que se seleccione un curso y posteriormente un <i>sprint</i> de los disponibles.
	Paso 3	El profesor elegirá un curso, y después un <i>sprint</i> de los disponibles.
	Paso 4	El sistema mostrará los objetivos disponibles para ese <i>sprint</i> y las historias seleccionadas en la planificación en caso de que exista alguna.
	Paso 5	El profesor elige las historias que se planifican para dicho <i>sprint</i> .
	Paso 6	El sistema procesará los datos introducidos informando de los problemas en caso de encontrar alguno.
	Paso 7	El caso de uso finaliza satisfactoriamente.

Tabla 4.4: Especificación del caso de uso CU-03: Planificar *sprint* curso.

CU-04	Gestionar producto aprendizaje.	
Descripción	Describe el proceso mediante el cual el profesor tiene la opción de gestionar todos los elementos relacionados con el producto de aprendizaje, entre los que están los objetivos, las historias y los criterios.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita gestionar alguno de los elementos del producto de aprendizaje.
	Paso 2	el sistema facilita la posibilidad de gestionar objetivos, historias y criterios de aceptación.
	Paso 3	a. Si el usuario solicita gestionar objetivos, se realiza el caso de uso CU-05 Gestionar objetivos. b. Si el usuario solicita gestionar historias, se realiza el caso de uso CU-06 Gestionar historias. c. Si el usuario solicita gestionar criterios, se realiza el caso de uso CU-07 Gestionar criterios.
	Paso 4	El caso de uso finaliza satisfactoriamente.

Tabla 4.5: Especificación del caso de uso CU-04: Gestionar producto aprendizaje.

CU-05	Gestionar objetivos.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de los objetivos del producto de aprendizaje de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de los objetivos.
	Paso 2	El sistema muestra toda la información de los objetivos.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figura 4.17).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la acción elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.17) o no cumplen alguna de las restricciones indicadas.

Tabla 4.6: Especificación del caso de uso CU-05: Gestionar objetivos.

CU-06	Gestionar historias.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de las historias del producto de aprendizaje de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de las historias.
	Paso 2	El sistema muestra toda la información de las historias.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figura 4.16).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la acción elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.16) o no cumplen alguna de las restricciones indicadas.

Tabla 4.7: Especificación del caso de uso CU-06: Gestionar historias.

CU-07	Gestionar criterios.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de los criterios del producto de aprendizaje de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de los criterios.
	Paso 2	El sistema muestra toda la información de los criterios.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figura 4.12).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la acción elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.12) o no cumplen alguna de las restricciones indicadas.

Tabla 4.8: Especificación del caso de uso CU-07: Gestionar criterios.

CU-08	Gestionar acciones.	
Descripción	Describe el proceso mediante el cual el profesor tiene la opción de gestionar algunos de los elementos relacionados con el proceso de aprendizaje, en concreto las acciones (de aprendizaje y evaluación).	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita gestionar alguno de los tipos de acciones disponibles.
	Paso 2	Se realizan los casos de uso CU-09: Gestionar acciones de evaluación y CU-10: Gestionar acciones de aprendizaje.
	Paso 3	El caso de uso finaliza satisfactoriamente.

Tabla 4.9: Especificación del caso de uso CU-08: Gestionar acciones.

CU-09	Gestionar acciones de evaluación.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de las acciones de evaluación de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de las acciones de evaluación.
	Paso 2	El sistema muestra toda la información de las acciones de evaluación.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figuras 4.10 y 4.15).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la petición elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figuras 4.10 y 4.15) o no cumplen alguna de las restricciones indicadas.

Tabla 4.10: Especificación del caso de uso CU-09: Gestionar acciones de evaluación.

CU-10	Gestionar acciones de aprendizaje.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de las acciones de aprendizaje de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de las acciones de aprendizaje.
	Paso 2	El sistema muestra toda la información de las acciones de aprendizaje.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figuras 4.10 y 4.11).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la petición elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figuras 4.10 y 4.11) o no cumplen alguna de las restricciones indicadas.

Tabla 4.11: Especificación del caso de uso CU-10: Gestionar acciones de aprendizaje.

CU-11	Gestionar método evaluación.	
Descripción	Describe el proceso mediante el cual el profesor tiene la opción de gestionar todos los elementos relacionados con el método de evaluación de la asignatura, es decir, las preguntas que van a aparecer en las acciones de evaluación y las plantillas que siguen dichas preguntas.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita gestionar las plantillas o las preguntas.
	Paso 2	Se realizan los casos de uso CU-12: Gestionar plantillas y CU-13: Gestionar preguntas.
	Paso 3	El caso de uso finaliza satisfactoriamente.

Tabla 4.12: Especificación del caso de uso CU-11: Gestionar método evaluación.

CU-12	Gestionar plantillas.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de las plantillas de evaluación disponibles, junto a los criterios que estas evalúan.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de las plantillas.
	Paso 2	El sistema muestra toda la información de las plantillas.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figura 4.18).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la acción elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.18) o no cumplen alguna de las restricciones indicadas.

Tabla 4.13: Especificación del caso de uso CU-11: Gestionar plantillas.

CU-13	Gestionar preguntas.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar, editar, eliminar o crear (CRUD) todos los datos de las preguntas de evaluación disponibles, además de la plantilla a la que están asociadas.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar los datos de las preguntas.
	Paso 2	El sistema muestra toda la información de las preguntas.
	Paso 3	El profesor selecciona la acción que quiere realizar.
	Paso 4	El sistema muestra la información a completar (ver Figura 4.19).
	Paso 5	El profesor completa la información solicitada.
	Paso 6	El sistema completa la acción elegida en caso de que sea posible, informando del éxito o error.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 2 si no se han introducido todos los datos obligatorios (ver Figura 4.19) o no cumplen alguna de las restricciones indicadas.

Tabla 4.14: Especificación del caso de uso CU-13: Gestionar preguntas.

CU-14	Procesar autoevaluaciones.	
Descripción	Describe el proceso mediante el cual el profesor puede procesar las autoevaluaciones de los estudiantes asociadas a alguna de las acciones de evaluación realizadas durante la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita procesar las autoevaluaciones.
	Paso 2	El sistema requiere la información del curso, <i>sprint</i> y acción.
	Paso 3	El profesor indica un curso, <i>sprint</i> y acción.
	Paso 4	El sistema solicita que se carguen las autoevaluaciones de los estudiantes.
	Paso 5	El profesor proporciona las autoevaluaciones.
	Paso 6	El sistema procesa los datos informando de los errores.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 4 si no se respeta el formato de las autoevaluaciones.

Tabla 4.15: Especificación del caso de uso CU-14: Procesar autoevaluaciones.

CU-15	Procesar evaluaciones.	
Descripción	Describe el proceso mediante el cual el profesor puede procesar sus evaluaciones sobre el desempeño de los alumnos en alguna de las acciones de evaluación realizadas durante la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita procesar las evaluaciones.
	Paso 2	El sistema requiere la información del curso y el <i>sprint</i> .
	Paso 3	El profesor determina un curso y <i>sprint</i> .
	Paso 4	El sistema solicita la información de la evaluación.
	Paso 5	El profesor proporciona la evaluación.
	Paso 6	El sistema procesa los datos informando de los errores.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 4 si no se proporcionan evaluaciones válidas.

Tabla 4.16: Especificación del caso de uso CU-15: Procesar evaluaciones.

CU-16	Procesar tareas.	
Descripción	Describe el proceso mediante el cual el profesor puede procesar las tareas que los alumnos han ido registrando en sus cuadernos de trabajo a lo largo de la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita procesar las tareas.
	Paso 2	El sistema requiere la información del curso y el <i>sprint</i> .
	Paso 3	El profesor determina un curso y <i>sprint</i> .
	Paso 4	El sistema solicita la información de las tareas.
	Paso 5	El profesor proporciona las tareas.
	Paso 6	El sistema procesa los datos informando de los errores.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 4 si no se proporcionan tareas válidas.

Tabla 4.17: Especificación del caso de uso CU-16: Procesar tareas.

CU-17	Procesar resultados.	
Descripción	Describe el proceso mediante el cual el profesor puede procesar los resultados (rutinas de pensamiento) de los estudiantes asociados a alguna de las acciones de evaluación realizadas durante la asignatura.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita procesar los resultados.
	Paso 2	El sistema requiere la información del curso y el <i>sprint</i> .
	Paso 3	El profesor determina un curso y <i>sprint</i> .
	Paso 4	El sistema solicita la información de los resultados.
	Paso 5	El profesor proporciona la evaluación.
	Paso 6	El sistema procesa los datos informando de los errores.
	Paso 7	El caso de uso finaliza satisfactoriamente.
Excepciones	Ex 1	El caso de uso vuelve al paso 4 si no se proporcionan los resultados válidas.

Tabla 4.18: Especificación del caso de uso CU-17: Procesar resultados.

CU-18	Consultar información agregada.	
Descripción	Describe el proceso mediante el cual el profesor puede consultar resúmenes agregados de algunos resultados obtenidos por alumnos en particular o por toda la clase en una historia que abarque varias acciones de evaluación.	
Autor	Profesor.	
Actor	Analista de datos.	
Secuencia normal	Paso 1	El profesor solicita consultar la información agregada.
	Paso 2	El sistema pide completar el curso, <i>sprint</i> y tipo de agregación.
	Paso 3	El profesor selecciona curso y la historia entre los disponibles, así como el tipo de agregación.
	Paso 4	El sistema muestra la información agregada para la selección realizada.
	Paso 5	El caso de uso finaliza satisfactoriamente.

Tabla 4.19: Especificación del caso de uso CU-18: Consultar información agregada.

4.3. Requisitos funcionales

En esta sección se detallan los requisitos funcionales del sistema, es decir, aquellas funciones específicas que debe ser capaz de realizar para cumplir con los objetivos definidos. Estos requisitos describen el comportamiento esperado del sistema ante distintas entradas o acciones del usuario, definiendo lo que este debe hacer para cumplir las necesidades de quien utilice la aplicación.

Detallaremos los requisitos funcionales en la tabla 4.20, donde cada uno de ellos tiene un identificador único para asegurar su trazabilidad a lo largo del proyecto.

ID	Requisitos funcionales
RF-01	El profesor podrá consultar, editar y eliminar toda la información almacenada en la base de datos.
RF-02	El profesor añadir nueva información a la base de datos, mediante el procesamiento automático o la gestión directamente desde el sistema.
RF-03	El sistema realizará las transformaciones necesarias para que los datos cargados en la base de datos tengan la forma adecuada definida en el apartado 4.6.1.
RF-04	El sistema mantendrá un histórico de los ficheros procesados, almacenándolos en un sistemas de carpetas preparado para ello y generado automáticamente.
RF-05	El sistema consultará y generará los datos necesarios para que puedan ser mostrados en la visualización en <i>dashboards</i> .

Tabla 4.20: Requisitos funcionales.

4.4. Requisitos no funcionales

Además de las funcionalidades que el sistema debe ofrecer, es importante tener en cuenta una serie de características que afectan a cómo se comporta y se percibe dicho sistema. Los requisitos no funcionales abarcan estos aspectos, como la eficiencia, la escalabilidad, la accesibilidad o la seguridad, y ayudan a definir la calidad general del producto final. Aunque no describen acciones concretas, son esenciales para garantizar una experiencia satisfactoria y un correcto funcionamiento en distintas condiciones de uso.

Al igual que para los requisitos no funcionales, hemos construido la Tabla 4.21 para especificar los requisitos no funcionales y asegurar su trazabilidad.

ID	Requisitos no funcionales
RNF-01	El sistema solo admitirá archivos <i>ZIP</i> para los comprimidos.
RNF-02	El sistema solo admitirá archivos <i>XLSX</i> para las hojas de cálculo.

Tabla 4.21: Requisitos no funcionales.

4.5. Requisitos de interfaz externa

Para que la aplicación funcione correctamente, debemos definir una serie de requisitos de como deben ser las estructuras de los ficheros que son necesarios para completar el procesamiento automático de la información. Para ello, construimos la Tabla 4.22 que, junto a las imágenes detalladas, completan el detalle que se debe conocer para un uso adecuado del sistema que se va a construir

ID	Requisitos de interfaz externa
RIE-01	El archivo comprimido de las autoevaluaciones debe contener las carpetas generadas al descargar los archivos directamente de una entrega de Moodle. Esta estructura contendrá una carpeta por cada alumno y dentro de ella cada uno de los archivos que haya entregado dicho alumno. Se puede consultar esta estructura en las Figuras 4.2 y 4.2.
RIE-02	El archivo comprimido de las tareas debe contener las carpetas generadas al descargar los archivos directamente de una entrega de Moodle. La estructura será la misma que en el caso de RIE-01, detallada en la Figura 4.2, con la diferencia de que contendrá los archivos correspondientes a las tareas.
RIE-03	La hoja de cálculo que contiene la información de los resultados debe ser la generada automáticamente al descargar los resultados de la rutina de pensamiento de Microsoft Forms. El formato que debe tener esta hoja se muestra en la Figura 4.6.
RIE-04	La hoja de cálculo que contiene la información de las evaluaciones debe tener como columnas todos los criterios evaluados en dicha acción y por filas los resultados obtenidos por los estudiantes en dichos criterios. Se puede consultar con detalle la estructura de este fichero en la imagen 4.7.
RIE-05	La hoja de cálculo que contiene la información de los estudiantes matriculados debe tener la estructura de la hoja que se puede descargar automáticamente desde el “sigma” de la Universidad. Se puede consultar con detalle la estructura de este fichero en la imagen 4.8.

Tabla 4.22: Requisitos de interfaz externa.

En las siguientes imágenes encontramos los detalles necesarios para entender los requisitos mostrados anteriormente.

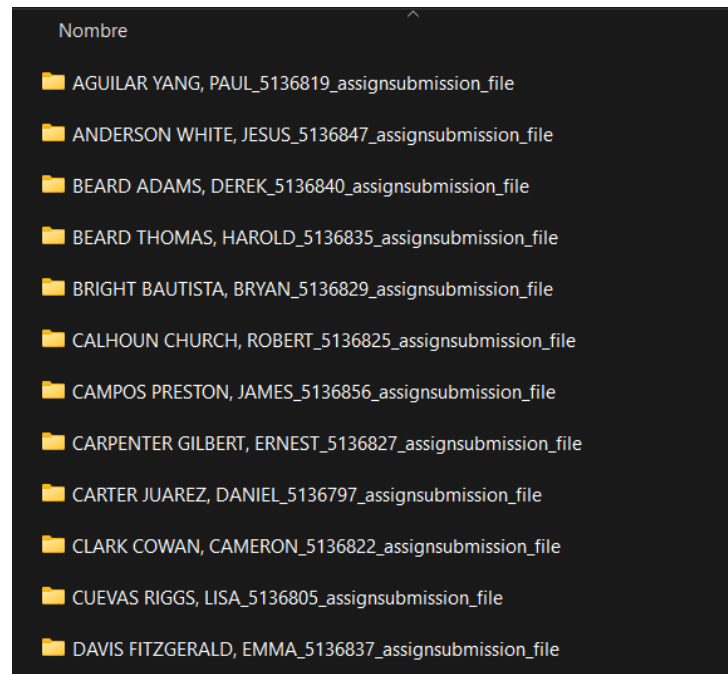


Figura 4.2: Estructura de carpetas descargada de *Moodle*.

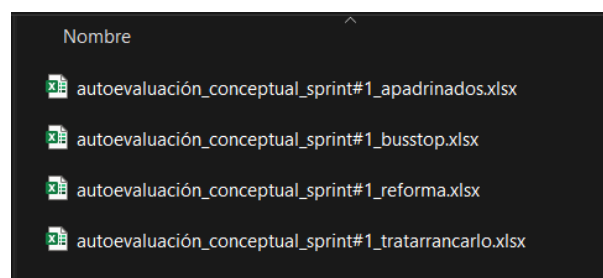


Figura 4.3: Archivos contenidos en una carpeta de una entrega de *Moodle*.

	A	B	C	D	E	F
1						
7		Apellidos, Nombre				
8						
9		Historia de aprendizaje	Criterios de aceptación			Evaluación
10		Modelo ER	CONCEPTUAL-2.2.	Soy capaz de caracterizar las entidades presentes en el "mini-mundo" de una base de datos.	Entidades	0,50
11			CONCEPTUAL-2.3.	Soy capaz de decidir cuándo una entidad debe modelarse como débil, de acuerdo con su dependencia existencial de una entidad fuerte.		
12			CONCEPTUAL-2.4.	Soy capaz de caracterizar los atributos que describen cualquier entidad (o relación) presente en el "mini-mundo" de una base de datos.	Atributos	0,63
13			CONCEPTUAL-2.5.	Soy capaz de elegir el identificador de una entidad, a partir de sus atributos descriptivos.	Identificadores	1,00
14			CONCEPTUAL-2.6.	Soy capaz de caracterizar las relaciones existentes entre cualquier entidad presente en el "mini-mundo" de una base de datos.	Relaciones	0,33
15			CONCEPTUAL-2.7.	Soy capaz de determinar la participación y la cardinalidades propias de relaciones de cualquier grado.	Multiplicidades	0,50
16			CONCEPTUAL-2.8.	Soy capaz de construir el modelo entidad-relación que establece el diseño conceptual de una base de datos.	Modelo	1,00
17			CONCEPTUAL-2.9.	Soy capaz de utilizar una notación estandarizada para expresar el diseño conceptual (y mantener un uso consistente de la misma durante todo el ciclo de vida).	Nombres	0,50
18		Objetivo	Conceptual	5,29		
19		Historia	ER			

Figura 4.4: Estructura de la hoja de cálculo de autoevaluaciones.

4.5. Requisitos de interfaz externa

	A	B	C	D	E	F	G
1							
2		Estudiante		Asignatura		Sistemas de Bases de Datos	
3							
4		Fecha	Tipo de actividad	Tarea realizada	Tiempo (minutos)	Comentario	
5		11/09/2024	Asistencia a clases	Teoría	120	Presentación de la asignatura	
6		12/09/2024	Asistencia a clases	Teoría	120		
7		18/09/2024	Asistencia a clases	Apoyo	120	Presentación playrock	
8		18/09/2024	Asistencia a clases	Teoría	120		
9		23/09/2024	Trabajo individual	Realización de tarea	60	empiezo ejercrcios modelo E-R	
10		23/09/2024	Trabajo en equipo	Reunión	45	Reunión de iniciación	
11		24/09/2024	Trabajo individual	Realización de tarea	90	acabo ejercicios modelo E-R	
12		25/09/2024	Asistencia a clases	Apoyo	120	Ej tipo examen y corrección del mismo	
13		25/09/2024	Trabajo individual	Hall of Fame	16	Entrenador personal (diseño conceptual)	
14		25/09/2024	Asistencia a clases	Teoría	120	Diseño lógico	
15		26/09/2024	Trabajo en equipo	Reunión	12	Reunión de sincronización#1	
16		26/09/2024	Asistencia a clases	Teoría	120	Diseño lógico	
17		26/09/2024	Trabajo individual	Realización de tarea	120	Hago la autoevaluación	
18		01/10/2024	Trabajo individual	Realización de tarea	120	Diseño lógico	
19		02/10/2024	Trabajo individual	Realización de tarea	60	Autoevaluación y formulario	
20		02/10/2024	Asistencia a clases	Laboratorio	120	Inicio de SQL	
21		03/10/2024	Asistencia a clases	Laboratorio	120		
22		01/10/2024	Trabajo individual	Otros	50	Entidades de PR	
23		09/10/2024	Asistencia a clases	Examen	105	Examen sprint 1	
24		09/10/2024	Trabajo individual	Otros	100	Termino las entidades de PR	
25		10/10/2024	Trabajo individual	Realización de tarea	120	Base de datos en SQL	
26		15/10/2024	Trabajo individual	Realización de tarea	60	Autoevaluación y formulario del examen	
27		16/10/2024	Asistencia a clases	Apoyo	120	presentación del sprint 2 de playRock	
28		16/10/2024	Asistencia a clases	Teoría	120	Comenzamos con el DML	
29		17/10/2024	Asistencia a clases	Teoría	120	DML	

Figura 4.5: Estructura de la hoja de cálculo de tareas.

	A	B	C	D	E	F	G	H
1	ID	Hora de inicio	Hora de finalización	Correo electrónico	Nombre	¿En qué se parecen t	¿En qué se diferencian	¿Qué conclusiones o
2	1	9/25/24 21:29:33	9/25/24 21:39:24		CUEVAS RIGGS, LISA	Depende del modelo los En los nombres de las er Que lo que mas me cues		
3	2	9/25/24 22:13:36	9/25/24 22:18:49		RAMIREZ PHILLIPS, ERIC	No en mucho, algunos n Muy diferentes, sobre tc Tengo que practicar más		
4	3	9/26/24 9:26:24	9/26/24 9:44:50		PHILLIPS JAMES, AVA	En primer lugar destacar He detectado que en alg La principal conclusión q		
5	4	9/26/24 10:37:20	9/26/24 11:22:07		OWENS MYERS, MICHAELA	En el modelo que he pro En cuanto a diferencias Mis sensación es que ter		
6	5	9/26/24 10:33:52	9/26/24 11:38:42		HENSON RAYMOND, DIANA	Mis 4 modelos se parec Se diferencian en varias Siento que el trabajo qui		
7	6	9/26/24 12:29:11	9/26/24 12:30:40		WILSON HARRISON, ROBERT	En bastates cosas, he he He tenido problemas co Tengo que prestar mas e		
8	7	9/26/24 12:18:37	9/26/24 12:43:02		PEREZ THOMAS, MIA	Para empezar, el proble En los modelos entidad Las conclusiones que sai		
9	8	9/26/24 14:05:14	9/26/24 14:55:18		FIELDS REYNOLDS, JERRY	Debo mencionar que en Como antes citaba los a Mis conclusiones son qu		
10	9	9/26/24 18:31:31	9/26/24 19:03:28		HARRIS CONNER, ERIC	Por lo general, a excepci En busstop, diferimos en He conseguido clarificar		
11	10	9/26/24 18:56:33	9/26/24 19:05:18		DAVIS FITZGERALD, EMMA	En general identifico cor En general me cuesta lo i Creo que tengo que prac		
12	11	9/26/24 21:36:09	9/26/24 22:01:14		HUNT GUTIERREZ, ALAN	En primer lugar quiero d En los modelos la mayor Mis sensaciones son pos		
13	12	9/26/24 22:17:07	9/26/24 22:17:14		CLARK COWAN, CAMERON	Para empezar, obviando Empecemos primero coi Empezando con el probl		
14	13	9/26/24 22:11:46	9/27/24 0:14:15		NIELSEN SANTOS, RODNEY	En primer lugar mis mod En primer lugar creo que Mis sensaciones son alg		
15	14	9/27/24 0:43:20	9/27/24 1:30:40		LANDRY COLON, HARRY	Estoy de acuerdo con la En errores que he comet Que puedo dar más y co		
16	15	9/27/24 2:37:44	9/27/24 4:32:38		WOODS GONZALEZ, JOSEPH	Para empezar, los result: Al realizar la autoevalua Creo que todavia necesi		
17	16	9/27/24 10:32:00	9/27/24 11:01:53		WATSON TAYLOR, GRETA	Dependiendo del model Como mencioné antes, i La verdad es que estoy r		
18	17	9/27/24 11:31:07	9/27/24 11:44:46		WHITEHEAD SKINNER, ERIC	En primer lugar quisiera Como mencionaba ante Como conclusiones mas		
19	18	9/27/24 11:53:23	9/27/24 12:26:45		WILLIAMS TREVINO, GREGORY	En relación a los model En relación a las entidad En conclusión, creo que		
20	19	9/27/24 12:36:40	9/27/24 12:38:51		VALDEZ HUDSON, CHARLES	Se parecen en ciertas En Se diferencian en la may He de profundizar en la i		
21	20	9/27/24 16:19:45	9/27/24 16:55:53		HUNT MORALES, ADDISON	Primero quiero destacar En mis modelos he obse He podido sacar que he		
22	21	9/27/24 16:20:00	9/27/24 17:21:28		SCHAEFER SANCHEZ, EMILY	Para empezar, tengo co Como decía, mis model Considero que mi estad		
23	22	9/27/24 16:46:08	9/27/24 17:41:48		GILL BARRETT, JORDAN	He conseguido identifi Lo que más cabe destac: Tengo que intentar simpl		
24	23	9/27/24 19:37:34	9/27/24 19:40:02		NUNEZ MARTIN, RONALD	En pocas cosas. En princ En la manera en la que p Tengo que estudiar mas		
25	24	9/27/24 18:25:03	9/27/24 19:55:00		WALKER SANCHEZ, DAVID	Pues sinceramente en "f En "Apadrinad@s" la ide Pues sobre todo he pecc		

Figura 4.6: Estructura de la hoja de cálculo de resultados.

	A	B	C	X	Y	Z	AA	AB	AC	AD	AE	AF
1												
2		Sprint #1		CONCEPTUAL-2.1	CONCEPTUAL-2.2	CONCEPTUAL-2.3	CONCEPTUAL-2.4	CONCEPTUAL-2.5	CONCEPTUAL-2.6	CONCEPTUAL-2.7	CONCEPTUAL-2.8	CONCEPTUAL-2.9
3		SIMMONS CUEV/ SCOTT		1,00	0,80	0,80	0,93	1,00	0,57	0,86	1,00	1,00
4		GILL BARRETT JORDAN		1,00	1,00	1,00	0,93	1,00	1,00	0,89	1,00	1,00
5		OCONNOR ROSS STEPHEN		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
6		LONG BLANCHAF CHRISTOPHER		0,00	0,20	0,20	0,61	0,40	0,00	0,18	1,00	0,67
7		HENSON RAYMO DIANA		1,00	0,60	0,60	0,64	0,80	0,57	0,68	1,00	0,75
8		CUEVAS RIGGS LISA		0,00	0,40	0,40	0,54	0,40	0,43	0,75	1,00	0,92
9		GLASS JOHNSON MARIAN		1,00	0,60	0,60	0,86	0,60	0,86	0,86	1,00	1,00
10		WALKER SANCHE DAVID		0,00	0,60	0,60	0,61	0,60	0,57	0,75	1,00	0,00
11		WOODWARD DA MARTHA		1,00	0,80	0,80	0,64	0,60	0,86	0,61	1,00	1,00
12		GARCIA DUDLEY CRISTIAN		1,00	1,00	1,00	0,82	0,80	1,00	0,86	1,00	1,00
13		DAVIS FITZGERA EMMA		1,00	1,00	1,00	0,79	1,00	0,86	0,93	1,00	0,92
14		WILLIS GONZALE OLIVIA		1,00	0,80	0,80	0,64	1,00	0,57	0,79	1,00	0,92
15		SCHAEFER SANCI EMILY		1,00	1,00	1,00	0,93	0,80	1,00	1,00	1,00	1,00
16		OWENS MYERS MICHAELA		0,00	0,40	0,40	0,68	0,60	0,14	0,93	1,00	1,00
17		BEARD ADAMS DEREK		0,00	1,00	1,00	0,61	0,20	0,57	0,00	1,00	0,75
18		MACIAS CARROL WILLIAM		1,00	0,60	0,60	0,54	0,60	0,57	0,64	1,00	1,00
19		ANDERSON WHI JESUS		0,00	0,60	0,60	0,68	0,60	0,14	0,54	1,00	0,58
20		CLARK COWAN CAMERON		1,00	1,00	1,00	0,86	0,80	1,00	0,89	1,00	1,00
21		NIELSEN SANTOS RODNEY		1,00	0,80	0,80	0,50	0,80	0,57	0,79	1,00	0,92
22		WAGNER GIBSOI SOPHIA		1,00	0,80	0,80	0,93	0,80	0,86	0,75	1,00	0,92
23		WILLIAMS TREVI GREGORY		1,00	0,60	0,60	0,75	0,80	0,57	0,61	1,00	1,00
24		NUNEZ CONNER EVAN		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
25		MOORE ROSS JOE		1,00	1,00	1,00	0,82	0,80	0,86	0,79	1,00	1,00
26		WHITEHEAD SKII ERIC		0,00	0,60	0,60	0,54	0,60	0,43	0,64	1,00	0,92

Figura 4.7: Estructura de la hoja de cálculo de evaluaciones.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2		LISTA DE CLASE																	
3																			
4																			
5		Datos de la Asignatura																	
6																			
7		Año académico	2024/2025-0																
8		Centro	229 - Escuela de Ingeniería Informática de Segovia																
9		Asignatura	40811 - SISTEMAS DE BASES DE DATOS																
10		Grupo	1																
11		Tipo de docencia	Teoría																
12		Créditos	6.0																
13		Periodo	Primer Cuatrimestre																
14																			
15		Profesores responsables																	
16																			
17		BREGON BREGON, ANIBAL																	
18		MARTÍNEZ PRIETO, MIGUEL ÁNGEL																	
19		MIELGO MARTIN, PAULA																	
20																			
21																			
22		Apellidos y nombre					D.N.I.		N.I.A.		Dirección electrónica								
23		SIMMONS CUEVAS, SCOTT									desmatriculado@estudiantes.uva.es								
24		GILL BARRETT, JORDAN							652462		anonymous@estudiantes.uva.es								
25		OCONNOR ROSS, STEPHEN							346345		anonymous@estudiantes.uva.es								
26		LONG BLANCHARD, CHRISTOPHER							562562		anonymous@estudiantes.uva.es								
27		HENSON RAYMOND, DIANA							875645		anonymous@estudiantes.uva.es								
28		CUEVAS RIGGS, LISA							676267		anonymous@estudiantes.uva.es								
29		GLASS JOHNSON, MARIAN							356248		anonymous@estudiantes.uva.es								
30		WALKER SANCHEZ, DAVID							245726		anonymous@estudiantes.uva.es								
31		CAMPOS PRESTON, JAMES							478245		anonymous@estudiantes.uva.es								
32		WOODWARD DAWSON, MARTHA							678345		anonymous@estudiantes.uva.es								
33		GARCIA DUDLEY, CRISTIAN							783668		anonymous@estudiantes.uva.es								
34		DAVIS FITZGERALD, EMMA							246779		anonymous@estudiantes.uva.es								
35		WILLIAMS CARSON, BRIAN							345467		anonymous@estudiantes.uva.es								
36		WILLIS GONZALEZ, OLIVIA							345666		anonymous@estudiantes.uva.es								

Figura 4.8: Estructura de la hoja de cálculo de estudiantes.

4.6. Diseño conceptual

El diseño conceptual se compone del modelo de dominio y los diccionarios de datos de las entidades y relaciones que forman este modelo.

4.6.1. Modelo de dominio

El modelo de dominio se utiliza para “poner de acuerdo” a todos los involucrados en el diseño de la base de datos. Este es un modelo que describe de forma sencilla lo que se quiere representar para que pueda ser comprendido fácilmente por todos y no haya ambigüedades. El modelo de dominio es una técnica “top-down”, es decir, primero se identifican las entidades más generales del modelo, y luego se va entrando en detalle definiendo todos los atributos y las relaciones que hay entre ellas, buscando así representar de manera simplificada cómo personas, objetos o conceptos se relacionan entre sí. Además, este modelo se utilizará para construir posteriormente la base de datos, exponiendo cómo se organizará la información en esta.

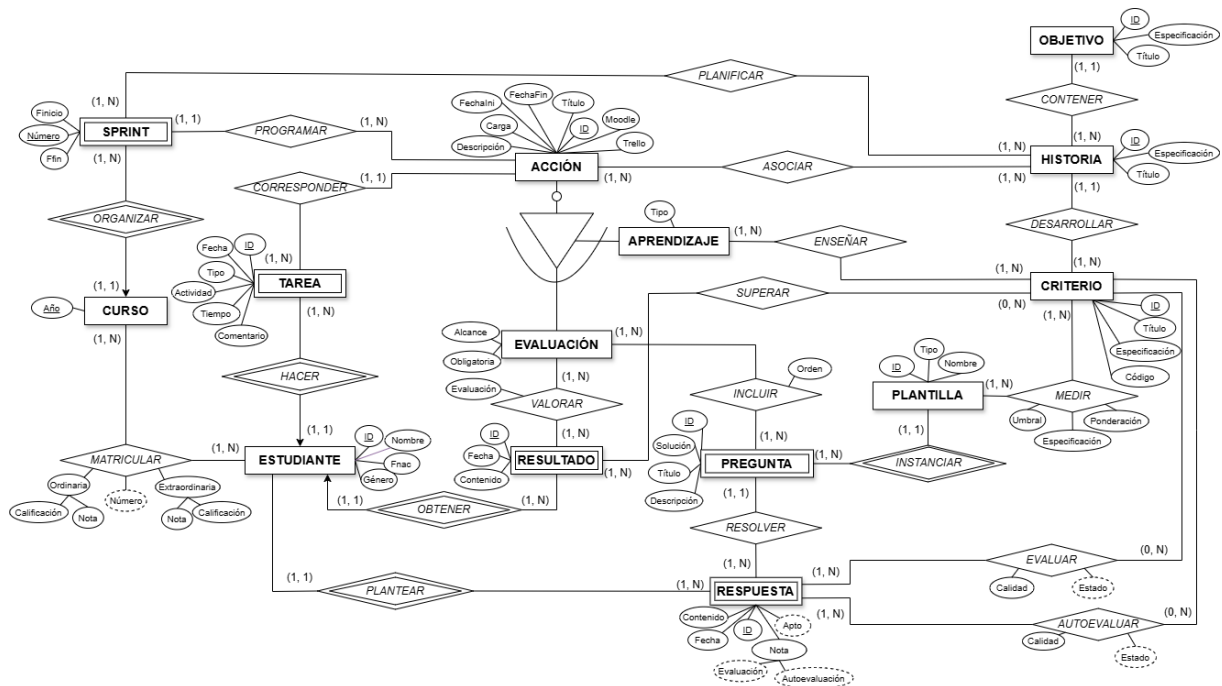


Figura 4.9: Modelo de dominio del proyecto.

La imagen 4.9 ilustra el modelo de dominio del proyecto. Para entender este modelo, vamos a describir con palabras el mini-mundo de una asignatura en la que se siga la metodología *Test-Driven Learning* (TDL), explicada previamente y que está representada en este modelo. Comenzaremos por la parte superior derecha, en la que vemos cómo se representan los objetivos, historias y criterios que definen la asignatura al completo. Como ya sabemos, en *eXtreme Learning* se definen una serie de objetivos principales, que se descomponen en historias y estas, a su vez, en criterios de aceptación más concretos. Para todos ellos se definirán aspectos como el título o la especificación, además de un identificador para distinguirlos unívocamente.

Pasando ahora a la parte izquierda del diagrama de la Figura 4.9, vemos que tenemos una

entidad CURSO y otra entidad débil dependiente de ella: SPRINT. Estas dos permiten organizar la información por cursos y *sprints*, que son las divisiones que se hacen en cada curso tal y como indica *eXtreme Learning*. Como en cada *sprint* se abordan una serie de historias predefinidas, se conectará la entidad SPRINT con HISTORIA de forma que queden definidas las historias a tratar en cada uno de estos *sprints*.

Para conectar las dos partes que hemos definido, tenemos las acciones, que encontramos en la parte media del diagrama, y que pueden ser única y obligatoriamente de dos tipos: aprendizaje y evaluación. Cada una de las acciones tiene una serie de atributos que las definen al completo, y, además, se conectan con las partes mencionadas ya que se modela que como estas acciones se desarrollan en un *sprint* concreto y se asocian con unas historias en particular. En concreto, las acciones de aprendizaje se enfocarán además en enseñar una serie de criterios definidos que los estudiantes deberían poder completar al realizar dicha acción.

Continuaremos con la parte central, enfocándonos ahora en la entidad PLANTILLA, que representa cada una de las bases que sigue el profesor para medir si los criterios de aceptación se han superado en alguna de las acciones de evaluación. Las preguntas que aparecen en dichas acciones son instancias de estas plantillas, con un enunciado y una solución asociadas, que serán con las que se deba comparar la respuesta del alumno para poder determinar si los criterios se han superado o no.

Por último, para completar la imagen de la asignatura tenemos a los estudiantes, que son los encargados de elaborar las respuestas a las preguntas planteadas por los docentes. Estas respuestas serán evaluadas por el profesor, autoevaluadas o ambas, según el tipo de pregunta que sea. Para poder organizar la información, cada estudiante se matriculará en un curso, de forma que mantendremos el registro histórico de todos los estudiantes que han pasado por la asignatura. Además, estos alumnos completarán unos formularios de retrospectiva al terminar cada una de las actividades de evaluación, que serán los resultados. A ellos se asociará también la calificación obtenida por el estudiante en dicha acción de evaluación, con lo que se completaría esta idea general del planteamiento de la asignatura.

4.6.2. Diccionario de datos

Una vez tenemos el modelo de dominio, podemos completar el diccionario de datos de entidades y relaciones. El diccionario de datos es un documento fundamental que describe detalladamente el esquema de una base de datos, incluyendo las entidades, atributos, relaciones, restricciones, reglas, así como el origen y uso de los datos. Su principal función es servir como referencia para validar los requisitos de información con los usuarios y proporcionar una guía clara y actualizada de la estructura del almacenamiento durante todo el ciclo de vida de la base de datos.

Diccionario de datos de entidades

Para cada una de las entidades que encontramos en el modelo de dominio de la Figura 4.9 hemos creado un diccionario de datos, que podemos ver en las imágenes que siguen, ordenadas alfabéticamente por el nombre de la entidad.

ID	E04			Nombre:	ACCIÓN			
Definición:	La entidad ACCIÓN representa cada una de las acciones formativas que se llevan a cabo durante el sprint con el fin de alcanzar el objetivo de dicho sprint. Estas acciones pueden ser de aprendizaje o evaluación, y están orientadas a alcanzar el objetivo del sprint.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E04.01	ID	Identificador de la acción.	INTEGER	AUTOINCREMENTAL	SI		NO
	E04.02	Título	Título de la acción.	VARCHAR(64)		SÍ		NO
	E04.03	Descripción	Descripción de la acción, especificando como se va a desarrollar dicha acción. Se deberán indicar aspectos como el lugar, sprint al que pertenece, etc.	VARCHAR(2048)		NO		NO
	E04.04	FechaIni	Fecha de inicio de la acción.	DATE	Formato yyyy-MM-dd	NO		NO
	E04.05	FechaFin	Fecha de fin de la acción.	DATE	Formato yyyy-MM-dd	NO		NO
	E04.06	Carga	Carga de trabajo que conlleva la acción en minutos.	INTEGER	Entero positivo.	NO		NO
	E04.07	Trello	Enlace a la tarjeta de Trello que hace referencia a la acción que estamos considerando.	VARCHAR(256)	URL	SI		SI
	E04.08	Moodle	Enlace al elemento o entrega de Moodle asociado con la acción que estamos considerando.	VARCHAR(256)	URL	SI		SI

Identificadores:	ID	Nombre Atributo
	E04.01	ID

Figura 4.10: Diccionario de datos de la entidad ACCIÓN.

ID	E05			Nombre:	APRENDIZAJE			
Definición:	La entidad APRENDIZAJE hace referencia a aquellas acciones formativas que no conlleven una evaluación asociada y cuyo objetivo sea desarrollar los contenidos de la asignatura para que los alumnos tengan los conocimientos suficientes para cumplir los criterios asociados al aprendizaje. La entidad APRENDIZAJE es una subclase de la entidad ACCIÓN.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E05.01	Tipo	Tipo de acción de aprendizaje realizada, puede ser una clase, un seminario, un vídeo o una lectura.	ENUM	"Clase teórica", "Clase problemas", "Clase laboratorio", "Seminario", "Vídeo", "Lectura", "Retrospectiva", "Soft skills"	NO		NO

Identificadores:	ID	Nombre Atributo						
	E04.01	ID						

Figura 4.11: Diccionario de datos de la entidad APRENDIZAJE.

ID	E03			Nombre:	CRITERIO			
Definición:	La entidad CRITERIO hace referencia a los criterios de aceptación que especifican qué se va a aprender en cada historia y cómo se va a evaluar ese aprendizaje. La especificación de los criterios debe facilitar su medición, para propósitos de evaluación formativa y sumativa.							
Reglas	Cada criterio estará asociado con una sola historia.							

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E03.01	ID	Identificador del criterio.	INTEGER	AUTOINCREMENTAL	SI		NO
	E03.02	Código	Código que identifica el criterio.	VARCHAR(16)		SI		NO
	E03.03	Título	Título del criterio.	VARCHAR(64)		SÍ		NO
	E03.04	Especificación	Descripción del criterio. Debe estar expresada de forma que sea medible y fácilmente identificable si se ha cumplido o no.	VARCHAR(256)		NO		NO

Identificadores:	ID	Nombre Atributo						
	E03.01	ID						

Figura 4.12: Diccionario de datos de la entidad CRITERIO.

ID	E12			Nombre:	CURSO			
Definición:	La entidad CURSO representa cada uno de los cursos académicos en los que se imparte una asignatura.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E12.01	Año	Años del curso académico.	VARCHAR(16)	Formato: yyyy-yyyy	SI		NO

Identificadores:	ID	Nombre Atributo						
	E12.01	Año						

Figura 4.13: Diccionario de datos de la entidad CURSO.

ID	E08			Nombre:	ESTUDIANTE			
Definición:	La entidad ESTUDIANTE representa a los alumnos que forman parte de la asignatura.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E08.01	ID	NIA del estudiante, compuesto por 6 dígitos.	VARCHAR(6)	Debe contener 6 dígitos.	SI		NO
	E08.02	Nombre	Nombre del estudiante.	VARCHAR(512)		NO		NO
	E08.03	Fnac	Fecha de nacimiento del estudiante.	DATE	Formato yyyy-MM-dd	NO		NO
	E08.04	Género	Género del estudiante.	ENUM	"H" o "M"	NO		NO

Identificadores:	ID	Nombre Atributo						
	E08.01	ID						

Figura 4.14: Diccionario de datos de la entidad ESTUDIANTE.

ID	E06		Nombre:	EVALUACIÓN				
Definición:	La entidad EVALUACIÓN es subclase de ACCIÓN y representa las acciones formativas en la que se lleva a cabo una evaluación a los alumnos para consolidar los contenidos desarrollados durante las acciones de aprendizaje. La entidad APRENDIZAJE es una subclase de la entidad ACCIÓN.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E06.01	Alcance	Indica un tipo de acción de evaluación que se puede llevar a cabo en la asignatura.	ENUM	"Básica", "Control", "Final" o "Aceptación".	NO		NO
	E06.02	Obligatoria	Booleano que indica si la actividad de evaluación es obligatoria o voluntaria para la evaluación continua de la asignatura.	BOOLEAN		NO	true	NO

Identificadores:	ID	Nombre Atributo						
	E04.01	ID						

Figura 4.15: Diccionario de datos de la entidad EVALUACIÓN.

ID	E02			Nombre:	HISTORIA			
Definición:	La entidad HISTORIA representa un subobjetivo, es decir, las divisiones específicas de cada uno de los objetivos de aprendizaje. Estas historias de aprendizaje deben tener un alcance acotado, representado por los puntos que hay que cumplir para completar cada uno de los objetivos al completo.							
Reglas	Cada historia estará asociada con un único objetivo.							

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E02.01	ID	Identificador de la historia.	INTEGER	AUTOINCREMENTAL	SI		NO
	E02.02	Título	Título de la historia.	VARCHAR(64)		SÍ		NO
	E02.03	Especificación	Descripción de la historia que permite al alumno conocer con exactitud que debe conseguir para completar la historia.	VARCHAR(512)		NO		NO

Identificadores:	ID	Nombre Atributo						
	E02.01	ID						

Figura 4.16: Diccionario de datos de la entidad HISTORIA.

ID	E01			Nombre:	OBJETIVO			
Definición:	La entidad OBJETIVO establece los hitos principales de la asignatura, dividiendo la visión general de esta en bloques temáticos más pequeños. Estas subdivisiones permiten al estudiante comprender como el objetivo se integra en el producto de aprendizaje que se quiere obtener al final de la asignatura.							
Reglas	En la asignatura Sistemas de Bases de Datos hay un total de 5 objetivos.							

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E01.01	ID	Identificador del objetivo.	INTEGER	AUTOINCREMENTAL	SI		NO
	E01.02	Título	Título del objetivo.	VARCHAR(64)		SÍ		NO
	E01.03	Especificación	Descripción del objetivo que permita al estudiante comprender fácilmente como se integra en el producto de aprendizaje final y que aporta a la asignatura.	VARCHAR(1024)		NO		NO

Identificadores:	ID	Nombre Atributo						
	E01.01	ID						

Figura 4.17: Diccionario de datos de la entidad OBJETIVO.

ID	E14			Nombre:	PLANTILLA			
Definición:	La entidad PLANTILLA hace referencia a la rúbrica que se plantea para evaluar los diferentes tipos de preguntas disponibles en la asignatura.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E07.01	ID	Identificador de la plantilla.	INTEGER		NO		NO
	E07.02	Nombre	Nombre de la plantilla utilizada.	TEXT		NO		NO
	E07.03	Tipo	Tipo de preguntas que evalúa la plantilla.	VARCHAR(128)		NO		NO

Identificadores:	ID	Nombre Atributo						
	E07.01	ID						

Figura 4.18: Diccionario de datos de la entidad PLANTILLA.

ID	E07			Nombre:	PREGUNTA			
Definición:	La entidad PREGUNTA hace referencia a cada una de las partes de una acción de evaluación. Por ejemplo, en una resolución de problemas sería cada uno de los problemas que componen el ejercicio o en un examen cada una de las partes del mismo. Se modela como una entidad débil de PLANTILLA.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E07.01	ID	Identificador de la pregunta.	INTEGER		NO		NO
	E07.02	Título	Título de la pregunta realizada	VARCHAR(64)		NO		NO
	E07.03	Descripción	Descripción de la prueba realizada, puede contener el enunciado de la prueba o un resumen de como se ha planteado el ejercicio para que los alumnos lo resuelvan.	TEXT		NO		NO
	E07.04	Solución	Enlace a la solución de la prueba. También podría ser una secuencia de respuestas en caso de ser la solución de un cuestionario.	VARCHAR(256)	URL	NO		SÍ

Identificadores:	ID	Nombre Atributo						
	E04.01	ID (PLANTILLA)						
	E07.01	ID						

Figura 4.19: Diccionario de datos de la entidad PREGUNTA.

ID	E09			Nombre:	RESPUESTA			
Definición:	La entidad RESPUESTA representa cada una de las soluciones que presenta un estudiante para las pruebas que realiza. Normalmente será una única respuesta por estudiante por prueba, pero puede haber ocasiones en las que un estudiante pueda presentar varias soluciones para la misma prueba. Esta entidad se modela como una entidad débil dependiente de la entidad ESTUDIANTE.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E09.01	ID	Identificador de la solución.	INTEGER	Autoincremental para cada estudiante	NO		NO
	E09.02	Fecha	Fecha en la que se entregó la respuesta.	DATE	Formato yyyy-MM-dd	NO		NO
	E09.03	Contenido	Contenido de la solución presentada como respuesta a la prueba. Puede ser un enlace a una solución o un conjunto de respuestas.	TEXT		NO		NO
	E09.04	Apto	Representa si el estudiante ha superado el mínimo de nota para considerarse apto. Es un atributo derivado de la (auto)evaluación de todos los criterios que evaluados en la pregunta a la que se da respuesta.	ENUM	"Apto" o "No apto". Se considera "apto" si todos los criterios evaluados en la pregunta se (auto)evalúan como "apto".	NO		NO
	E09.05	Evaluación	La nota que ha obtenido el alumno en dicha prueba sobre 10 en la evaluación realizada por el profesor. Es un atributo derivado.	DECIMAL(4,2)	Solo reciben calificación numérica las respuestas aptas y se calculan a partir del valor de calidad de cada criterio evaluado y su ponderación en la prueba.	NO		Sí
	E09.06	Autoevaluación	La nota que ha obtenido el alumno en dicha prueba sobre 10 en su propia autoevaluación de la respuesta dada. Es un atributo derivado.	DECIMAL(4,2)	Solo reciben calificación numérica las respuestas aptas y se calculan a partir del valor de calidad de cada criterio autoevaluado y su ponderación en la prueba.	NO		Sí

Identificadores:	ID	Nombre Atributo
	E08.01	ID (ESTUDIANTE)
	E09.01	ID

Figura 4.20: Diccionario de datos de la entidad RESPUESTA.

ID	E11			Nombre:	RESULTADO			
Definición:	La entidad RESULTADO hace referencia a los cuestionarios completados al finalizar cada una de las acciones de evaluación en los que evalúan los resultados obtenidos. Esta entidad se modela como una entidad débil dependiente de la entidad ESTUDIANTE.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E11.01	ID	Identificador del resultado.	INTEGER	AUTOINCREMENTAL para cada estudiante.	NO		NO
	E11.02	Fecha	Fecha de realización del cuestionario.	DATE	Formato yyyy-MM-dd	NO		NO
	E11.03	Contenido	Respuesta completa del estudiante al cuestionario propuesto.	TEXT		NO		NO

Identificadores:	ID	Nombre Atributo						
	E08.01	ID (ESTUDIANTE)						
	E11.01	ID						

Figura 4.21: Diccionario de datos de la entidad RESULTADO.

ID	E13		Nombre:	SPRINT				
Definición:	La entidad SPRINT hace referencia a los sprints que se desarrollan a lo larg del curso en la asignatura. Para la asignatura de Sistemas de Bases de Datos tendremos tres sprints por curso. Esta entidad se modela como una entidad débil dependiente de la entidad CURSO.							
Reglas	Hay un máximo de tres sprints por curso.							

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E13.01	Número	Número de orden del sprint en el curso.	INTEGER	Puede tomar los valores 1, 2 o 3.	NO		NO
	E13.02	Finicio	Fecha de inicio del sprint.	DATE	Formato yyyy-MM-dd	SÍ		NO
	E13.03	Ffin	Fecha de finalización del sprint.	DATE	Formato yyyy-MM-dd	SÍ		NO

Identificadores:	ID	Nombre Atributo						
	E12.01	Año						
	E13.01	Número						

Figura 4.22: Diccionario de datos de la entidad SPRINT.

ID	E10			Nombre:	TAREA			
Definición:	La entidad TAREA representa cada una de las actividades que el estudiante realiza a lo largo de la asignatura. Todo el esfuerzo que un estudiante invierte en la asignatura se recoge en el cuaderno de trabajo, con su respectiva descripción y el tiempo invertido en ella. Esta entidad se modela como una entidad débil dependiente de la entidad ESTUDIANTE.							
Reglas								

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	E10.01	ID	Identificador de la tarea.	INTEGER	AUTOINCREMENTAL para cada estudiante.	NO		NO
	E10.02	Fecha	Fecha de realización de la tarea.	DATE	Formato yyyy-MM-dd	NO		NO
	E10.03	Actividad	Actividad que ha conllevado la realización de la tarea.	ENUM	Puede ser: "Asistencia a clases", "Trabajo individual" o "Trabajo en equipo".	NO		NO
	E10.04	Tipo	Dependiendo de la actividad realizada, la tarea puede ser de un tipo o de otro. Indica de forma más detallada el tipo de actividad que se ha realizado.	ENUM	Dependiendo del valor de actividad, puede ser: - Asistencia a clases: "Teoría", "Problemas", "Laboratorio", "Apoyo", "Examen". - Trabajo individual: "Realización de tarea", "Autoevaluación", "Hall of Fame", "Lectura o preparación de apuntes", "Estudio", "Tutoría individual", "Otros". - Trabajo en equipo: "Trabajo individual", "Trabajo con otros compañeros", "Reunión", "Tutoría grupal", "Otros".	NO		NO
	E10.05	Tiempo	Tiempo en minutos invertido para realizar dicha tarea.	INTEGER		NO		NO
	E10.06	Comentario	Breve descripción o apunte de la tarea realizada.	VARCHAR(200)		NO		Sí

Identificadores:	ID	Nombre Atributo						
	E08.01	ID (ESTUDIANTE)						
	E10.01	ID						

Figura 4.23: Diccionario de datos de la entidad TAREA.

Diccionario de datos de relaciones

Por otro lado, cada una de las relaciones que encontramos en el modelo de dominio de la Figura 4.9 aparece en uno de los diccionarios de datos mostrados a continuación, ordenados alfabéticamente por el nombre de la relación.

ID	R03		Nombre:	ASOCIAR
Definición:	La relación ASOCIAR vincula cada acción formativa con las historias de aprendizaje que se abordan en ella (a nivel de aprendizaje o evaluación).			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E02	HISTORIA	1	N
	E04	ACCION	1	N

Figura 4.24: Diccionario de datos de la relación ASOCIAR.

ID	R08		Nombre:	AUTOEVALUAR
Definición:	La relación AUTOEVALUAR concreta el estado del aprendizaje del estudiante (desde su propia perspectiva) que plantea la respuesta respecto a cada uno de los criterios de aceptación valorados en la prueba.			
Reglas	Esta información se obtiene a partir de las rúbricas de autoevaluación completadas por el estudiante.			

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E09	RESPUESTA	0	N
	E03	CRITERIO	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R07.01	Calidad	Da un valor numérico que representa la calidad de la respuesta presentada por el estudiante.	DECIMAL(3,2)	Es un valor entre 0 y 1	NO		NO
	R07.02	Estado	Define si la respuesta del estudiante cumple el umbral de calidad necesario para superar el criterio en cuestión. Se calcula observando si la calidad es superior al umbral definido para ese criterio.	ENUM	"APTO" o "NO APTO". Se obtiene a partir de la calidad de la respuesta y el umbral fijado en la prueba para el criterio de aceptación.	NO		NO

Figura 4.25: Diccionario de datos de la relación AUTOEVALUAR.

ID	R01		Nombre:	CONTENER
Definición:	La relación CONTENER asocia cada objetivo de aprendizaje con las historias de aprendizaje que lo materializan. Es decir, refina en subobjetivos cada uno de los objetivos de aprendizaje de la asignatura.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E01	OBJETIVO	1	N
	E02	HISTORIA	1	1

Figura 4.26: Diccionario de datos de la relación CONTENER.

ID	R12		Nombre:	CORRESPONDER
Definición:	La relación CORRESPONDER vincula cada tarea realizada por un estudiante en una determinada acción formativa.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E04	ACCION	1	N
	E10	TAREA	1	1

Figura 4.27: Diccionario de datos de la relación CORRESPONDER.

ID	R02		Nombre:	DESARROLLAR
Definición:	La relación DESARROLLAR asocia cada historia con los criterios de aceptación que desarrollan el aprendizaje que se aborda en ella.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E02	HISTORIA	1	N
	E03	CRITERIO	1	1

Figura 4.28: Diccionario de datos de la relación DESARROLLAR.

ID	R04		Nombre:	ENSEÑAR
Definición:	La relación ENSEÑAR asocia cada acción formativa (de aprendizaje) con los criterios que describen el aprendizaje que se aborda en ella.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E05	APRENDIZAJE	1	N
	E03	CRITERIO	1	N

Figura 4.29: Diccionario de datos de la relación ENSEÑAR.

ID	R07		Nombre:	EVALUAR	
Definición:	La relación EVALUAR concreta el estado del aprendizaje del estudiante (desde la perspectiva del profesor) que plantea la respuesta respecto a cada uno de los criterios de aceptación valorados en la prueba.				
Reglas	Esta información se obtiene a partir de las rúbricas de evaluación completadas por el profesor.				

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E09	RESPUESTA	0	N
	E03	CRITERIO	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R07.01	Calidad	Da un valor numérico que representa la calidad de la respuesta presentada por el estudiante.	DECIMAL(3,2)	Es un valor entre 0 y 1	NO		NO
	R07.02	Estado	Define si la respuesta del estudiante cumple el umbral de calidad necesario para superar el criterio en cuestión. Se calcula observando si la calidad es superior al umbral definido para ese criterio.	ENUM	"APTO" o "NO APTO". Se obtiene a partir de la calidad de la respuesta y el umbral fijado en la prueba para el criterio de aceptación.	NO		NO

Figura 4.30: Diccionario de datos de la relación EVALUAR.

ID	R11		Nombre:	HACER	
Definición:	La relación HACER vincula a cada estudiante con las tareas que realiza en el ámbito de la asignatura.				
Reglas	Esta información se obtiene a partir de los cuadernos de trabajo del estudiante.				

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E08	ESTUDIANTE	1	N
	E10	TAREA	1	1

Figura 4.31: Diccionario de datos de la relación HACER.

ID	R05		Nombre:	INCLUIR				
Definición:	La relación INCLUIR establece las diferentes preguntas que se tiene una acción formativa de evaluación.							
Reglas								

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E06	EVALUACIÓN	1	N
	E07	PREGUNTA	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R05.01	Orden	Define el orden de la pregunta en la acción de evaluación en la que se está utilizando.	INTEGER	Para cada acción de evaluación debe ser AUTOINCREMENTAL.	NO		NO

Figura 4.32: Diccionario de datos de la relación INCLUIR.

ID	R21		Nombre:	INSTANCIAR
Definición:	La relación INSTANCIAR representa que plantilla se utiliza para evaluar la pregunta en cuestión.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E07	PREGUNTA	1	1
	E14	PLANTILLA	1	N

Figura 4.33: Diccionario de datos de la relación INSTANCIAR.

ID	R13	Nombre:	IS_A_ACCIÓN
Definición:	La relación IS_A_ACCIÓN modela que cada una de las acciones formativas que se realizan a lo largo de la asignatura son o bien una acción de aprendizaje o una acción de evaluación. Las primeras deben servir para desarrollar una serie de conceptos que se afianzarán en las acciones de evaluación. La IS_A es obligatoria y disjunta.		
Reglas			

Entidades	ID	Nombre Entidad	Superclase / Subclase	Obligatoria	Disjunta
	E04	ACCIÓN	Superclase	SI	SI
	E05	APRENDIZAJE	Subclase		
	E06	EVALUACIÓN	Subclase		

Figura 4.34: Diccionario de datos de la relación IS_A_ACCIÓN.

ID	R17		Nombre:	MATRICULAR				
Definición:	La relación MATRICULAR vincula a cada estudiante con el curso en el que está matriculado.							
Reglas								

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E08	ESTUDIANTE	1	N
	E12	CURSO	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R17.01	Calif_o	Calificación de la convocatoria ordinaria del estudiante. Es la representación como texto de la nota obtenida por el estudiante en la convocatoria ordinaria.	ENUM	"SUSPENSO", "APROBADO", "NOTABLE", "SOBRESALIENTE" O "MATRÍCULA HONOR".	NO		SI
	R17.02	Nota_o	Nota de la convocatoria ordinaria del estudiante. Es el valor numérico obtenido que representa el aprendizaje adquirido por el estudiante.	DECIMAL(3,1)	Valor entre 0 y 10.	NO		SI
	R17.03	Calif_e	Calificación de la convocatoria extraordinaria del estudiante. Es la representación como texto de la nota obtenida por el estudiante en la convocatoria extraordinaria.	ENUM	"SUSPENSO", "APROBADO", "NOTABLE", "SOBRESALIENTE" O "MATRÍCULA HONOR".	NO		SI
	R17.04	Nota_e	Nota de la convocatoria extraordinaria del estudiante. Es el valor numérico obtenido que representa el aprendizaje adquirido por el estudiante.	DECIMAL(3,1)	Valor entre 0 y 10.	NO		SI
	R17.05	Número	Representa el número de veces que se ha matriculado un estudiante en la asignatura. Se calcula con el número de cursos diferentes en los que el estudiante ha estado matriculado.	INTEGER	Se calcula a partir de los cursos diferentes en los que un estudiante ha estado matriculado.	NO		NO

Figura 4.35: Diccionario de datos de la relación MATRICULAR.

ID	R06		Nombre:	MEDIR				
Definición:	La relación MEDIR asocia cada plantilla con los diferentes criterios de aceptación que se valoran en ella.							
Reglas	Los criterios asociados a cada prueba deben forma parte de los criterios que desarrollan el aprendizaje de las historias vinculadas a la acción que contiene la prueba.							

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E07	PLANTILLA	1	N
	E03	CRITERIO	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R06.01	Umbral	Define el límite inferior que debe conseguir el alumno en cada uno de los criterios para que se considere APTO.	DECIMAL(3,2)	Es un valor entre 0 y 1	NO		NO
	R06.02	Ponderación	Define cuanto porcentaje del total de la nota de la acción de evaluación cuenta el criterio.	DECIMAL(4,3)	Es un valor entre 0 y 1	NO		NO
	R06.03	Especificación	Indica la forma en la que se calcula la valoración del criterio.	VARCHAR (256)		NO	NULL	Sí

Figura 4.36: Diccionario de datos de la relación MEDIR.

ID	R19		Nombre:	OBTENER	
Definición:	La relación OBTENER asocia a cada estudiante los resultados que ha obtenido.				
Reglas					

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E08	ESTUDIANTE	1	N
	E11	RESULTADO	1	1

Figura 4.37: Diccionario de datos de la relación OBTENER.

ID	R16		Nombre:	ORGANIZAR	
Definición:	La relación ORGANIZAR concreta que cada uno de los cursos de la asignatura se divide en tres sprints.				
Reglas					

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E12	CURSO	1	N
	E13	SPRINT	1	1

Figura 4.38: Diccionario de datos de la relación ORGANIZAR.

ID	R14		Nombre:	PLANIFICAR	
Definición:	La relación PLANIFICAR vincula cada sprint con todas las historias que se ha planificado abordar en dicho sprint.				
Reglas					

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E02	HISTORIA	1	N
	E13	SPRINT	1	N

Figura 4.39: Diccionario de datos de la relación PLANIFICAR.

ID	R10		Nombre:	PLANTEAR
Definición:	La relación PLANTEAR vincula cada respuesta con el estudiante que la realiza, dentro de una determinada acción formativa de evaluación.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E08	ESTUDIANTE	1	N
	E09	RESPUESTA	1	1

Figura 4.40: Diccionario de datos de la relación PLANTEAR.

ID	R15		Nombre:	PROGRAMAR
Definición:	La relación PROGRAMAR concreta que cada una de las acciones, ya sean de aprendizaje o evaluación, tienen lugar en uno de los sprints del curso.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E04	ACCION	1	1
	E13	SPRINT	1	N

Figura 4.41: Diccionario de datos de la relación PROGRAMAR.

ID	R09		Nombre:	RESOLVER
Definición:	La relación RESOLVER asocia cada respuesta del estudiante con la prueba en la que se realiza, dentro de una determinada acción formativa de evaluación.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E07	PREGUNTA	1	N
	E09	RESPUESTA	1	1

Figura 4.42: Diccionario de datos de la relación RESOLVER.

ID	R20		Nombre:	SUPERAR
Definición:	La relación REALIZAR representa que criterios se considera que se han superado y cuales no en el resultado del estudiante en cada acción de evaluación.			
Reglas				
Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E03	CRITERIO	1	N
	E11	RESULTADO	0	N

Figura 4.43: Diccionario de datos de la relación SUPERAR.

ID	R18		Nombre:	VALORAR				
Definición:	La relación VALORAR vincula el resultado obtenidi por un estudiante en la acción de evaluación.							
Reglas								

Entidades	ID	Nombre Entidad	Participación	Cardinalidad
	E06	EVALUACIÓN	1	N
	E11	RESULTADO	1	N

Atributos:	ID	Nombre	Definición	Tipo	Reglas	UNIQUE	DEFAULT	NULL
	R18.01	Evaluación	Nota de evaluación, indicada por el profesor, obtenida en total por el estudiante en la acción de evaluación.	DECIMAL (3,1)	Valor entre 5 y 10. Solo se obtendrá una nota si se ha superado la acción de evaluación.	NO		SI

Figura 4.44: Diccionario de datos de la relación VALORAR.

4.7. Mapeo de los orígenes de datos

Una vez disponemos del diseño de la base de datos, tenemos que saber cómo vamos a informar dichas tablas con las fuentes de datos que disponemos. Para ello, hemos construido, para cada una de estas fuentes de datos, un *Logical Data Map*, de forma que veamos esquemáticamente el origen, transformaciones y destino de los datos que se van a utilizar.

Un mapa lógico de datos [2] (o *Logical Data Map* en inglés) es una herramienta que permite representar de manera estructurada cómo se van a utilizar los datos obtenidos de las distintas fuentes de datos. En este caso, cada tabla representa una fuente de datos, indicando el origen específico de los datos, las transformaciones aplicadas (si las hay) y su destino final asociado con el modelo de dominio explicado anteriormente. Estos mapas facilitan la comprensión del flujo de datos dentro del sistema, permitiendo identificar claramente cómo se informa cada parte del modelo con la información disponible.

En nuestro caso, hemos completado las siguientes tablas para representar estos mapas lógicos de datos.

ORIGEN				TRANSFORMACIONES			DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Paso 3	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Autoevaluación	Ruta del archivo	Nombre de la carpeta	STRING	Obtener el nombre del estudiante. Formato APELLIDO1 APELLIDO2, NOMBRE	Cambiar formato a: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	AUTOEVALUAR	ESTUDIANTE	INTEGER
Autoevaluación	Hoja: evaluación	Columna: C - Código del criterio	STRING	Obtener el Código del criterio a evaluar.	Encontrar el ID del CRITERIO que se corresponde con ese código.	-	AUTOEVALUAR	CRITERIO	INTEGER
Autoevaluación	Hoja: evaluación	Columna: F - Evaluación del criterio	FLOAT	Obtener el valor de la evaluación del criterio.	Transformar el formato, convertirlo en decimal separado por punto.	-	AUTOEVALUAR	Calidad	DECIMAL(2,2)
Autoevaluación	Hoja: evaluación	Columna: I - Umbral del criterio	FLOAT	Obtener el valor del umbral del criterio.	Transformar el formato, convertirlo en decimal separado por punto.	-	MEDIR	Umbral	DECIMAL(3,2)
Autoevaluación	Hoja: evaluación	Columna: J - Valoración del criterio	FLOAT	Obtener el valor de la valoración del criterio.	Transformar el formato, convertirlo en decimal separado por punto.	-	MEDIR	Ponderación	DECIMAL(2,2)
Autoevaluación	Ruta del archivo	Nombre de la carpeta	STRING	Obtener el nombre del estudiante. Formato APELLIDO1 APELLIDO2, NOMBRE	Cambiar formato a: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	PLANTEAR	Estudiante	INTEGER
Autoevaluación	Ruta del archivo	Nombre del archivo	STRING	Obtener nombre del archivo.	Buscar coincidencias entre el nombre del archivo y las preguntas de la acción procesada.	Se obtiene el ID de la PLANTILLA a la que esta asociada dicha pregunta.	RESOLVER	PLANTILLA	INTEGER
Autoevaluación	Ruta del archivo	Nombre del archivo	STRING	Obtener nombre del archivo.	Buscar coincidencias entre el nombre del archivo y las preguntas de la acción procesada.	Se obtiene el ID de la PREGUNTA cuyo nombre está contenido en el del archivo.	RESOLVER	PREGUNTA	INTEGER

Figura 4.45: Mapeo lógico de la fuente de datos: AUTOEVALUACIONES.

ORIGEN				TRANSFORMACIONES			DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Paso 3	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Cuaderno de trabajo	Ruta del archivo	Nombre de la carpeta	STRING	Obtener el nombre del estudiante. Formato APELLIDO1 APELLIDO2, NOMBRE	Cambiar formato a: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	HACER	-	INTEGER
Cuaderno de trabajo	Hoja: cuaderno	Columna: B - Fecha	STRING	Cambiar formato a: yyyy-MM-dd	-	-	TAREA	Fecha	DATE
Cuaderno de trabajo	Hoja: cuaderno	Columna: E - Tiempo	INTEGER	Verificar que no hay datos erróneos que no sean texto.	-	-	TAREA	Tiempo	INTEGER
Cuaderno de trabajo	Hoja: cuaderno	Columna: F - Comentario	STRING	-	-	-	DATE	Comentario	VARCHAR(200)
Cuaderno de trabajo	-	-	-	No se dispone de esta información.	-	-	CORRESPONDER	-	INTEGER

Figura 4.46: Mapeo lógico de la fuente de datos: CUADERNOS DE TRABAJO.

4.7. Mapeo de los orígenes de datos

ORIGEN				TRANSFORMACIONES		DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Listado de estudiantes	Hoja: listado	Columna: L - N.I.A.	STRING	Obtener valor numérico.	Añadir 0 al inicio hasta completar los 6 dígitos que lo componen.	ESTUDIANTE	NIA	VARCHAR(6)
Listado de estudiantes	Hoja: listado	Columnas: B a H - Apellidos y Nombre	STRING	Transformar a formato de Nombre Apellido1 Apellido2	-	ESTUDIANTE	Nombre	VARCHAR(512)
Listado de estudiantes	Hoja: listado	Columnas: T a V - Fecha nacimiento	STRING	Convertir a formato yyyy-MM-dd	-	ESTUDIANTE	Fnac	DATE
Listado de estudiantes	Hoja: listado	Columnas: W a X - Género	STRING	Transformar "Hombre" a "H" y "Mujer" a "M"	-	ESTUDIANTE	Género	ENUM
Listado de estudiantes	Hoja: listado	Columna: L - N.I.A.	STRING	Obtener valor numérico.	Añadir 0 al inicio hasta completar los 6 dígitos que lo componen.	MATRICULAR	ESTUDIANTE	INTEGER
Listado de estudiantes	Ruta del archivo	Carpeta en la que está contenido	STRING	Obtener nombre de la carpeta en la que está contenida la carpeta "estudiantes"	Verificar formato del curso: yyyy-yyyy.	MATRICULAR	CURSO	VARCHAR(16)
Listado de estudiantes	Hoja: listado	Columna: Y a AA - Veces. Matri.	INTEGER	Obtener valor numérico.	-	MATRICULAR	Número	INTEGER

Figura 4.47: Mapeo lógico de la fuente de datos: LISTADO DE ESTUDIANTES.

ORIGEN				TRANSFORMACIONES			DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Paso 3	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Evaluación	Hoja: Examen	Columnas B, C	STRING	Concatenar ambas columnas del nombre.	Verificar formato: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	EVALUAR	ESTUDIANTE	INTEGER
Evaluación	Hoja: Examen	Fila 2 - Columnas D hasta final de archivo.	STRING	Obtener el Código del criterio a evaluar.	Encontrar el ID del CRITERIO que se corresponde con ese código.	-	EVALUAR	CRITERIO	INTEGER
Evaluación	Hoja: Examen	Columnas D hasta final de archivo.	FLOAT	Obtener el valor de la evaluación del criterio.	Transformar el formato, convirtiéndolo en decimal separando por punto.	-	EVALUAR	Calidad	DECIMAL(2,2)
Evaluación	Hoja: Examen	Nombre de la carpeta	STRING	Concatenar ambas columnas del nombre.	Verificar formato: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	PLANTEAR	Estudiante	INTEGER
Evaluación	Ruta del archivo	Nombre del archivo	STRING	Obtener nombre del archivo.	Buscar coincidencias entre el nombre del archivo y las preguntas de la acción procesada.	Se obtiene el ID de la PLANTILLA a la que esta asociada dicha pregunta.	RESOLVER	PLANTILLA	INTEGER
Evaluación	Ruta del archivo	Nombre del archivo	STRING	Obtener nombre del archivo.	Buscar coincidencias entre el nombre del archivo y las preguntas de la acción procesada.	Se obtiene el ID de la PREGUNTA cuyo nombre está contenido en el del archivo.	RESOLVER	PREGUNTA	INTEGER

Figura 4.48: Mapeo lógico de la fuente de datos: EVALUACIONES.

ORIGEN				TRANSFORMACIONES		DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Formulario de retrospectiva	Hoja: Form1	Columna: E - Nombre	STRING	Cambiar formato a: Nombre Apellido1 Apellido2	Encontrar el ID del ESTUDIANTE que se corresponde con ese nombre.	OBTENER	-	INTEGER
Formulario de retrospectiva	Hoja: Form1	Columna: E - Nombre	STRING	Obtener parte correspondiente a la fecha, omitimos la hora.	Convertir en: DATE con formato yyyy-MM-dd	RESULTADO	Fecha	DATE
Formulario de retrospectiva	Hoja: Form1	Columnas de respuesta a las preguntas	STRING	Concatenar todas las preguntas.		RESULTADO	Contenido	TEXT
Formulario de retrospectiva	Hoja: Form1	Columnas de criterios superados	STRING	Obtener código de cada criterio.	Encontrar el ID del CRITERIO que se corresponde con ese código.	SUPERAR	CRITERIO	INTEGER
Formulario de retrospectiva	Ruta del archivo	Nombre del archivo	STRING	Encontrar el ID de la ACCIÓN que se corresponde con ese Nombre.	-	VALORAR	EVALUACIÓN	INTEGER

Figura 4.49: Mapeo lógico de la fuente de datos: FORMULARIOS.

ORIGEN				TRANSFORMACIONES	DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Objetivos de los sprints	Historias de aprendizaje	Criterios de aceptación	STRING	Obtener código (destacado en negrita y azul)	CRITERIO	Código	VARCHAR(16)
Objetivos de los sprints	Historias de aprendizaje	Criterios de aceptación	STRING	Pequeño resumen de especificación	CRITERIO	Título	VARCHAR(64)
Objetivos de los sprints	Historias de aprendizaje	Criterios de aceptación	STRING	Obtener descripción de cada criterio	CRITERIO	Especificación	VARCHAR(256)
Objetivos de los sprints	Historias de aprendizaje	Título del subapartado.	STRING	Encontrar el ID de la HISTORIA que se corresponde con ese Título.	DESARROLLAR	-	INTEGER
Objetivos de los sprints	Historias de aprendizaje	Título del subapartado.	STRING	-	HISTORIA	Título	INTEGER
Objetivos de los sprints	Objetivos de aprendizaje	Como, quiero, para. Después del título, resaltado en azul.	STRING	Obtener fracción de texto posterior al QUIERO, hasta el final.	HISTORIA	Especificación	VARCHAR(512)
Objetivos de los sprints	Objetivos de aprendizaje.	Título de los apartados	STRING	Encontrar el ID del OBJETIVO que se corresponde con ese Título.	CONTENER	-	INTEGER
Objetivos de los sprints	Objetivos de aprendizaje.	Título de los apartados	STRING	-	OBJETIVO	Título	VARCHAR(64)
Objetivos de los sprints	Objetivos de aprendizaje.	Párrafo posterior al inicio del apartado.	STRING	Información contenida entre el inicio del apartado y el comienzo del siguiente subapartado.	OBJETIVO	Especificación	VARCHAR(1024)
Objetivos de los sprints	Documento al completo.	Título del documento.	STRING	Obtener número de sprint del título del documento.	PLANIFICAR	SPRINT	INTEGER
Objetivos de los sprints	Historias de aprendizaje	Subapartados del documento.	STRING	Encontrar el ID de la historia que se corresponde con ese Título.	PLANIFICAR	HISTORIA	INTEGER

Figura 4.50: Mapeo lógico de la fuente de datos: OBJETIVOS DE LA ASIGNATURA.

ORIGEN				TRANSFORMACIONES		DESTINO		
Fuente	Objeto	Campo	Tipo	Paso 1	Paso 2	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Tablero de Trello	Tarea asociada a la acción	Título de la tarea	STRING	-	-	ACCIÓN	Título	VARCHAR(64)
Tablero de Trello	Tarea asociada a la acción	Descripción de la tarea	STRING	-	-	ACCIÓN	Descripción	VARCHAR(1024)
Tablero de Trello	Tarea asociada a la acción	Fecha (de inicio) de la tarea	STRING	Convertir en: DATE de formato yyyy-MM-dd	-	ACCIÓN	FechaIni	DATE
Tablero de Trello	Tarea asociada a la acción	Fecha (de finalización) de la tarea	STRING	Convertir en: DATE de formato yyyy-MM-dd	-	ACCIÓN	FechaFin	DATE
Tablero de Trello	Tarea de entrega en Moodle	Enlace: Tarea al completo	STRING	-	-	ACCIÓN	Moodle	VARCHAR(256)
Tablero de Trello	Tarea asociada a la acción	Enlace: Tarea al completo	STRING	-	-	ACCIÓN	Trello	VARCHAR(256)
Tablero de Trello	Tablero	Nombre del tablero	STRING	Extraer cadena texto entre corchetes: [yyyy-yyyy]	-	ACCIÓN	Curso	VARCHAR(16)
Tablero de Trello	Tarea asociada a la acción	Fecha (de finalización) de la tarea	STRING	Insertar a mano sprint que corresponde según la fecha en la que se ha realizado la acción.	-	PROGRAMAR	-	INTEGER
Tablero de Trello	Tarea asociada a la acción	Etiqueta tipo de la acción	STRING	-	-	APRENDIZAJE	Tipo	ENUM
Tablero de Trello	Tarea asociada a la acción	Descripción de la tarea	STRING	Leer descripción para obtener la información necesaria.	Identificar cuál de las opciones posibles en el ENUM es.	EVALUACIÓN	Alcance	ENUM
Tablero de Trello	Tarea asociada a la acción	Descripción de la tarea	STRING	Leer descripción para obtener la información necesaria.	Identificar si es obligatoria o no.	EVALUACIÓN	Obligatoria	BOOLEAN
Tablero de Trello	Tarea asociada a la acción	Checklist de criterios: [...] En esta clase...	STRING	Obtener código de cada criterio. En negrita.	Encontrar el ID del CRITERIO que se corresponde con ese código.	ENSEÑAR	CRITERIO	INTEGER
Tablero de Trello	Tarea asociada a la acción	Checklist de criterios: [...] En esta clase...	STRING	Obtener código de cada criterio. En negrita.	Encontrar el ID de la HISTORIA a la que está asociado dicho Criterio.	ASOCIAR	HISTORIA	INTEGER

Figura 4.51: Mapeo lógico de la fuente de datos: TABLERO DE TRELLO.

ORIGEN		TRANSFORMACIONES	DESTINO		
Fuente	Tipo	Paso 1	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Inserción manual	STRING	Comprobar formato yyyy-yyyy.	CURSO	Año	VARCHAR(16)
Inserción manual	STRING	Comprobar formato decimal.	MATRICULAR	Nota_o	DECIMAL(3,1)
Inserción manual	STRING	Comprobar formato decimal.	MATRICULAR	Nota_e	DECIMAL(3,1)
Inserción manual	STRING	Comprobar valor entre los cursos existentes.	PLANIFICAR	CURSO	VARCHAR(16)
Inserción manual	STRING	Comprobar valor entre los cursos existentes.	SPRINT	Curso	VARCHAR(16)
Inserción manual	STRING	Debe ser el siguiente disponible para ese curso.	SPRINT	Número	SMALLINT
Inserción manual	STRING	Comprobar formato: yyyy-MM-dd y que es posterior a la de fin del sprint anterior.	SPRINT	Finicio	DATE
Inserción manual	STRING	Comprobar formato: yyyy-MM-dd y que es posterior a la de inicio.	SPRINT	Ffin	DATE
Inserción manual	STRING	ID de la ACCIÓN que contiene las preguntas.	INCLUIR	ACCIÓN	INTEGER
Inserción manual	STRING	ID de la PLANTILLA que sigue la pregunta incluida.	INCLUIR	PLANTILLA	INTEGER
Inserción manual	STRING	ID de la pregunta que está en la acción creada.	INCLUIR	PREGUNTA	INTEGER
Inserción manual	STRING	Orden de aparición de la pregunta en la acción de evaluación.	INCLUIR	Orden	INTEGER
Inserción manual	STRING	Nombre de la plantilla a seguir.	PLANTILLA	Nombre	TEXT
Inserción manual	STRING	Tipo de la plantilla.	PLANTILLA	Tipo	VARCHAR(128)
Inserción manual	STRING	Debe estar entre los valores de los criterios existentes.	MEDIR	CRITERIO	INTEGER
Inserción manual	STRING	Explicación de como se mide el criterio	MEDIR	Especificación	VARCHAR(256)
Inserción manual	STRING	Plantilla con la que se evalúa la pregunta a insertar.	INSTANCIAR	-	INTEGER
Inserción manual	STRING	Título de la pregunta que debe estar contenido en el nombre del archivo de las (auto)evaluaciones.	PREGUNTA	Título	VARCHAR(32)
Inserción manual	STRING	Completar con el enunciado de la pregunta.	PREGUNTA	Descripción	TEXT
Inserción manual	STRING	Enlace a la solución de la pregunta.	PREGUNTA	Solución	VARCHAR(256)
Inserción manual	STRING	-	RESPUESTA	Contenido	TEXT

Figura 4.52: Mapeo lógico de los datos insertados MANUALMENTE.

4.7. Mapeo de los orígenes de datos

ORIGEN		TRANSFORMACIONES	DESTINO		
Fuente	Tipo	Paso 1	Entidad/Relación	Atributo/Entidad asociada	Tipo dato
Generado automáticamente	INTEGER	AUTOINCREMENTAL	ACCIÓN	ID	INTEGER
Generado automáticamente	STRING	Generado en función de la calidad de los criterios y los umbrales determinados.	AUTOEVALUAR	Estado	ENUM
Generado automáticamente	INTEGER	AUTOINCREMENTAL	CRITERIO	ID	INTEGER
Generado automáticamente	STRING	Generado en función de la calidad de los criterios y los umbrales determinados.	EVALUAR	Estado	ENUM
Generado automáticamente	INTEGER	AUTOINCREMENTAL	HISTORIA	ID	INTEGER
Generado automáticamente	INTEGER	AUTOINCREMENTAL	PLANTILLA	ID	INTEGER
Generado automáticamente	INTEGER	AUTOINCREMENTAL	OBJETIVO	ID	INTEGER
Generado automáticamente	INTEGER	Genera un número superior al anterior valor de ID obtenido para dicho ESTUDIANTE.	RESULTADO	ID	INTEGER
Generado automáticamente	INTEGER	Genera un número superior al anterior valor de ID obtenido para dicho ESTUDIANTE.	TAREA	ID	INTEGER
Generado automáticamente	DECIMAL(3,1)	Se genera a partir de la media de las notas de las preguntas que componen dicha acción.	VALORAR	Evaluación	DECIMAL(3,1)
Generado automáticamente	STRING	Se calcula automáticamente a partir de la Nota_o obtenida.	MATRICULAR	Calif_o	ENUM
Generado automáticamente	STRING	Se calcula automáticamente a partir de la Nota_e obtenida.	MATRICULAR	Calif_e	ENUM
Generado automáticamente	STRING	Se carga automáticamente el valor generado para el ID de la plantilla.	MEDIR	PLANTILLA	INTEGER
Generado automáticamente	INTEGER	Genera un número superior al anterior valor de ID obtenido para dicha PLANTILLA.	PREGUNTA	ID	INTEGER
Generado automáticamente	INTEGER	Genera un número superior al anterior valor de ID obtenido para dicho ESTUDIANTE.	RESPUESTA	ID	INTEGER
Generado automáticamente	INTEGER	Se toma el valor del ID de respuesta generado.	AUTOEVALUAR	RESPUESTA	INTEGER
Generado automáticamente	DATE	Se toma el día en el que se procesa la respuesta.	RESPUESTA	Fecha	DATE
Generado automáticamente	STRING	Generado en función del estado de todos los criterios que se evalúan en ella.	RESPUESTA	Apto	ENUM
Generado automáticamente	DECIMAL(4,2)	Se calcula a partir de las ponderaciones de los criterios evaluados y sus notas de calidad.	RESPUESTA	Evaluación	DECIMAL(4,2)
Generado automáticamente	INTEGER	Se calcula a partir de las ponderaciones de los criterios evaluados y sus notas de calidad.	RESPUESTA	Autoevaluación	DCIMAL(4,2)

Figura 4.53: Mapeo lógico de los datos insertados AUTOMÁTICAMENTE.

Capítulo 5

Diseño

Una vez se ha completado la fase de análisis del software, pasaremos a la etapa de diseño. En ella, nos enfocaremos en diseñar una solución que satisfaga los objetivos del proyecto, de acuerdo con los requisitos y condicionantes que hemos ido identificando previamente. El resultado de esta será el planteamiento de un sistema que satisfaga las necesidades explicadas en capítulos anteriores, es decir, la obtención de un diseño o modelo a partir del cual podamos construir el sistema.

En este apartado, comenzaremos viendo la estructura lógica del almacenamiento (diseño de la base de datos), para seguir la línea en la que acabamos el capítulo anterior con el modelo ER, los diccionarios de datos y el mapeo lógico. Posteriormente, nos centraremos en el diseño arquitectónico y el diseño de los componentes, para finalmente terminar con el diseño de la interfaz de la aplicación.

5.1. Arquitectura física

La arquitectura física de la aplicación mostrará cómo los componentes software se distribuyen entre los servidores y otros dispositivos en la red, viendo así cuáles son los componentes físicos de la aplicación a desarrollar y la relación que existe entre ellos.

En nuestro caso, tenemos una arquitectura física de cuatro capas, que se adapta perfectamente a la aplicación que estamos desarrollando. Esta arquitectura de 4 capas se podría extender idealmente a una arquitectura con más elementos, de forma que podamos garantizar que es más segura y escalable, pero por el momento, debido al tamaño y limitaciones de este proyecto, hemos optado por las cuatro capas.

En la imagen 5.1 se puede ver la arquitectura física de la aplicación, en la que vemos las cuatro capas que la componen, que son:

- **Cliente ligero:** Representa a los usuarios finales que accederán a la aplicación. Estos lo harán accediendo directamente al servidor de aplicaciones.
- **Servidor web:** los clientes ligeros realizarán peticiones HTTP desde sus navegadores web, que serán recibidas por un servidor web, que las procesará y las enviará a nuestro servidor de aplicaciones.
- **Servidor de aplicaciones:** Este servidor procesará las solicitudes entrantes, ejecutando la lógica de negocio y posiblemente devolviendo alguna información al cliente ligero en caso

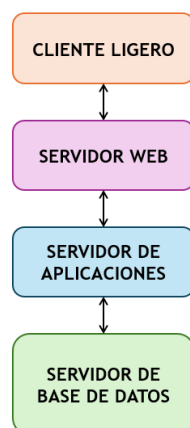


Figura 5.1: Arquitectura física de la aplicación.

de que sea necesario. Además, si es necesario, realizará consultas al servidor de base de datos para consultar la información necesaria.

- **Servidor de base de datos:** Habrá ocasiones en las que la lógica de negocio ejecutada por el servidor de aplicaciones requiera de datos que están almacenados en la base de datos. En estos casos, el servidor de aplicaciones se comunicará con este servidor para realizar las consultas y modificaciones de datos pertinentes, devolviendo desde este servidor los datos que se hayan solicitado.

Una vez conocemos la arquitectura física que hemos elegido, podemos representarla en un diagrama de despliegue, donde se puede ver cómo se comunican los distintos servidores que hemos explicado previamente. Se puede consultar este diagrama en la Figura 5.2.

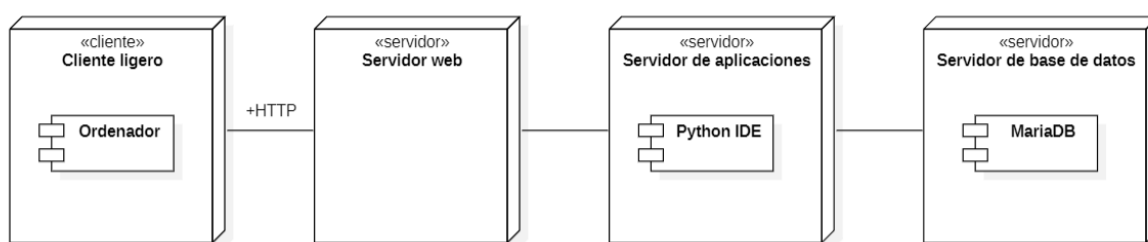


Figura 5.2: Diagrama de despliegue de la aplicación.

5.2. Arquitectura lógica

En cuanto a la arquitectura lógica, para adaptarnos a lo visto en la arquitectura física, necesitamos en este caso un modelo de 3 capas, que es el más recomendado en general y, además,

el que mejor se adapta a lo mencionado anteriormente. Gracias a este estilo arquitectónico conseguimos que el mantenimiento y la reutilización de componentes sea muy fácil, permitiendo la reutilización de la lógica de negocio. Además, los usuarios solo disponen de la interfaz gráfica, por tanto, la aplicación será sencilla de utilizar.

Las tres capas de las que se compone esta arquitectura son:

- **Capa de presentación:** En esta capa se albergan todos los componentes de la interfaz de usuario, que son los que verá el usuario al utilizar la aplicación. La capa de presentación permitirá a los usuarios realizar llamadas a los *endpoint* disponibles.
- **Capa de negocio:** La segunda capa se encarga de procesar los datos que se envían desde la capa de presentación, normalmente leídos de la interfaz cuando son introducidos por el usuario. Es la capa en la que se implementa la lógica de negocio, en este caso programada en *Python*, con la que se procesan dichos datos mencionados.
- **Capa de persistencia:** Por último, tenemos la capa encargada de almacenar y administrar los datos. En ella se encuentran los principales componentes que permiten abstraer las operaciones básicas de la base de datos.

Para representar la arquitectura lógica, se ha planteado un diagrama (ver Figura 5.3) en el que se muestra a alto nivel el planteamiento de esta arquitectura. En este diagrama, podemos encontrar las tres capas mencionadas y cómo se conectan entre ellas.

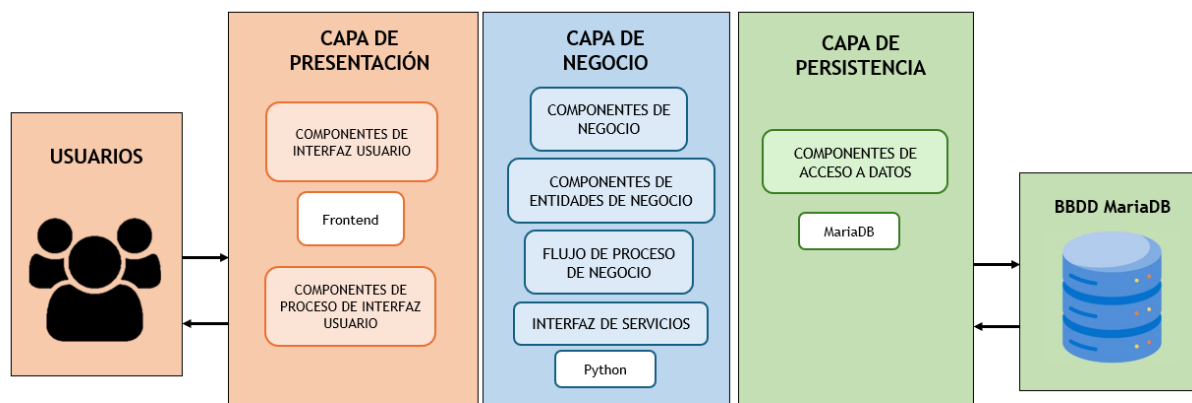


Figura 5.3: Arquitectura lógica de la aplicación.

5.3. Estructura lógica del almacenamiento

En el capítulo anterior, y en concreto en el apartado 4.6, vimos el diseño conceptual de la aplicación. Este diseño no es fácilmente extrapolable al diseño de una base de datos, por lo que es necesario transformarlo a un formato que podamos entender mejor, el diseño lógico. Con este diseño, la estructura lógica del almacenamiento quedará claramente definida, y simplificará en gran medida la construcción de la base de datos a utilizar.

El diseño lógico se refiere a la fase en la que se traduce el diseño conceptual previamente establecido en una representación más detallada y estructurada. Mientras que el diseño conceptual se centra en los requisitos y la estructura de alto nivel de la base de datos, el diseño lógico se adentra en la construcción de un modelo de datos que describa la información corporativa de acuerdo a un modelo lógico concreto, pero con independencia de cualquier consideración física. En esencia, el diseño lógico establece las bases para la creación de la base de datos real que respaldará la aplicación.

Para desarrollar el diseño lógico de nuestra base de datos hemos usado el modelo relacional, que implica la creación de una estructura de datos que describa cómo se organizarán los datos y las relaciones entre ellos en la base de datos. Para definir las relaciones entre las tablas del modelo relacional, se utilizarán los conceptos de clave primaria (PK) y clave foránea (FK) en las tablas. Las claves primarias son atributos únicos que identifican de manera única a cada fila en una tabla, mientras que las claves foráneas establecen vínculos entre tablas relacionadas.

En cuanto a la notación utilizada, destacaremos que los nombres de las claves primarias se representarán como un atributo subrayado, pudiendo haber varios atributos combinados como claves primarias. Las claves primarias aparecerán en el primer lugar de los atributos. Los nombres de las claves foráneas se declaran justo abajo de la tabla y se añaden al final, junto a los atributos de la tabla.

A continuación se muestra el diseño lógico del proyecto, ordenado alfabéticamente según el nombre de las tablas, obtenido a partir del diseño conceptual mostrado en el apartado 4.6 de este mismo trabajo.

- ACCIÓN (ID, Título, Descripción, FechaIni, FechaFin, Carga, Moodle, Trello, Curso, Sprint)
 - FK: (Curso, Sprint) → SPRINT (Curso, Número)
- ACCIÓN_APRENDIZAJE (ID, Tipo)
 - FK: ID → ACCIÓN (ID)
- ACCIÓN_EVALUACIÓN (ID, Alcance, Obligatoria)
 - FK: ID → ACCIÓN (ID)
- AUTOEVALUAR_RESPUESTA (Estudiante, Respuesta, Criterio, Calidad, Estado)
 - FK: (Estudiante, Respuesta) → RESPUESTA (Estudiante, ID)
 - FK: Criterio → CRITERIO (ID)
- CRITERIO (ID, Código, Título, Especificación, Historia)
 - FK: Historia → HISTORIA (ID)
- CURSO (Año)
- ENSEÑAR_CRITERIO (Acción_aprendizaje, Criterio)
 - FK: Acción_aprendizaje → ACCIÓN_APRENDIZAJE (ID)
 - FK: Criterio → CRITERIO (ID)
- ESTUDIANTE (NIA, Nombre, Fnac, Género)
- EVALUAR_RESPUESTA (Estudiante, Respuesta, Criterio, Calidad, Estado)
 - FK: (Estudiante, Respuesta) → RESPUESTA (Estudiante, ID)
 - FK: Criterio → CRITERIO (ID)

- HISTORIA (ID, Título, Especificación, Objetivo)
 - FK: Objetivo → OBJETIVO (ID)
- HISTORIA_EN_ACCIÓN (Acción, Historia)
 - FK: Acción → ACCIÓN (ID)
 - FK: Historia → HISTORIA (ID)
- INCLUIR_PREGUNTA (Acción, Plantilla, Pregunta, Orden)
 - FK: Acción → ACCIÓN_EVALUACIÓN (ID)
 - FK: (Plantilla, Pregunta) → PREGUNTA (Plantilla, ID)
- MATRICULAR_ESTUDIANTE (Estudiante, Curso, Calif_o, Nota_o, Calif_e, Nota_e, Número)
 - FK: Estudiante → ESTUDIANTE (NIA)
 - FK: Curso → CURSO (Año)
- MEDIR_CRITERIO (Plantilla, Criterio, Umbral, Ponderación, Especificación)
 - FK: Plantilla → Plantilla (ID)
 - FK: Criterio → CRITERIO (ID)
- OBJETIVO (ID, Título, Especificación)
- PLANIFICAR_SPRINT (Curso, Sprint, Historia)
 - FK: (Curso, Sprint) → SPRINT (Curso, Número)
 - FK: Historia → HISTORIA (ID)
- PLANTILLA (ID, Nombre, Tipo)
- PREGUNTA (Plantilla, ID, Título, Descripción, Solución)
 - FK: Plantilla → PLANTILLA (ID)
- RESPUESTA (Estudiante, ID, Fecha, Contenido, Apto, Evaluación, Autoevaluación, Plantilla, Pregunta)
 - FK: Estudiante → ESTUDIANTE (NIA)
 - FK: (Plantilla, Pregunta) → PREGUNTA (Plantilla, ID)
- RESULTADO (Estudiante, ID, Fecha, Contenido)
 - FK: Estudiante → ESTUDIANTE (NIA)
- SPRINT (Curso, Número, Finicio, Ffin)
 - FK: Curso → CURSO (Año)
- SUPERAR_CRITERIO (Estudiante, Resultado, Criterio)
 - FK: (Estudiante, Resultado) → RESULTADO (Estudiante, ID)
 - FK: Criterio → CRITERIO (ID)
- TAREA (Estudiante, ID, Fecha, Tiempo, Comentario, Acción)
 - FK: Estudiante → ESTUDIANTE (NIA)
 - FK: Acción → ACCIÓN (ID)

- VALORAR_EVALUACIÓN (Estudiante, Resultado, Acción_evaluación, Evaluación)
 - FK: (Estudiante, Resultado) → RESULTADO (Estudiante, ID)
 - FK: Acción_evaluación → ACCIÓN_EVALUACIÓN (ID)

5.4. Interfaz de usuario

Con el objetivo de definir la estructura visual y la experiencia de uso del sistema, se han desarrollado una serie de bocetos de la interfaz de usuario. Estos esquemas nos permiten tener una primera visión de cuál será la disposición de los elementos en pantalla, facilitando así la comprensión del flujo de interacción con la aplicación. Esto nos sirve, además, como base para la posterior implementación que desarrollaremos en el siguiente capítulo. Los bocetos se han diseñado teniendo en cuenta tanto la usabilidad como la coherencia con los requisitos establecidos previamente.

En concreto, hemos diseñado una serie de bocetos, mostrados a continuación, para las diferentes páginas que tendrá la aplicación que se va a implementar. En la imagen 5.4 encontramos la página principal de la aplicación, desde la que se podrá acceder al resto de páginas mostradas a continuación. La Figura 5.5 muestra una interfaz intermedia de selección, en la que se podrá, dependiendo del caso de uso que estemos siguiendo, elegir el curso, *sprint* o la acción, entre otras cosas. Pasando con las páginas de los casos de uso en particular, tenemos que las Figuras 5.6 y 5.7 se corresponden con los casos de uso representados en las tablas de especificación 4.2 y 4.3 respectivamente. La imagen 5.8 está directamente relacionada con los casos de uso destinados al procesamiento automático de datos, en concreto los que están representados en las especificaciones 4.15, 4.16, 4.17 y 4.18. Para terminar, tenemos otras dos grandes funcionalidades representadas en las dos imágenes restantes. Comenzando por la Figura 5.9, esta representa todas las páginas relacionadas con la gestión de los elementos que se puede hacer directamente desde el sistema a desarrollar, permitiendo realizar todas las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre estos elementos. Por último, la imagen 5.10 muestra una representación visual del estilo del *dashboard* que se quiere construir.

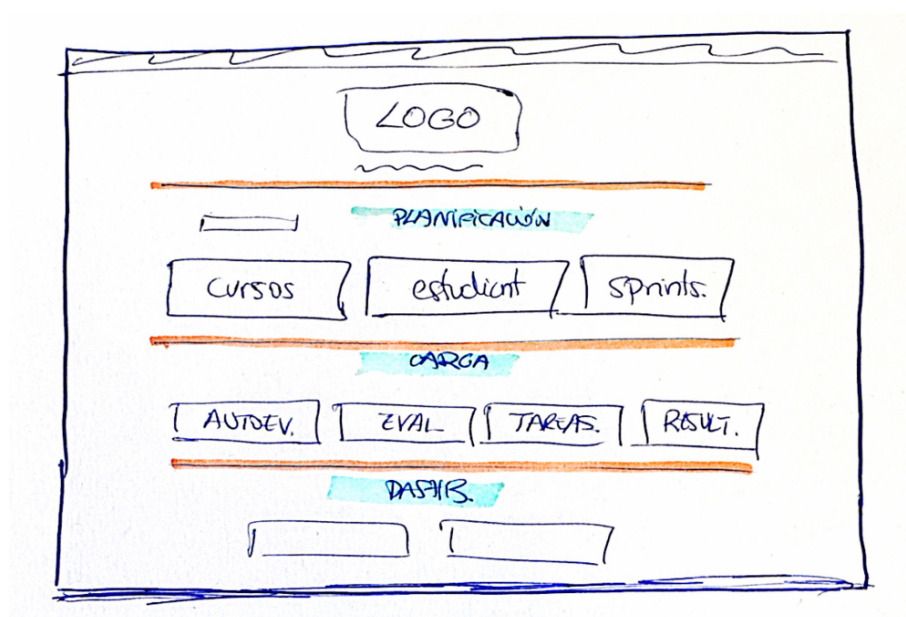


Figura 5.4: Página principal de la aplicación.

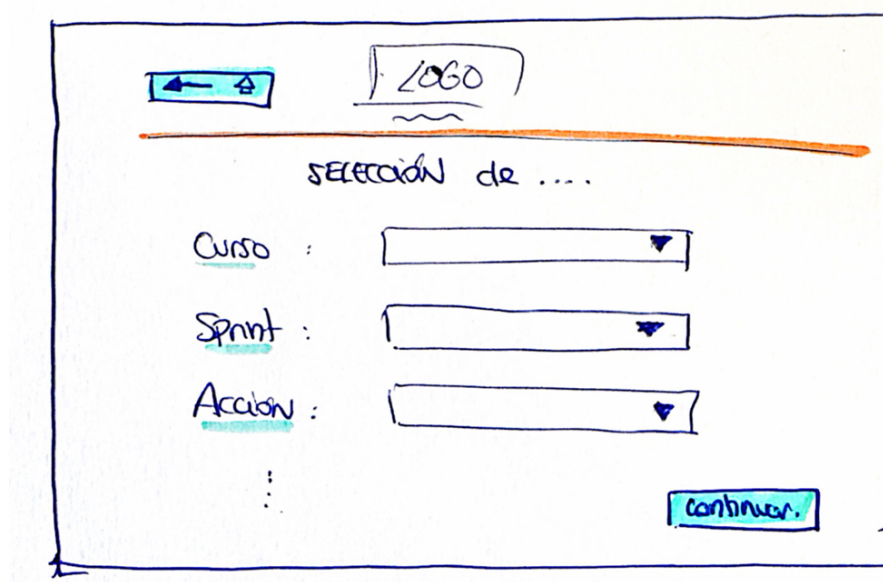


Figura 5.5: Página de selección previa.

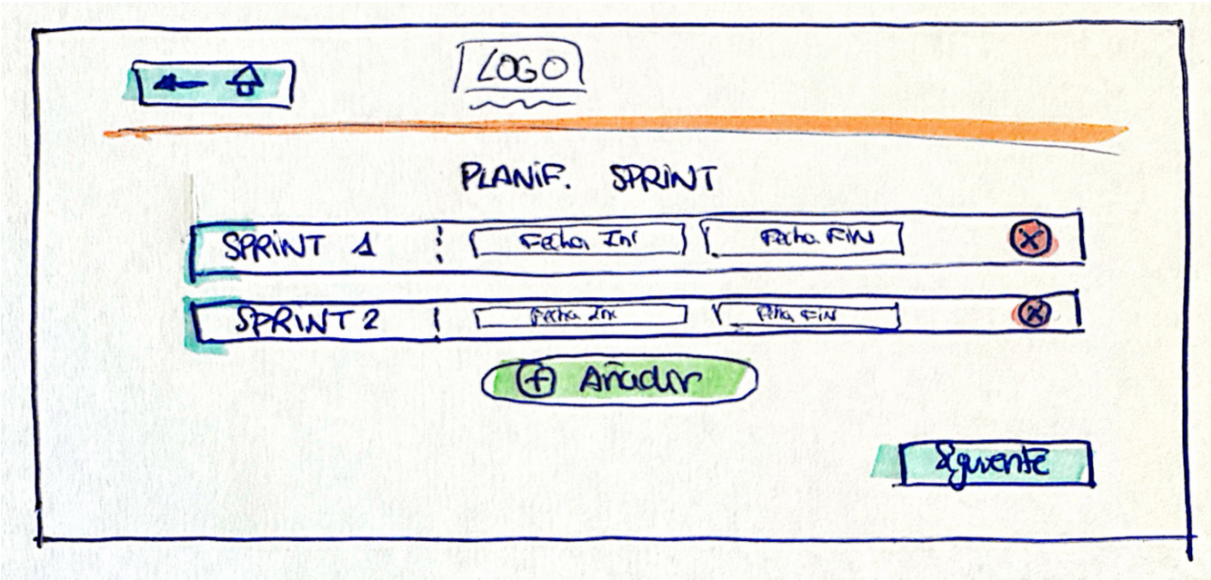


Figura 5.6: Página de creación de *sprints*.

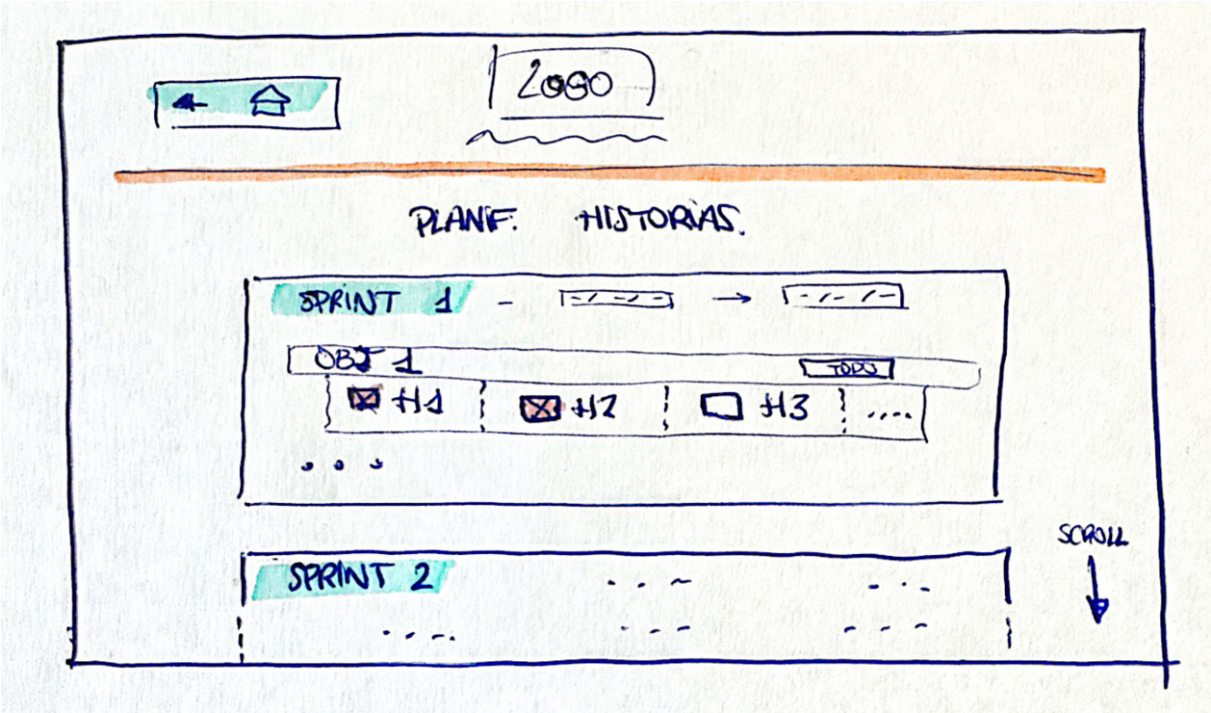


Figura 5.7: Página para la planificación de historias en *sprints*.

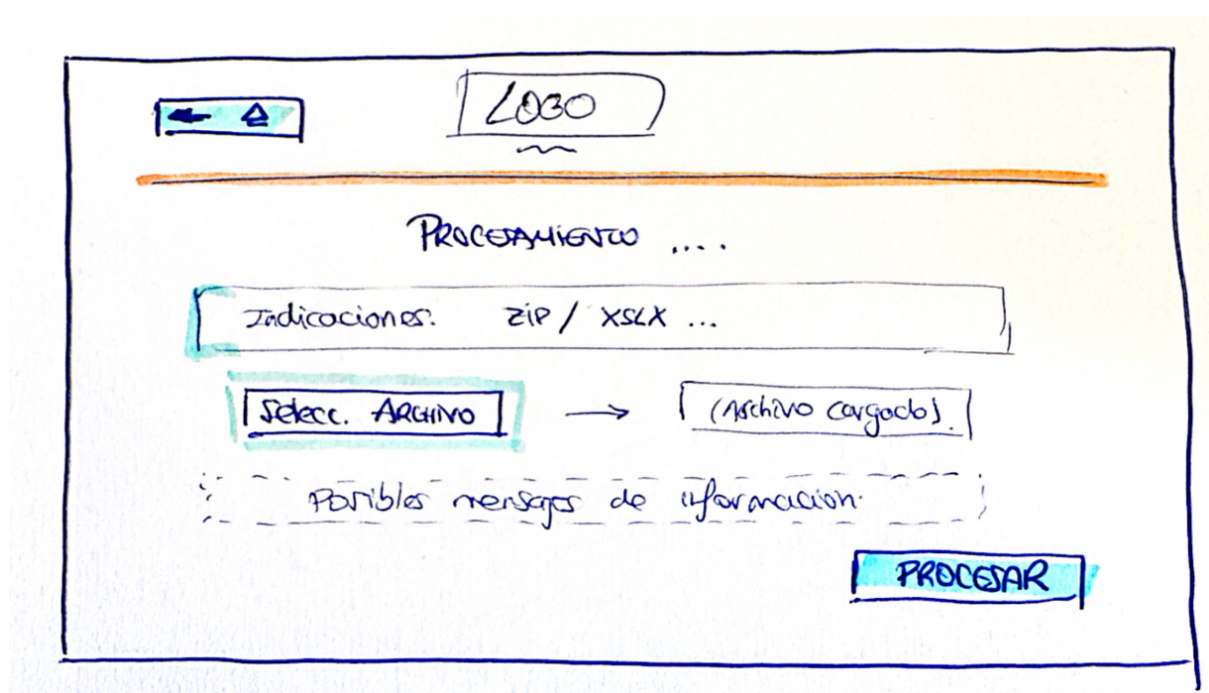


Figura 5.8: Página para la carga y procesamiento de archivos.

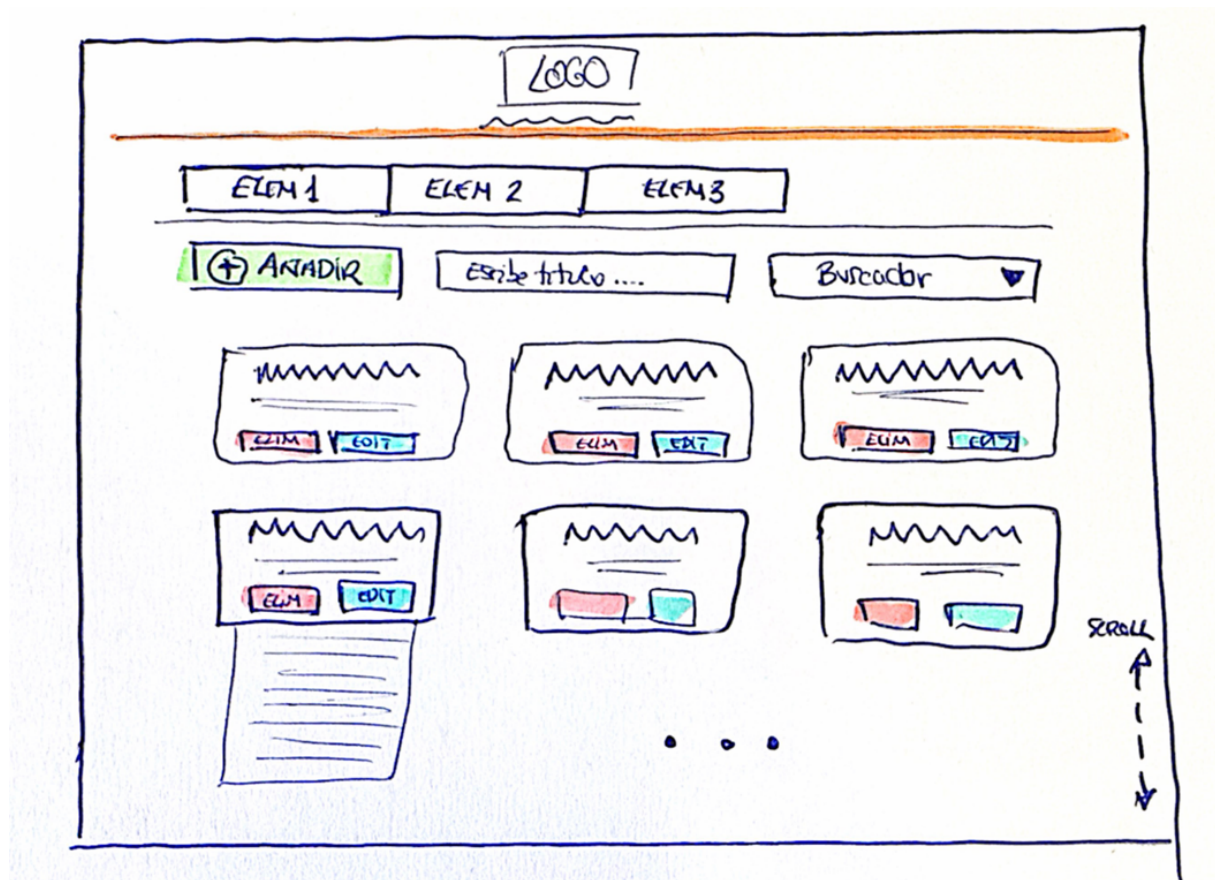
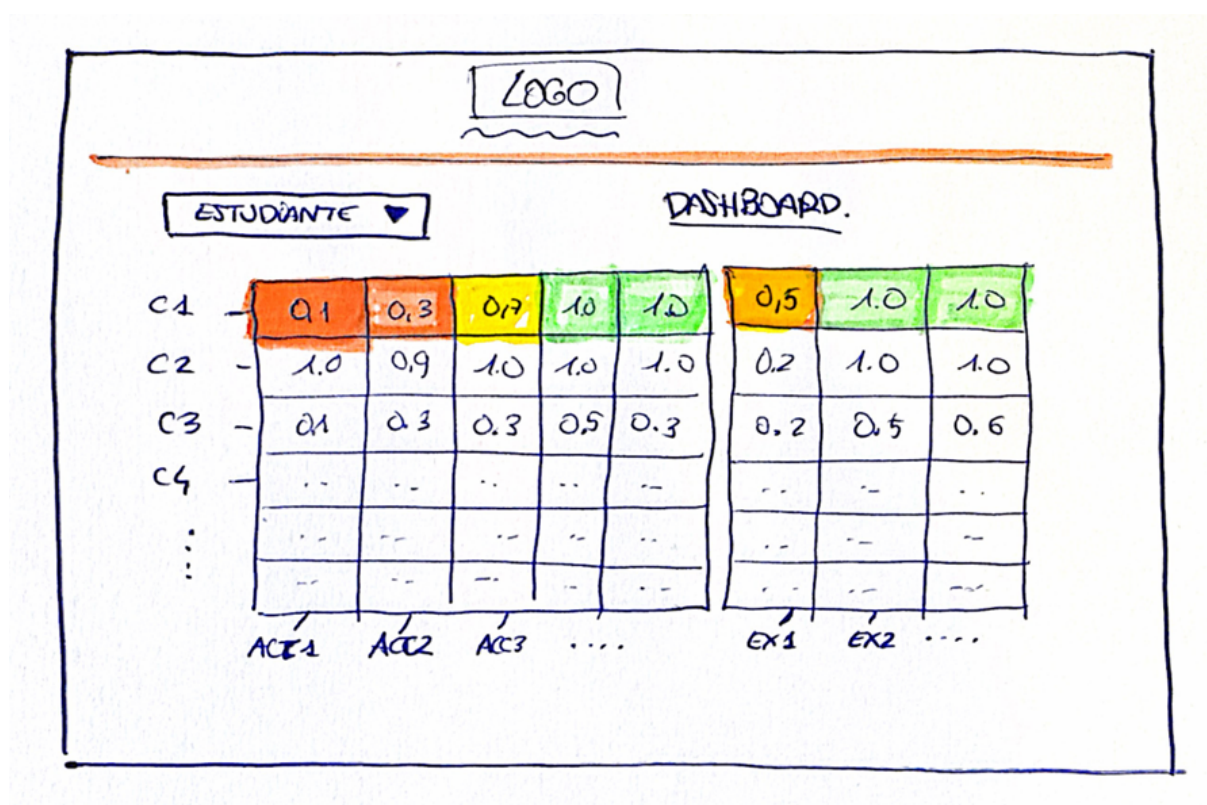


Figura 5.9: Página para la visualización de los elementos del producto y proceso de aprendizaje.

Figura 5.10: Página de visualización de los *dashboards*.

Capítulo 6

Implementación

En este capítulo se detalla el proceso de implementación de la solución desarrollada en este Trabajo Fin de Grado. La implementación se ha basado en los fundamentos técnicos presentados en el Capítulo 3, donde se analizaron las diferentes alternativas tecnológicas disponibles para cada componente del sistema. Tras evaluar las opciones, se seleccionaron las herramientas que mejor se adaptaban a los requisitos del proyecto, equilibrando rendimiento, facilidad de desarrollo y escalabilidad.

Nuestra aplicación, como ya hemos visto anteriormente también con la explicación sobre las arquitecturas lógica y física (ver apartados 5.1 y 5.2), se compone de tres partes principales. El desarrollo de la parte visual de la aplicación (*frontend*) con la que interactuarán los usuarios, la lógica (*backend*) a la que se conecta este frontal y la base de datos en la que se almacenarán los datos procesados. Por ello, hemos tenido que realizar tres elecciones para estas tres partes de nuestra aplicación, con lo que hemos completado la selección del entorno en el que vamos a desarrollar el proyecto. La elección realizada sobre las tecnologías disponibles se detalla en el apartado 6.1 de este capítulo.

Por otra parte, la implementación de este TFG ha tenido dos grandes partes muy diferenciadas. La primera de ellas, ha estado relacionada con la preparación de los datos que posteriormente se han utilizado en la aplicación. Este trabajo, aunque es menos visual, ha tenido una gran importancia en este proyecto, ya que era necesario hacer aterrizar todos los conceptos de *Test-Driven Learning* y poder unirlos para brindar una coherencia que pudiera ser representada en una base de datos. Una vez disponibilizados los datos, cuando también hayamos comprobado la validez de las fuentes de datos utilizadas, podremos continuar con la segunda parte del proyecto, que ha sido la creación de los servicios que automatizan el procesamiento de estos datos, permitiendo así que cualquier profesor sin conocimientos de programación pueda gestionar y visualizar toda la información relacionada con su asignatura desde la aplicación que hemos desarrollado. La explicación de la implementación de estos dos pasos, se encuentra en los apartados 6.2 y 6.3 del capítulo actual.

6.1. Herramientas

Comenzando por el *frontend*, habíamos presentado varias opciones en los fundamentos teóricos, y, finalmente, nos hemos decantado por el uso de *HTML*, *CSS* y *JavaScript* básico. Esta elección se ha basado en su sencillez a la hora de utilizarlo, ya que no es necesario un entorno muy

complejo para desarrollar esta aplicación. La elección de estas tecnologías se debe principalmente a que son herramientas con las que ya se contaba con conocimientos previos adquiridos, lo que ha permitido acelerar el desarrollo sin necesidad de invertir tiempo en aprender nuevos *frameworks* o lenguajes.

Dado que el objetivo principal del trabajo se centra en el procesamiento y análisis de datos, se ha decidido priorizar el desarrollo de un proceso de ETL que maneje todo tipo de excepciones en el *backend*, asegurándonos así de que no se ingestan datos erróneos en la base de datos. Al ser el procesamiento un aspecto más crítico, se ha optado, como hemos comentado, por tecnologías web estándar y ampliamente documentadas, logrando así un equilibrio entre funcionalidad y esfuerzo, garantizando que el tiempo disponible se destinara a los aspectos clave del proyecto.

El segundo de los puntos técnicos que queremos desarrollar es la forma en la que se ha realizado la programación de la lógica de la aplicación en el *backend*. Entre todas las tecnologías disponibles que habíamos presentado en el apartado 3.4, se ha optado por el lenguaje *Python*. La elección de este lenguaje de programación se basa en su simplicidad, legibilidad y amplia adopción en tareas de procesamiento de datos. Una de las razones principales de esta elección es que *Python* cuenta con un ecosistema muy completo de librerías especializadas, que permiten construir soluciones robustas en tiempos reducidos, integrando con facilidad operaciones de lectura, transformación y almacenamiento de datos. En nuestro caso, estas herramientas nos ayudan, entre otras cosas, a leer los datos de las hojas de cálculo y procesarlos de forma simultánea, acelerando así el proceso de ETL mencionado anteriormente.

Entre estas herramientas, destacan especialmente las siguientes:

- **pandas**: Esta es una de las librerías más utilizadas para análisis y manipulación de datos estructurados. Como ya comentamos en los fundamentos teóricos, nos proporciona estructuras como los *DataFrame*, que permiten representar datos en forma de tablas, y facilita operaciones como el filtrado, agrupamiento, limpieza y transformación de grandes volúmenes de datos.
- **openpyxl**: Esta segunda librería destacada está orientada al trabajo con archivos Excel en formato *.xlsx*, fundamental para automatizar la fase de extracción de este proyecto. Con ella, podemos leer las hojas de cálculo, accediendo a los valores almacenados en celdas específicas, necesarios para poder posteriormente cargarlos en la base de datos.
- **Flask**: La librería Flask, como ya habíamos comentado previamente, permite crear de forma rápida un sistema de *endpoints* que actúa como intermediario entre el *frontend* y la base de datos, gestionando las peticiones HTTP y facilitando la comunicación entre componentes.

El conjunto de estas herramientas hace posible un flujo de trabajo fluido, modular y escalable, que integra desde la entrada de datos hasta su exposición en una aplicación web de forma coherente y mantenible.

Una vez definidas las tecnologías que queremos utilizar para el *frontend* y el *backend*, podemos elegir un tipo de bases de datos que, además de ser compatible con las tecnologías utilizadas, pueda adaptarse para almacenar la información que se va a generar con el procesamiento de los documentos disponibles. En nuestro caso, se ha optado por una base de datos relacional, por cómo hemos planteado el diseño del proyecto en el capítulo anterior. Entre las bases de datos relacionales, la opción que mejor se adapta por su conectividad con *Python* y la sencillez de uso es una base de datos MariaDB. Además, la Universidad de Valladolid ya dispone de varios

servidores con estas bases de datos desplegadas, que junto con la interfaz de *phpMyAdmin* son muy simples de utilizar y se adaptan perfectamente a lo que se busca para este proyecto.

6.2. Preparación de los datos

En la introducción comentamos que habíamos dividido la implementación en dos partes muy diferenciadas. En este apartado, nos centraremos en la primera de estas fases de la implementación. Para comenzar la implementación, ha sido necesario materializar todos los conceptos abstractos que componen la metodología *Test-Driven Learning*, explicada en el apartado 3.1.1, en un modelo de datos operativo. Este proceso ha requerido un esfuerzo iterativo de análisis y refinamiento, ya que la información relevante estaba dispersa en múltiples documentos y formatos no estandarizados, por lo que era necesario diseñar una forma de agrupar toda esta información en un mismo modelo de dominio. Durante estas iteraciones, gracias a las reuniones con los profesores responsables de esta metodología, se pudieron identificar inconsistencias entre la información de la metodología y lo que se había representado en el modelo, por lo que pudimos solucionar los problemas en fases tempranas de la implementación.

Una vez solucionados los problemas iniciales y modelado el núcleo conceptual de TDL, se pasó con el análisis de los datos disponibles en las fuentes de datos. En las primeras revisiones de las fuentes, se detectaron algunas omisiones de datos (como las relaciones entre tareas y acciones), que no estaban representadas en la información disponible obtenida de las fuentes de datos, pero que eran necesarias para modelar la metodología de forma completa. Además, se identificaron otros elementos que, al contrario, estaban presentes en los documentos que contenían los datos pero no habíamos representado en el modelo, aunque supusieron cambios menores que no tuvieron mayor significancia en el trabajo realizado. Un ejemplo ilustrativo fue la vinculación de tareas con acciones; al no estar explícita en los datos brutos, fue necesario inferirla mediante reglas de negocio derivadas de la experiencia de los profesores que han trabajado con esta metodología.

Una vez consolidado el modelo, se realizó un análisis exhaustivo de la calidad de los archivos que serían las fuentes de datos de nuestro procesamiento. Este examen reveló la presencia de algunos problemas, ya fueran causados por la interacción de los alumnos con estos documentos o por inconsistencias puntuales que presentan los documentos. Presentaremos a continuación los principales problemas encontrados y como se ha diseñado una solución para cada uno de ellos:

■ Inconsistencias en archivos de TAREAS (cuadernos de trabajo)

- **Campos vacíos:** Los alumnos que interactúan con los cuadernos de trabajo, en ocasiones omiten valores obligatorios, requiriendo lógicas de imputación basadas en valores por defecto para evitar estas inconsistencias.
- **Incoherencias en valores:** Algunos estudiantes no completan algunos campos con los valores obligatorios indicados en los desplegables, causando así desviaciones de los dominios predefinidos. Estas incoherencias se han solucionado mediante un mapeo utilizando las fechas de realización de las tareas y como estas mapeaban con la acción correspondiente para esos días.
- **Errores de formato:** Las fechas de las tareas tenían en ocasiones formatos incorrectos o valores no válidos (por ejemplo, días inexistentes como el 30 de febrero). Para solucionar este problema se realizó un preprocesamiento con expresiones regulares, y corrigiendo aquellas fechas incorrectas a la más cercana con un valor adecuado.

■ Problemas en AUTOEVALUACIONES

- **Estructura variable:** Los archivos de autoevaluación no siempre tenían la misma estructura, concretamente en la parte de los criterios. En algunos de estos archivos, los valores de los códigos no seguían el estándar. La medida tomada ha sido transformar cada uno de estos códigos desactualizados en la versión estandarizada de los mismos.
- **Falta de trazabilidad:** Algunos archivos de alumnos carecían de los nombres de las preguntas que evaluaban como parte del nombre del archivo. Esto ha supuesto una revisión exhaustiva que permita garantizar la trazabilidad de los archivos con las preguntas que evalúan.
- **Versiones desactualizadas:** Los alumnos, en ocasiones, utilizan de versiones de las autoevaluaciones antiguas e incompatibles con el esquema actual. Estas evaluaciones se han descartado ya que los profesores tampoco las consideran cuando evalúan en las asignaturas que siguen la metodología de *eXtreme Learning*.

Estos descubrimientos nos subrayan la importancia de, en el futuro próximo, estandarizar todos los archivos utilizados para la evaluación y seguimiento de los resultados de los alumnos en las asignaturas que apliquen la metodología de *Test-Driven Learning*, de forma que se asegure una mayor consistencia de los datos a la hora de procesarlos para futuros cursos. Pese a las dificultades, la corrección progresiva de los datos, combinando intervención manual para casos excepcionales y reglas automatizadas, nos ha permitido finalmente obtener un conjunto limpio y estructurado de datos, que ha supuesto la base fundamental para los servicios descritos en la Sección 6.3.

6.3. Servicios programados

A la hora de realizar la programación de este TFG, se ha querido seguir la línea del desarrollo profesional de software, en el que resulta esencial organizar el código de forma modular. Esta estructura responde a principios fundamentales de la ingeniería del software, como la separación de responsabilidades, la alta cohesión dentro de cada módulo y el bajo acoplamiento entre ellos. El objetivo principal es que cada componente del sistema tenga una función clara y delimitada, lo cual facilita el mantenimiento, la comprensión y la escalabilidad del código. Conseguimos con esto que, en caso de que se necesite incorporar alguna mejora a la aplicación, esta sea sencilla de integrar, gracias a la estructura clara que facilita a los nuevos desarrolladores entender en qué partes se debe incluir el código necesario para incorporar las nuevas funcionalidades.

Una arquitectura modular permite dividir el proyecto en carpetas y archivos organizados por función o propósito. Para poder acceder al código así organizado, es habitual encontrar un punto de entrada principal al programa, desde el cual se coordinan todos los procesos. En concreto, en nuestro caso, se ha utilizado *Flask* para programar el *backend* de la aplicación, por tanto, la orquestación de los procesos se realizará mediante una serie de *endpoints*, que realizarán diferentes funciones según la utilidad que tengan. El resto de la aplicación estará organizada en diferentes módulos:

- **Módulo de almacenamiento:** Este módulo nos permitirá mantener organizados todos los archivos que se generan con la metodología, almacenados de forma clasificada por asignatura, curso, *sprint*, etc. De esta forma, es sencillo para los usuarios que utilicen la aplicación encontrar los datos que se utilizarán posteriormente para el procesamiento.

- **Módulo de configuración:** Las configuraciones del sistema, tales como rutas de archivos, parámetros de conexión o constantes del dominio, se externalizan en ficheros independientes, a menudo en formatos como YAML o JSON, lo que permite su modificación sin alterar el código fuente.
- **Módulo principal:** El procesamiento de datos, si está presente, suele organizarse en una o varias capas según su naturaleza. Por ejemplo, pueden existir módulos encargados de la conexión con fuentes de datos externas, módulos intermedios que implementan la lógica de negocio o de transformación, y otros dedicados a tareas auxiliares o reutilizables. Esta separación favorece el desarrollo iterativo y la integración de nuevas funcionalidades, ya que cada parte del sistema puede evolucionar de forma independiente, sin generar efectos colaterales en el resto.

La parte más importante de esta estructura modular es el módulo principal, y dentro de él, la estructura de *endpoints* que se ha realizado. Esta parte del *backend*, desarrollada en *Flask* se organiza en cuatro categorías principales de *endpoints*, cada una diseñada para cumplir con un propósito específico dentro de la arquitectura del sistema. Para cada uno de los tipos, se dará una pequeña descripción y las características más importantes, además de un ejemplo de las partes más importantes del código, que no está completa en ninguno de los casos.

Endpoints de Procesamiento ETL

Estos *endpoints* (*/tareass*, */autoevaluaciones*, */evaluaciones*, etc.) orquestan el flujo completo de Extracción, Transformación y Carga de datos. Reciben archivos Excel (mediante uploads o rutas predefinidas), aplican validaciones y persisten la información estructurada en la base de datos MariaDB. Por ejemplo:

```

1 @app.route('/tareass', methods=['GET'])
2 def tasks_endpoint():
3     db_config = set_database_config()
4     run_tasks(db_config, ...)
```

Listado 6.1: Ejemplo de endpoint ETL

En estos *endpoints* se hace inicialmente una validación de los datos recibidos por los parámetros HTTP. Posteriormente, se ejecuta un proceso particular para cada *endpoint* que se encarga de leer, transformar y cargar los datos en la base de datos. En concreto, en el ejemplo mostrado en 6.1, vemos como el endpoint definiría la configuración general de la base de datos utilizando una función, y posteriormente llama a otra que se encarga de este procesamiento.

Endpoints CRUD para Gestión de Metadatos

Hay otras rutas como */objetivos*, */historias* o */criterios* disponibles que implementan operaciones Crear, Leer, Actualizar o Eliminar (CRUD) para los elementos fundamentales de la metodología.

Estos *endpoints* tienen una lógica que permite gestionar todos los problemas que puede suponer una inserción, actualización o eliminación en la base de datos. En caso de que el procedimiento de error, este se devolverá correctamente formateado de forma que el usuario pueda entender la causa del error que está obteniendo. Entre otras cosas, cabe destacar que se gestiona una eliminación en cascada controlada (por ejemplo, no se borra un objetivo con historias asociadas).

```

1 @app.route('/objetivos/<int:objetivo_id>', methods=['GET', 'PUT', 'DELETE'])
2 def eliminar_objetivo(objetivo_id):
3     if request.method == 'GET':
4         run_query("SELECT * FROM OBJETIVO WHERE ID = %s", ...)
5     elif request.method == 'PUT':
6         execute_query("UPDATE OBJETIVO SET ... WHERE ID = %s", ...)
7     elif request.method == 'DELETE':
8         run_query("DELETE FROM OBJETIVO WHERE ID = %s", ...)

```

Listado 6.2: Ejemplo de endpoint CRUD

En este caso, el ejemplo anterior de 6.2 está planteado de forma que el mismo *endpoint* se pueda utilizar para tres acciones, en este caso la consulta, actualización o eliminación de los datos.

Endpoints de Consulta para Frontend

Estos *endpoints* se utilizan habitualmente en el *frontend*, ya que proporcionan datos filtrados para alimentar los componentes visuales del mismo. Algunos ejemplo que podemos encontrar en el código son `/cursos_disponibles` o `/sprints_disponibles`. Un ejemplo podría ser el siguiente:

```

1 @app.route('/cursos_disponibles', methods=['GET'])
2 def cursos_disp_endpoint():
3     db_config = set_database_config()
4     find_cursos(db_config, ...)

```

Listado 6.3: Ejemplo de *endpoint* de consulta para *frontend*

El funcionamiento de 6.3 es muy similar al primero explicado, estableciendo primero la configuración de la base de datos, para posteriormente ejecutar una función en la que se utiliza dicha configuración para ejecutar las consultas sin problema.

Endpoints de Gestión de Archivos

Los últimos *endpoint* que se quieren mostrar son los preparados para la carga de archivos que pueda subir el profesor que utilice la aplicación. En este tipo de *endpoints*, como por ejemplo el caso de `/upload_autoevaluaciones`, se maneja la subida y organización de archivos ZIP o XLSX en el sistema de almacenamiento y clasificación de archivos que incorpora la aplicación. Un ejemplo podría ser el siguiente:

```

1 @app.route('/upload_autoevaluaciones', methods=['POST'])
2 def upload_autoevaluaciones():
3     file = request.files['file']
4     with zipfile.ZipFile(file) as zip_ref:
5         zip_ref.extractall(dest_dir)

```

Listado 6.4: Ejemplo de gestión de archivos

Todos los *endpoints* preparados para recibir archivos tienen una estricta lógica de validación detrás, ya que tienen que asegurar que los archivos que se van a cargar (y que serán los que posteriormente se procesen) tienen el formato adecuado y son aptos para el posterior procesamiento. En este proceso hemos tenido que lidiar con varias situaciones que se podrían dar, como el manejo de rutas demasiado largas en Windows, ya que los archivos que se deben subir, descargados directamente de Moodle, contienen nombres de carpetas y de archivos muy largos. En el ejemplo

resumido mostrado en 6.4, vemos como se realizaría la extracción de un fichero “zip” para poder utilizar todos los archivos que contiene en su interior.

6.4. Construcción del dashboard

Para terminar esta sección, vamos a comentar el proceso que se ha seguido para implementar el *dashboard* que se muestra en la aplicación. El sistema se apoya en un *endpoints* de tipo consultas para el *frontend* que ejecuta una consulta compleja a la base de datos para recuperar los resultados de autoevaluaciones y evaluaciones del estudiante indicado (o la media de las mismas en caso de consultar toda la información agregada), y los metadatos necesarios para contextualizar estos resultados. Para el caso concreto en el que se selecciona un estudiante en particular, tenemos que se ha utilizado la siguiente consulta:

```
SELECT      a.Título AS Acción,
            pr.Título AS Pregunta,
            cr.Codigo,
            ar.Calidad AS Calidad_autoeval,
            er.Calidad AS Calidad_eval,
            mc.Umbral
FROM        ESTUDIANTE AS e
JOIN        RESPUESTA AS r
ON          e.NIA = r.Estudiante
JOIN        PREGUNTA AS pr
ON          (r.Plantilla, r.Pregunta) = (pr.Plantilla, pr.ID)
JOIN        INCLUIR_PREGUNTA AS ip
ON          (r.Plantilla, r.Pregunta) = (ip.Plantilla, ip.Pregunta)
JOIN        ACCIÓN AS a
ON          ip.Acción = a.ID
JOIN        AUTOEVALUAR_RESPUESTA AS ar
ON          (r.Estudiante, r.ID) = (ar.Estudiante, ar.Respuesta)
LEFT JOIN   EVALUAR_RESPUESTA AS er
ON          (r.Estudiante, r.ID, ar.Criterio) = (er.Estudiante, er.Respuesta, er.Criterio)
JOIN        CRITERIO AS cr
ON          ar.Criterio = cr.ID
JOIN        MEDIR_CRITERIO AS mc
ON          (ip.Plantilla, ar.Criterio) = (mc.Plantilla, mc.Criterio)
WHERE       e.NIA = %s
AND         aCurso = %s
AND         (er.Criterio IN (SELECT ID FROM CRITERIO AS c WHERE c.Historia = %s)
            OR ar.Criterio IN (SELECT ID FROM CRITERIO AS c WHERE c.Historia = %s))
ORDER BY    a.FechaIni, pr.Título, ar.Criterio
```

En la consulta anterior, vemos cómo se seleccionan específicamente los títulos de acciones y preguntas para contextualizar los datos, los códigos de criterios para identificar las competencias evaluadas, y los valores numéricos de autoevaluación (ar.Calidad) y evaluación (er.Calidad) para permitir, en los casos en los que existan ambas, su comparación directa. Se incluye además el umbral de aprobación (mc.Umbral) como referencia para interpretar los resultados.

La construcción de los JOIN parte del estudiante, pasando desde ahí hacia sus respuestas y preguntas asociadas, garantizando así que solo se muestran los datos de las acciones en la que

ha participado el estudiante. Posteriormente, se une también la información de las acciones y los criterios que se han evaluado en ellas. El uso de `LEFT JOIN` con la evaluación de la respuesta garantiza que no se pierdan las autoevaluaciones de ejercicios que no hayan sido evaluados por el profesor. Los filtros aplicados en la cláusula `WHERE` cumplen tres propósitos esenciales: el NIA garantiza que el dashboard es concreto para cada estudiante, el curso contextualiza los resultados en el marco temporal adecuado, y la historia de aprendizaje focaliza el análisis en algunas acciones y criterios específicos. Esta estructura permite al *dashboard* mostrar no solo una foto del estado actual, sino también la progresión temporal mediante el ordenamiento por fechas de acción, permitiendo así que el profesor evalúe el progreso de los estudiantes a lo largo de las acciones de evaluación realizadas.

Capítulo 7

Pruebas

Las pruebas nos permiten probar la robustez de la aplicación una vez hemos finalizado su implementación. Para ello, estudiaremos los casos en los que un profesor que utilice la aplicación pueda introducir datos erróneos, analizando cómo se comporta la aplicación en esos casos.

Cabe destacar que esta aplicación es una prueba de concepto enfocada específicamente en la visualización todos los datos que ya tenemos cargados de la asignatura que hemos utilizado para las pruebas. Por esta razón, hay algunos aspectos como pueden ser las limitaciones a la hora de elegir fechas para los cursos o el formato que tienen que tener los códigos de los criterios, que no se han contemplado en la aplicación y, por ello, no se han incluido en los caso de prueba. Por ello, en este capítulo nos centramos en los puntos de la aplicación en los que se podrían introducir datos que sean obviamente incorrectos en cualquier contexto, además de la verificación de tipos de archivos cargados, que nos permite asegurar el correcto funcionamiento de la aplicación.

Vamos a dividir las pruebas en varios grupos según el tipo de dato que evalúen. Veremos pruebas realizadas a la gestión de fechas, a la carga de archivos y a la eliminación de datos que tengan otros asociados en la base de datos.

7.1. Pruebas con fechas

En esta sección se muestran las pruebas realizadas para asegurarse que las fechas introducidas en la aplicación son válidas y coherentes con lo definido. Las siguientes tablas e imágenes muestran las pruebas realizadas y los resultados obtenidos para cada una de ellas.

P-01	Fechas iniciales posteriores a las finales
Objetivo	Comprobar que el sistema no permite continuar con la creación de curso o de <i>sprint</i> si la fecha de inicio que ha indicado es posterior a la fecha de finalización.
Precondición	El usuario se encuentra en el flujo de crear curso.
Pasos	1. Completar los campos para crear un nuevo curso o un nuevo sprint. 2. Introducir una fecha de inicio que sea posterior a la de finalización. 3. Pulsar “Continuar”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que la fecha de inicio no puede ser posterior a la de finalización.
Resultado obtenido	CORRECTO (ver Figuras 7.1 y 7.1 para ver el mensaje de error).

Tabla 7.1: Especificación de la prueba 1.

Gestión de Curso

Crear Nuevo Curso

Curso (formato 0000-0000):
2024-2025

Fecha inicio (formato 00-00-0000):
05-01-2024

Fecha finalización (formato 00-00-0000):
01-01-2024

Continuar

La fecha de inicio no puede ser posterior a la de finalización.

[← Volver al inicio](#)

Figura 7.1: Error obtenido al introducir una fecha de inicio posterior a una de finalización al crear un curso.

Gestión de Curso

[◀ Atrás](#)

Gestionar Sprints

Sprint 1

Fecha inicio:
14-09-2024

Fecha finalización:
12-09-2024

+ Añadir Sprint

Continuar

Corrige los siguientes errores:
Sprint 1: La fecha de inicio no puede ser posterior a la de finalización

Figura 7.2: Error obtenido al introducir una fecha de inicio posterior a una de finalización al crear un sprint.

P-02	Fechas de sprint fuera del curso
Objetivo	Comprobar que el sistema no permite continuar con la creación de un <i>sprint</i> si la fecha de inicio o de finalización que ha indicado el usuario es externa a las indicadas para el curso.
Precondición	El usuario se encuentra en el flujo de crear curso y ha seleccionado una fecha de inicio y fin para dicho curso.
Pasos	1. Completar los campos para un nuevo <i>sprint</i>. 2. Introducir una fecha de inicio o finalización externa a los límites marcados por las introducidas para el curso. 3. Pulsar “Continuar”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que alguna de las fechas introducidas no están contenidas en el rango indicado por el curso, mostrando la fecha errónea y el rango en el que se pueden introducir las fechas.
Resultado obtenido	CORRECTO (ver Figura 7.3 para ver el mensaje de error).

Tabla 7.2: Especificación de la prueba 2.

The screenshot shows a web interface for adding sprints. It contains two sections, 'Sprint 1' and 'Sprint 2', each with a red 'X' icon in the top right corner. Each section has two input fields: 'Fecha inicio:' and 'Fecha finalización:'. Below these sections is a green button labeled '+ Añadir Sprint'. At the bottom, there is a green button labeled 'Continuar' and a red-bordered box containing error messages.

Sprint 1 ✖

Fecha inicio: 11-09-2024

Fecha finalización: 31-10-2024

Sprint 2 ✖

Fecha inicio: 01-11-2024

Fecha finalización: 22-12-2024

+ Añadir Sprint

Continuar

Corrige los siguientes errores:

Sprint 1: Las fechas deben estar dentro del periodo del curso (12-09-2024 a 21-12-2024)

Sprint 2: Las fechas deben estar dentro del periodo del curso (12-09-2024 a 21-12-2024)

Figura 7.3: Error obtenido al introducir fechas externas a las indicadas por el curso.

P-03	Fechas de <i>sprint</i> coincidentes con <i>sprint</i> anterior
Objetivo	Comprobar que el sistema no permite continuar con la creación de un <i>sprint</i> si la fecha de inicio o de finalización que ha indicado el usuario pertenece al rango temporal de uno de los sprints anteriores.
Precondición	El usuario se encuentra en el flujo de crear curso y ha seleccionado una fecha de inicio y fin para dicho curso.
Pasos	1. Completar los campos para un nuevo <i>sprint</i>. 2. Introducir una fecha de inicio o finalización que pertenezca al rango temporal de un <i>sprint</i> anterior. 3. Pulsar “Continuar”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que alguna de las fechas introducidas están contenidas en el rango de un <i>sprint</i> anterior, mostrando la fecha errónea.
Resultado obtenido	CORRECTO (ver Figura 7.4 para ver el mensaje de error).

Tabla 7.3: Especificación de la prueba 3.

Gestión de Curso

[◀ Atrás](#)

Gestionar Sprints

Sprint 1

Fecha inicio:
12-09-2024

Fecha finalización:
15-09-2024

Sprint 2

Fecha inicio:
14-09-2024

Fecha finalización:
15-09-2024

+ Añadir Sprint

Continuar

Corrige los siguientes errores:
Sprint 2 se solapa con el Sprint 1

[← Volver al inicio](#)

Figura 7.4: Error obtenido al introducir una fecha que pertenece al rango temporal de otro sprint.

P-04	Fechas de acciones correctas
Objetivo	Comprobar que el sistema no permite continuar con la creación o edición de una acción si la fecha de inicio es posterior a la de finalización o si la fecha de inicio no está dentro de los límites marcados por el <i>sprint</i> al que pertenece. Además, el sistema deberá mostrar una advertencia si la fecha de fin de la acción es externa a dichos límites del <i>sprint</i> .
Precondición	El usuario se encuentra en el apartado de la aplicación relativo a la gestión de acciones de aprendizaje y evaluación.
Pasos	1. Abrir el modal de crear o editar una acción y completar los campos de curso y <i>sprint</i>. 2. Introducir una fecha que no cumpla las condiciones requeridas, es decir, que o bien siendo la de inicio sea posterior a la de fin o sea externa a los límites indicados por el <i>sprint</i>; o bien siendo la de fin es externa a dichas restricciones. 3. Pulsar “Guardar”.
Resultado esperado	El sistema no permite seleccionar fechas externas a los límites indicados por el <i>sprint</i> . Además, mostrará un mensaje de error o advertencia si las fechas de inicio o fin respectivamente están fuera de los límites marcados por las fechas que definen el <i>sprint</i> .
Resultado obtenido	CORRECTO (ver Figura 7.5 para ver el mensaje de advertencia y la limitación a la hora de seleccionar una fecha).

Tabla 7.4: Especificación de la prueba 4.

The screenshot shows a web form titled "Nueva Acción de Aprendizaje". It has a "Descripción:" label followed by a text area. A date picker is open, showing the month of September 2025. The date 23 is selected. Below the date picker, there are two date input fields: "23/09/2025" and "Fecha Fin: 20/11/2025". An orange warning box contains the text: "Advertencia: La fecha fin seleccionada (2025-11-20) está después de la fecha máxima recomendada del sprint (2025-10-31)". At the bottom, there is a "Carga (minutos):" label and two buttons: "Guardar" and "Cancelar".

Figura 7.5: Advertencia y limitación al definir las fechas de una acción.

7.2. Pruebas para la carga de archivos

La segunda batería de pruebas nos permite confirmar que la aplicación desarrollada solo permite cargar archivos con la extensión o nombre indicados.

P-05	Archivo de estudiantes correcto
Objetivo	Comprobar que el sistema no permite realizar la carga de estudiantes si el archivo no tiene el nombre “estudiantes.xlsx” concretamente.
Precondición	El usuario se encuentra en el flujo de crear curso o de procesar estudiantes.
Pasos	1. Pulsar en el botón de “Elegir archivo”. 2. Seleccionar un fichero que no tenga el nombre “estudiantes.xlsx”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que la el archivo cargado no es válido.
Resultado obtenido	CORRECTO (ver Figura 7.6 para ver el mensaje de error).

Tabla 7.5: Especificación de la prueba 5.



Figura 7.6: Error obtenido al cargar un archivo de estudiantes erróneo.

P-06	Archivo comprimido correcto
Objetivo	Comprobar que el sistema no permite realizar la carga de autoevaluaciones o tareas si el archivo no tiene la extensión “.zip”.
Precondición	El usuario se encuentra en el flujo de procesar tareas o de procesar autoevaluaciones.
Pasos	1. Pulsar en el botón de “Elegir archivo”. 2. Seleccionar un fichero que no tenga la extensión “.zip”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que la el archivo cargado no es válido.
Resultado obtenido	CORRECTO (ver Figura 7.7 para ver el mensaje de error).

Tabla 7.6: Especificación de la prueba 6.

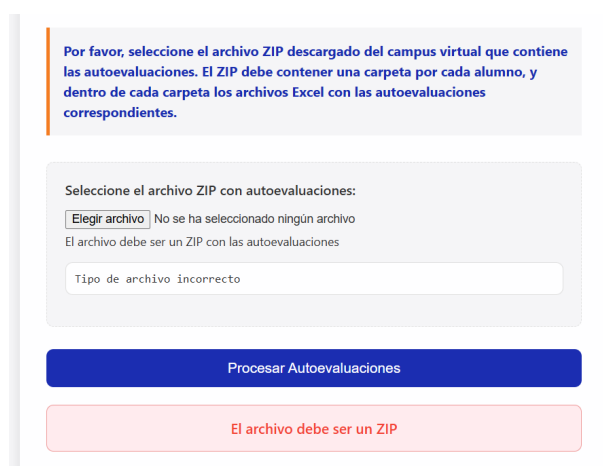


Figura 7.7: Error obtenido al cargar un archivo que no tiene la extensión “.zip”.

P-07	Archivo de hoja de cálculo correcto
Objetivo	Comprobar que el sistema no permite realizar la carga de evaluaciones o resultados si el archivo no tiene la extensión “.xlsx”.
Precondición	El usuario se encuentra en el flujo de procesar evaluaciones o de procesar resultados.
Pasos	1. Pulsar en el botón de “Elegir archivo”. 2. Seleccionar un fichero que no tenga la extensión “.xlsx”.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que la el archivo cargado no es válido.
Resultado obtenido	CORRECTO (ver Figura 7.8 para ver el mensaje de error).

Tabla 7.7: Especificación de la prueba 7.

Por favor, seleccione el archivo XLSX descargado de Microsoft Forms que contiene los resultados de las respuestas de los alumnos a las rutinas de pensamiento.

Seleccione el archivo XLSX con los resultados:

No se ha seleccionado ningún archivo

El archivo debe ser un XLSX con los resultados de Microsoft Forms

Tipo de archivo incorrecto

Procesar Resultados

El archivo debe ser un XLSX

Figura 7.8: Error obtenido al cargar un archivo que no tiene la extensión “.xlsx”.

7.3. Pruebas para eliminación de información

Por último, mostraremos una prueba que hemos realizado con el fin de asegurar que la eliminación de datos que tienen otros previamente asociados en la base de datos está controlada, para evitar reacciones en cadena o la existencia de datos con errores después de las eliminaciones.

P-08	Eliminación controlada
Objetivo	Comprobar que el sistema no permite eliminar objetivos, historias, criterios, preguntas, plantillas o acciones que tengan asociados otros datos en la aplicación.
Precondición	El usuario se encuentra consultando los elementos de producto de aprendizaje, las plantillas y preguntas o las acciones.
Pasos	1. Pulsar en el botón de “Eliminar” de alguno de los elementos mostrados. 2. Pulsar en el botón de “Aceptar” en el cuadro de diálogo que se muestra.
Resultado esperado	El sistema muestra un mensaje de alerta indicando que no se puede eliminar dicho elemento porque tiene otros asociados.
Resultado obtenido	CORRECTO (ver Figura 7.9 para ver el mensaje de error).

Tabla 7.8: Especificación de la prueba 8.

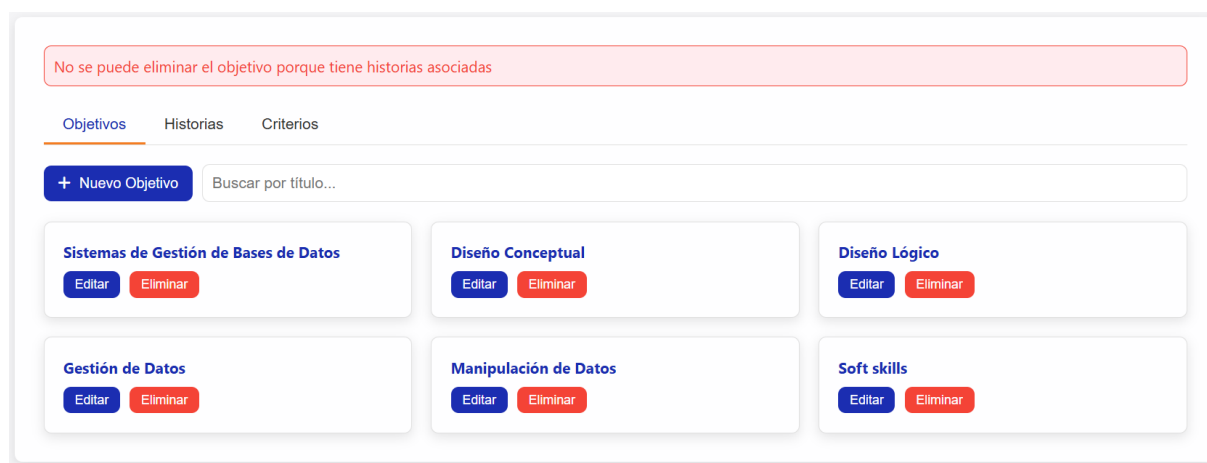


Figura 7.9: Error obtenido al tratar de eliminar un elemento con dependencias.

Parte III

Resultados

Capítulo 8

Aceptación

En este capítulo se detallan los procedimientos seguidos para verificar el cumplimiento (aceptación) de cada uno de los objetivos planteados en el trabajo, utilizando métodos sistemáticos que permiten evaluar tanto los aspectos técnicos como los pedagógicos de la solución desarrollada. Para ello, hemos decidido estructurar este capítulo en tres apartados principales, uno por cada uno de los objetivos que habíamos fijado al inicio de este trabajo (véase la Tabla 1.1).

8.1. Objetivo 1: Modelo Conceptual

El primer objetivo, centrado en el diseño del modelo conceptual para representar el funcionamiento de una asignatura bajo la metodología *Test-Driven Learning*, ha sido verificado mediante un proceso que incluye dos puntos principalmente. El primero de ellos se basa en la validación de un experto, y este se combina con otro basado en pruebas técnicas que verifican si la implementación realizada es correcta y se adapta completamente a la información disponible antes de desarrollar este Trabajo Fin de Grado (TFG).

Comenzando por el primero de los puntos, el modelo conceptual desarrollado ha sido sometido a revisiones iterativas por parte del autor de la metodología *Test-Driven Learning*, a la que da soporte este modelo. Este experto ha evaluado, entre otras cosas, cómo se adapta la representación conceptual realizada a todos los elementos que componen dicha metodología, definiendo si todos estos están correctamente representados y asociados en el modelo. Este proceso de validación ha asegurado que el modelo capture adecuadamente todos los elementos fundamentales de la metodología, incluyendo el producto y proceso de aprendizaje, y los resultados obtenidos de su evaluación.

Por otro lado, la capacidad de representar todos los datos disponibles del modelo ha sido probada mediante la carga de datos reales. Además, como se explica de forma detallada en la Sección 6.2, este proceso ha permitido identificar y resolver diversas inconsistencias en los datos de entrada, demostrando la robustez del modelo.

Dentro de las correcciones realizadas, queremos destacar, entre otras, las realizadas a los archivos de tareas (cuadernos de trabajo), que presentaban algunos problemas. Entre ellos, el más destacado es que la información que contenían estos cuadernos no permitía cargar todas las partes del modelo diseñado, ya que este era más completo que lo que había en los datos representados. Por ello, se realizó un mapeo de forma manual, utilizando las fechas o las descripciones de las tareas que habían llevado a cabo los alumnos con el fin de poder completar la información

faltante.

La superación exitosa de estos desafíos técnicos, junto con la validación conceptual por parte del experto, confirman que el Objetivo 1 ha sido cumplido satisfactoriamente, proporcionando una base sólida para los siguientes desarrollos.

8.2. Objetivo 2: Mecanismo de Gestión Automatizada

La consecución del segundo objetivo, relativo al diseño e implementación del sistema automatizado para la gestión de recursos educativos, ha sido evaluada mediante dos enfoques complementarios que demuestran tanto su capacidad técnica como su utilidad práctica.

Desde el punto de vista técnico, el sistema ha demostrado su eficacia al procesar automáticamente más de 34.000 registros de datos educativos, volumen que resultaría inviable manejar manualmente. Este procesamiento masivo incluye la importación, normalización y almacenamiento estructurado de todos los tipos de datos generados en el contexto de una asignatura que se desarrolla siguiendo la metodología de *Test-Driven Learning*, validando así la escalabilidad de la solución propuesta. De este modo, se ha logrado procesar la totalidad de los datos generados durante un curso académico en la asignatura.

Para evaluar la usabilidad y funcionalidad del sistema desde la perspectiva del usuario final, se ha desarrollado un formulario de evaluación implementado en *Microsoft Forms* (<https://forms.office.com/e/zpMfhdl1ydX>). Este instrumento permite evaluar que todas las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) se pueden realizar sobre los diferentes tipos de datos educativos que se pueden generar utilizando la solución desarrollada, así como probar los mecanismos de procesamiento automático implementados. En las Figuras 8.1, 8.2 y 8.3 encontramos las preguntas que se han realizado en dicho formulario. Este ha sido completado por el profesor responsable de la asignatura cuyos datos han sido utilizados en este trabajo, quien evaluará si el sistema cubre todas las necesidades operativas reales en un contexto educativo. Además, el instrumento incluye secciones específicas para recoger sugerencias de mejora y recomendaciones sobre funcionalidades adicionales que podrían incorporarse en futuras versiones.

Evaluación de las opciones de CRUD (Crear, Leer, Actualizar, Eliminar).

Crear producto y proceso de aprendizaje

**Gestionar producto de aprendizaje**
Crea, modifica y elimina los elementos del producto de aprendizaje (objetivos, historias y criterios).

**Gestionar acciones**
Crea, modifica, replanifica y edita acciones de aprendizaje y evaluación, programándolas para los sprints disponibles.

**Crear plantillas y preguntas**
Crea, modifica y elimina plantillas y preguntas, definiendo así los criterios que se medirán en las acciones de evaluación.

1

Que operaciones CRUD permite realizar la aplicación para los OBJETIVOS. *

☐ Crear

☐ Leer

☐ Eliminar

☐ Actualizar (Editar)

Figura 8.1: Interfaz del formulario de evaluación para el sistema de gestión.

Evaluación de las opciones de AUTOMATIZACIÓN.

Procesamiento de datos

**Autoevaluaciones**
Carga y procesa las autoevaluaciones de los estudiantes, eligiendo la acción a la que están asociadas.

**Evaluaciones**
Carga y procesa las evaluaciones del profesor para cada examen.

**Resultados**
Carga y procesa el excel que contiene los resultados de las rutinas que realizan los estudiantes al finalizar una acción de evaluación.

**Tareas**
Carga y procesa los cuadernos de trabajo que han completado los estudiantes durante el sprint.

8

Cuales de los siguientes archivos se pueden cargar de forma automática desde la aplicación eXtreme Processing. *

☐ Tareas (cuadernos de trabajo)

☐ Autoevaluaciones

☐ Evaluaciones

☐ Resultados de las rutinas de aprendizaje

Figura 8.2: Sección del formulario dedicada a evaluar el procesamiento automático.

Sección 3

Preguntas generales sobre la aplicación

9

¿En tus próximos cursos, utilizarías eXtreme Processing para generar los elementos del producto y proceso de aprendizaje de la asignatura? *

Escriba su respuesta

10

...

¿Qué crees que le falta a la herramienta eXtreme Programming en términos de información y funcionalidad? *

Escriba su respuesta

11

¿Consiente el uso de las respuestas proporcionadas en este cuestionario para la realización de un TFM y una posible publicación, garantizando siempre el anonimato de los datos? *

☐ Sí

☐ No

+ Agregar nueva pregunta

Figura 8.3: Sección del formulario dedicada a preguntas de respuesta abierta.

Los resultados de esta evaluación han sido positivos, pudiendo asegurar que todas las operativas que debe permitir realizar la aplicación son posibles, tal como se puede comprobar en la siguiente imagen de los resultados obtenidos para el formulario planteado, en la que se ve como todas las operaciones CRUD se pueden realizar, en concreto para los objetivos. Se ha obtenido el mismo resultado para el resto de elementos por los que se pregunta en el formulario.



Figura 8.4: Resultados obtenidos en el formulario que evalúa el objetivo 2.

En cuanto a las preguntas de respuesta abierta, los resultados obtenidos nos arrojan confianza de que esta herramienta se va a utilizar en el ámbito académico en un futuro próximo. También podemos verificar que la herramienta es una buena base sobre la que seguir construyendo otras herramientas más completas, ya que esta que hemos desarrollado es únicamente una prueba de concepto.

La combinación de estas dos líneas de evaluación -la técnica mediante el procesamiento masivo de datos y la práctica mediante el formulario de usuario- proporciona una verificación robusta del cumplimiento del Objetivo 2.

8.3. Objetivo 3: Solución de *Learning Analytics*

La validación del tercer objetivo, centrado en el desarrollo de una solución ilustrativa del potencial de la base de conocimiento obtenida, se ha realizado mediante un enfoque dual que combina evaluación por parte de un usuario final de la herramienta con la demostración de su aplicabilidad en un contexto real.

La evaluación cualitativa se ha llevado a cabo utilizando el cuestionario EFLA (Evaluation Framework for Learning Analytics) desarrollado por [18]. Este instrumento, representado gráficamente en las Figuras 8.5 y 8.6, y que se puede encontrar en <https://forms.office.com/e/WUpxyAQGGM>, evalúa tres dimensiones clave de las herramientas de LA: la transparencia en el manejo de datos, la capacidad para promover conciencia y reflexión pedagógica, y el impacto real en la práctica docente. Cada ítem se valora en una escala de 5 puntos, donde 1 representa total desacuerdo y 5 acuerdo completo.

El cuestionario ha sido completado por el profesor responsable de la metodología *eXtreme Learning*, proporcionando una evaluación exhaustiva de cómo la herramienta desarrollada apoya los procesos de toma de decisiones educativas. Los ítems evaluados incluyen aspectos como la claridad en la recogida y uso de datos, la capacidad para visualizar el estado actual de aprendizaje de los estudiantes, la utilidad para prever escenarios futuros, y el estímulo para reflexionar y

1

Las siguientes preguntas corresponden al cuestionario EFLA.

* 

	Extremadamente en desacuerdo	Desacuerdo	Neutro	De acuerdo	Extremadamente de acuerdo
¿Para esta herramienta está claro qué datos se recopilan?	☐	☐	☐	☐	☐
¿Para esta herramienta está claro por qué se recopilan los datos?	☐	☐	☐	☐	☐
¿Esta herramienta me permite conocer la situación actual de aprendizaje de mis alumnos?	☐	☐	☐	☐	☐
¿Esta herramienta me permite predecir la posible situación futura de aprendizaje de mis alumnos, considerando su comportamiento (sin no hubiese cambios en el currículum)?	☐	☐	☐	☐	☐
¿Esta herramienta me estimula a reflexionar sobre mi comportamiento docente anterior?	☐	☐	☐	☐	☐
¿Esta herramienta me estimula a adaptar mi comportamiento docente si es necesario?	☐	☐	☐	☐	☐
¿Esta herramienta me estimula a enseñar de forma más eficiente?	☐	☐	☐	☐	☐
¿Esta herramienta me estimula a enseñar de forma más eficaz?	☐	☐	☐	☐	☐

Figura 8.5: Primera parte del cuestionario EFLA utilizado para la evaluación

adaptar las estrategias docentes.

Los resultados obtenidos para este formulario se presentan a continuación, y son muy positivos, ya que la herramienta de *Learning Analytics* implementada cumple con creces las expectativas, obteniendo resultados muy buenos en el cuestionario EFLA. Podemos consultar los resultados en la Figura 8.7.

Además, gracias a las respuestas en las preguntas abiertas podemos confirmar que, al igual que para la parte de gestión de la aplicación, el *dashboard* diseñado se va a utilizar en el futuro más próximo en una asignatura universitaria. El profesor responsable que ha respondido a esta encuesta nos ha indicado, además, que aunque la funcionalidad es acotada por ser una prueba de concepto, la base es muy buena para construir soluciones de *Learning Analytics* a partir del desarrollo realizado.

Para completar esta evaluación docente, el objetivo se ha verificado también mediante una demostración práctica de aplicabilidad: el desarrollo exitoso de un Trabajo Fin de Máster [5] que ha utilizado como base la infraestructura de datos creada en este TFG. Este trabajo paralelo ha implementado con éxito una herramienta que ofrece a los docentes analíticas del currículo de la asignatura de Bases de Datos, validando que la base de conocimiento generada permite desarrollar aplicaciones reales que cumplen objetivos pedagógicos concretos.

Preguntas generales sobre la aplicación

...

2

¿En tus próximos cursos, utilizarías eXtreme Processing para analizar el progreso de aprendizaje de los alumnos en términos de evaluación? *

Escriba su respuesta

3

¿Qué crees que le falta al dashboard de eXtreme Programming para analizar el progreso de aprendizaje de los alumnos en términos de evaluación? *

Escriba su respuesta

4

¿Consiente el uso de las respuestas proporcionadas en este cuestionario para la realización de un TFM y una posible publicación, garantizando siempre el anonimato de los datos? *

☐ Sí

☐ No

Figura 8.6: Segunda parte del cuestionario EFLA utilizado para la evaluación

La convergencia de estos dos enfoques -la evaluación con usuarios mediante EFLA y la demostración práctica mediante el TFM asociado- confirma que el Objetivo 3 ha sido alcanzado satisfactoriamente, ofreciendo una solución de *Learning Analytics* viable y útil en el contexto de la metodología *Test-Driven Learning*.

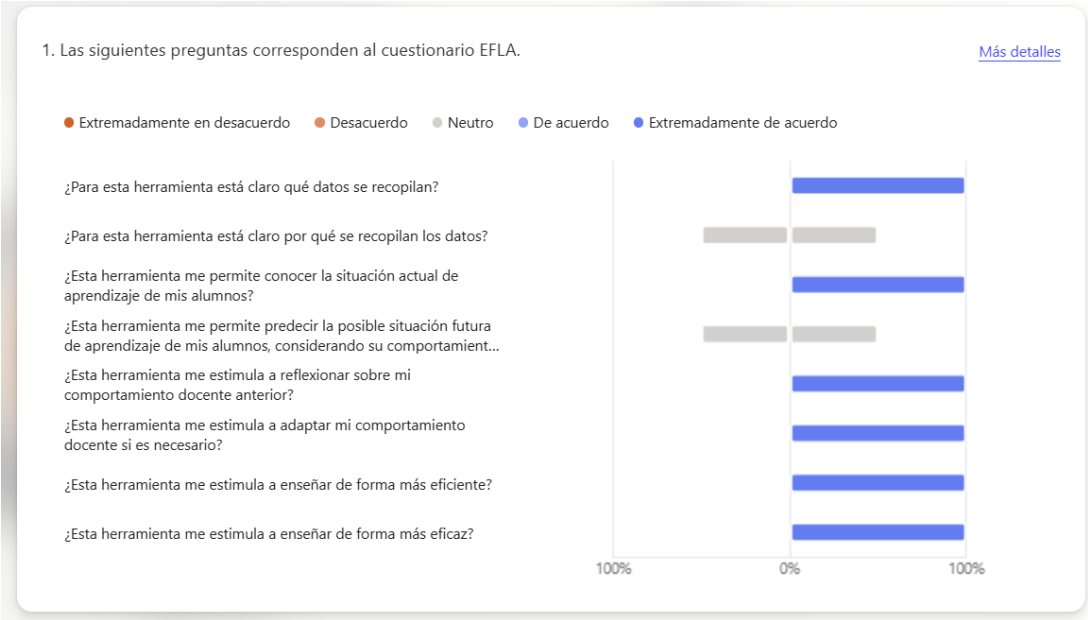


Figura 8.7: Segunda parte del cuestionario EFLA utilizado para la evaluación

Capítulo 9

Conclusiones y trabajo futuro

Presentaremos, para terminar la parte de Resultados de este Trabajo Fin de Grado (TFG), las conclusiones del proyecto y personales (ver apartado 9.1), así como las líneas de trabajo futuro que se han identificado y que permitirán sacar un mayor rendimiento del trabajo realizado en este TFG.

9.1. Conclusiones

La sección de conclusiones la vamos a dividir en dos apartados diferenciados. Un primer apartado en el que expondremos las conclusiones del proyecto, para posteriormente presentar unas conclusiones a nivel personal sobre el trabajo realizado.

9.1.1. Perspectiva del proyecto

En cuanto a las conclusiones del proyecto, vamos a organizarlas en tres puntos diferenciados y claves que queremos estudiar. Estos son el cumplimiento o no de los objetivos que presentamos al inicio (en la Sección 1.2), la utilidad de este proyecto para el futuro y cómo se ha desarrollado el proyecto en relación con la metodología de trabajo elegida.

- **Objetivos alcanzados:** Para afrontar este punto, vamos a ir analizando uno a uno los tres objetivos que se plantearon en la sección inicial de objetivos, y que se han cumplido satisfactoriamente. Comenzando por el Obj-01, recordemos que consistía en el diseño del modelo de dominio que represente una asignatura universitaria en la que se siga la metodología de *Test-Driven Learning*. Podemos asegurar que hemos cumplido este objetivo, pues si observamos la Sección 4.6, hemos construido ese modelo de dominio al completo que representa, como hemos podido comprobar, el funcionamiento al completo de esa asignatura. El segundo objetivo (Obj-2 en la Tabla 1.1) abordaba el diseño e implementación del mecanismo que automatiza el procesamiento de todos los recursos disponibles en una asignatura desarrollada con la metodología *Test-Driven Learning*. Este objetivo también se ha cumplido, ya que la parte del *backend* de la aplicación presentada en el capítulo 6, contiene toda la lógica para garantizar que el procesamiento de datos se hace de forma automática. Esto se ha podido verificar con la carga en la base de datos diseñada de la información disponible de una asignatura en la que se había aplicado la metodología de *Test-Driven Learning*. Esto último está directamente relacionado con el cumplimiento del tercero de los

objetivos, en el que se requería la construcción de una solución de *Learning Analytics* que validara la utilidad de la base de conocimiento obtenida al aplicar la automatización anterior sobre una asignatura universitaria real. A la consecución de este objetivo también ha contribuido la interfaz de usuario construida que permite facilitar el uso de este mecanismo de automatización, integrando todos los componentes en una aplicación amigable y fácil de utilizar para el docente, consiguiendo así la construcción de dicha solución de *Learning Analytics* pedida.

- **Utilidad para el futuro:** Parte de los resultados obtenidos en este TFG se están utilizando actualmente para el desarrollo de una herramienta de *Curriculum Analytics* en un Trabajo Fin de Máster [5]. En concreto el modelo conceptual realizado, y en consecuencia, la base de datos creada y toda la información cargada sobre la asignatura real utilizada. Estos datos se pueden explotar de muchas formas, como por ejemplo para obtener conclusiones sobre el desempeño de algunos alumnos o como impactan en ellos ciertas acciones de aprendizaje o evaluación, lo que puede ayudar al profesor a mejorar la enseñanza impartida en la asignatura. Por otra parte, la aplicación implementada será utilizada por los profesores responsables de asignaturas desarrolladas con *Test-Driven Learning*. Esta aplicación facilitará la planificación realizada por los profesores para la asignatura, así como la gestión y procesamiento de todos los resultados producidos por los estudiantes.
- **Desarrollo del proyecto:** La metodología ASAP utilizada ha sido clave para lograr cumplir los objetivos de este proyecto. Con esta metodología, hemos conseguido dividir el trabajo realizado en los *sprints* de forma lo más equitativa posible, manteniendo así el ritmo de trabajo durante todo el proyecto. Además, la aportación constante de *feedback* ofrecida por los tutores ha permitido identificar errores de forma temprana, de forma que se ha podido trabajar sin bloqueos, obteniendo así un producto de alta calidad.

9.1.2. Perspectiva personal

Este Trabajo Fin de Grado ha supuesto un gran aprendizaje en cuanto a cómo funciona un verdadero proyecto de desarrollo *software*. En este trabajo, hemos trabajado en casi todos los aspectos clave de este tipo de proyectos, logrando así un aprendizaje completo que me ha permitido crecer en el ámbito personal y profesional.

Me gustaría hacer en esta perspectiva personal un recorrido por todo el desarrollo del proyecto. En el mismo instante en el que se me presentó la opción de realizar este proyecto, no dudé en aceptarlo, ya que siempre he querido tener una visión más completa de cómo se hace un trabajo de análisis de datos al completo. Al inicio, el proyecto tenía un alcance incierto, ya que no sabíamos cuánto tiempo íbamos a disponer para el desarrollo, ya que la investigación requerida para este trabajo ha sido bastante grande. Los inicios estuvieron marcados por la construcción del modelo conceptual. De forma paralela, se fue trabajando en la investigación de toda la base de conocimiento necesaria para este proyecto y el desarrollo de este modelo conceptual, iterando varias veces sobre el mismo hasta encontrar el modelo completo.

Una vez disponíamos del modelo conceptual completo (o casi completo) pudimos avanzar con el siguiente resultado del TFG, la automatización del procesamiento de los datos. El aprendizaje en este punto ha sido muy grande, ya que he tenido que lidiar con datos que tenían muchos formatos diferentes y errores varios. Esto me ha hecho tener que pensar varias soluciones para conseguir evitar estos errores y que los datos cargados fueran consistentes, planteando todas

ellas y eligiendo la más adecuada para la situación en la que se encontraba el proyecto. Con ello he puesto a prueba mi capacidad de resolución de problemas, directamente relacionada con la carrera de Matemáticas que también he cursado, viendo así como se pueden relacionar ambas directamente y sacar partido de las cosas aprendidas en una para la otra.

Finalmente, con la aplicación que se ha implementado he aprendido como de forma sencilla se puede construir una herramienta que es de gran ayuda para los profesores en este caso. Observando como todo el proceso que había construido previamente quedaba reflejado en una aplicación funcional que cualquier persona podría utilizar sin requerir de conocimientos tecnológicos avanzados.

Para terminar esta valoración personal, quiero destacar que a lo largo de todo el proyecto he aplicado los conocimientos adquiridos a lo largo del grado, viendo como todo lo aprendido tiene una aplicación en un caso que podría ser real. Además, me ha servido como aprendizaje para el comienzo de mi vida profesional, al utilizar una metodología *Agile* para desarrollarlo, muy similar a las utilizadas en el ámbito del desarrollo *software* profesional.

9.2. Trabajo futuro

En el proyecto presentado en este trabajo, hemos desarrollado e integrado las contribuciones planteadas para cada uno de los objetivos detallados, pero aún quedan varias vías por las que se puede continuar trabajando, ya sea para mejorar este trabajo o para sacar partido a las soluciones planteadas.

En cuanto a las mejoras posibles, al ser una prueba de concepto, la aplicación no es completa y se podrían incorporar muchas funcionalidades para que esté lista para su uso académico. La primera de ellas es que actualmente solo se ha considerado que sea utilizada por los profesores, pero podría incorporarse una gestión de usuarios de forma que los alumnos también pudieran utilizarla y visualizar datos de su rendimiento en la asignatura. Además, actualmente solo se contempla la posibilidad de que se utilice para una única asignatura, pudiendo ser una línea futura de trabajo que se permita elegir varias asignaturas y crearlas, de forma que un profesor que imparta varias materias pueda tener organizada toda la información y clasificada para ello. Por último, queremos destacar la posibilidad de realizar una integración con herramientas de inteligencia artificial, que se podrían utilizar para “extraer” datos de fuentes menos estructuradas, como la asociación de las tareas a las acciones, que para este trabajo se han procesado manualmente. De esta forma, reduciríamos la carga de trabajo a la supervisión y curación de los resultados obtenidos con inteligencia artificial generativa.

Pasando a las formas de sacar partido a las contribuciones realizadas, tenemos varios puntos con los que podemos hacerlo, algunos de ellos, como hemos comentado antes, que ya se están implementando en un TFM. Estas vías de trabajo futuro son las siguientes:

1. Aumentar las opciones de visualización de los datos, haciendo *dashboards* completos tanto para profesores como para alumnos.
2. Realizar un análisis completo con los datos procesados, para obtener conclusiones sobre el desempeño de los alumnos en la asignatura.
3. Informar la toma de decisiones de profesores con los datos almacenados y visualizados.
4. Realizar un análisis predictivo con los datos procesados, aplicando modelos de *Machine Learning*.

5. Integrar modelos de Inteligencia Artificial generativa para que se puedan analizar los datos de forma automática, y aportar *feedback* en base a ello.

Bibliografía

- [1] Kent Beck et al. *Manifesto for Agile Software Development*. 2001. URL: <https://agilemanifesto.org/>.
- [2] Pedro Carvalho. *Data Warehouse ETL Toolkit – Chapter 03 – Part 1 – Extracting*. Publicado en el blog de Pedro Analytics. Mar. de 2022. URL: <https://pedroanalytics.com/2022/03/18/data-warehouse-etl-toolkit-chapter-03/>.
- [3] Liyanachchi Mahesha Harshani De Silva, María Jesús Rodríguez-Triana, Irene-Angelica Chounta y Gerti Pishtari. «Curriculum analytics in higher education institutions: a systematic literature review». En: *Journal of Computing in Higher Education* (2024), págs. 1-47.
- [4] Digital.ai. *17th Annual State of Agile Report*. Accedido: 29-03-2025. 2023. URL: <https://2288549.fs1.hubspotusercontent-na1.net/hubfs/2288549/RE-SA-17th-Annual-State-Of-Agile-Report.pdf>.
- [5] Clara Gándara González. *Learning analytics en el marco de eXtreme Learning*. TFM, Universidad de Valladolid. 2025.
- [6] Davinia Hernández-Leo, Roberto Martínez-Maldonado, Abelardo Pardo, Juan A Muñoz-Cristóbal y María J Rodríguez-Triana. «Analytics for learning design: A layered framework and tools». En: *British Journal of Educational Technology* 50.1 (2019), págs. 139-152.
- [7] Patricio Letelier y María del Carmen Penadés. «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)». En: *DYNA* 81.2 (2006), págs. 109-114.
- [8] Katerina Mangaroska y Michail Giannakos. «Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning». En: *IEEE Transactions on Learning Technologies* 12.4 (2018), págs. 516-534.
- [9] Mike Mannion y Barry Keepence. «SMART requirements». En: *ACM SIGSOFT Software Engineering Notes* 20.2 (1995), págs. 42-47.
- [10] Miguel Ángel Martínez Prieto, Jorge Silvestre, Aníbal Bregón, Patricia Baz Domínguez, Clara Gándara González, Paula Mielgo Martín e Irene Peñas Pérez. «Una metodología basada en prácticas ágiles para la realización de Trabajos Fin de Grado». En: *Actas de las JENUI - Vol. 8 (2023)*. 2023, págs. 367-374.
- [11] Miguel A. Martínez-Prieto, Jorge Silvestre, Aníbal Bregón y Diego García-Álvarez. «Proyectos de Aprendizaje (ágiles)». En: *Innovación docente en Educación Superior: interacción, participación y colaboración*. Agilice Digital, 2023, págs. 287-304. ISBN: 978-84-18583-04-9.

- [12] Miguel Ángel Martínez Prieto, Jorge Silvestre y Ruth Pinedo González. «Visibilizando el aprendizaje a través de la autoevaluación». En: *Actas de las JENUI - Vol. 9 (2024)*. 2024, págs. 51 -58.
- [13] H. Gilbert Miller y Peter Mork. «From Data to Decisions: A Value Chain for Big Data». En: *IT Professional* 15.1 (2013), págs. 57-59. DOI: 10.1109/MITP.2013.11.
- [14] Xavier Ochoa, Charles Lang, George Siemens, Alyssa Wise, Dragan Gasevic y Agathe Merceron. «Multimodal Learning Analytics-Rationale, Process, Examples, and Direction». En: *The Handbook of Learning Analytics* (2022), págs. 54-65.
- [15] Michael E Porter y Competitive Strategy. «Techniques for analyzing industries and competitors». En: *Competitive Strategy*. New York: Free 1 (1980).
- [16] María Jesús Rodríguez-Triana. «Linking scripting and monitoring support in blended cscl escenarios». Disponible en <https://uvadoc.uva.es/handle/10324/7049>. PhD thesis. Universidad de Valladolid, 2014.
- [17] Pasquale Salza, Paolo Musmarra y Filomena Ferrucci. «Agile methodologies in education: A review». En: *Agile and lean concepts for teaching and learning: Bringing methodologies from industry to the classroom* (2019), págs. 25-45.
- [18] Maren Scheffel. «The Evaluation Framework for Learning Analytics». Tesis doct. Open Universiteit, 2017.
- [19] Ken Schwaber y Jeff Sutherland. «La guía de Scrum». En: *Scrumguides. Org* 1 (2013), pág. 21.
- [20] Beat A Schwendimann, Maria Jesus Rodriguez-Triana, Andrii Vozniuk, Luis P Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet y Pierre Dillenbourg. «Perceiving learning at a glance: A systematic literature review of learning dashboard research». En: *IEEE Transactions on Learning Technologies* 10.1 (2016), págs. 30-41.
- [21] Shashi Kant Shankar, María Jesús Rodríguez-Triana, Adolfo Ruiz-Calleja, Luis P Prieto, Pankaj Chejara y Alejandra Martínez-Monés. «Multimodal data value chain (m-dvc): A conceptual tool to support the development of multimodal learning analytics solutions». En: *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 15.2 (2020), págs. 113-122.
- [22] Shashi Kant Shankar, Adolfo Ruiz-Calleja, Luis P Prieto, María Jesús Rodríguez-Triana y Pankaj Chejara. «An Architecture and Data Model to Process Multimodal Evidence of Learning». En: *Advances in Web-Based Learning-ICWL 2019: 18th International Conference, Magdeburg, Germany, September 23-25, 2019, Proceedings* 18. Springer. 2019, págs. 72-83.
- [23] George Siemens y Ryan SJ d Baker. «Learning analytics and educational data mining: towards communication and collaboration». En: *Proceedings of the 2nd international conference on learning analytics and knowledge*. 2012, págs. 252-254.
- [24] Society for Learning Analytics Research (SoLAR). *What is Learning Analytics?* Accedido: 17-03-2025. 2025. URL: <https://www.solaresearch.org/about/what-is-learning-analytics/>.
- [25] Esther van der Stappen. «Workplace Learning Analytics in Higher Engineering Education». En: *2018 IEEE Global Engineering Education Conference (EDUCON)*. 2018, págs. 15-20.

Parte IV

Apéndices

Apéndice A

Manual de Instalación

En este primer apéndice, vamos a presentar un manual de instalación detallado que se puede utilizar para desplegar y ejecutar la aplicación que hemos desarrollado como parte de este Trabajo Fin de Grado. Con este manual se pretende que todos los usuarios que lo deseen puedan realizar la instalación al completo de la aplicación para probarla, requiriendo solo conocimientos básicos de informática para ello. Se presentan a continuación los pasos a seguir para completar esta instalación.

A.1. Requisitos previos

Si queremos lanzar y probar la aplicación, previamente necesitaremos tener instalados y correctamente configurados los siguientes programas y herramientas.

A.1.1. PyCharm

Necesitaremos instalar la última versión disponible del entorno de desarrollo integrado (IDE) *PyCharm*, que está contenido en la licencia que proporciona la universidad. En nuestro caso, la versión que hemos utilizado es la “2025.1.2”.

Este IDE se puede descargar de la página oficial de *JetBrains*: <https://www.jetbrains.com/pycharm/data-science/>.

Una vez descargado, procederemos con su instalación, en la que solo deberemos hacer clic en *Next* o Siguiente hasta completar la misma. Se puede ver una imagen del proceso de instalación en la Figura A.1.

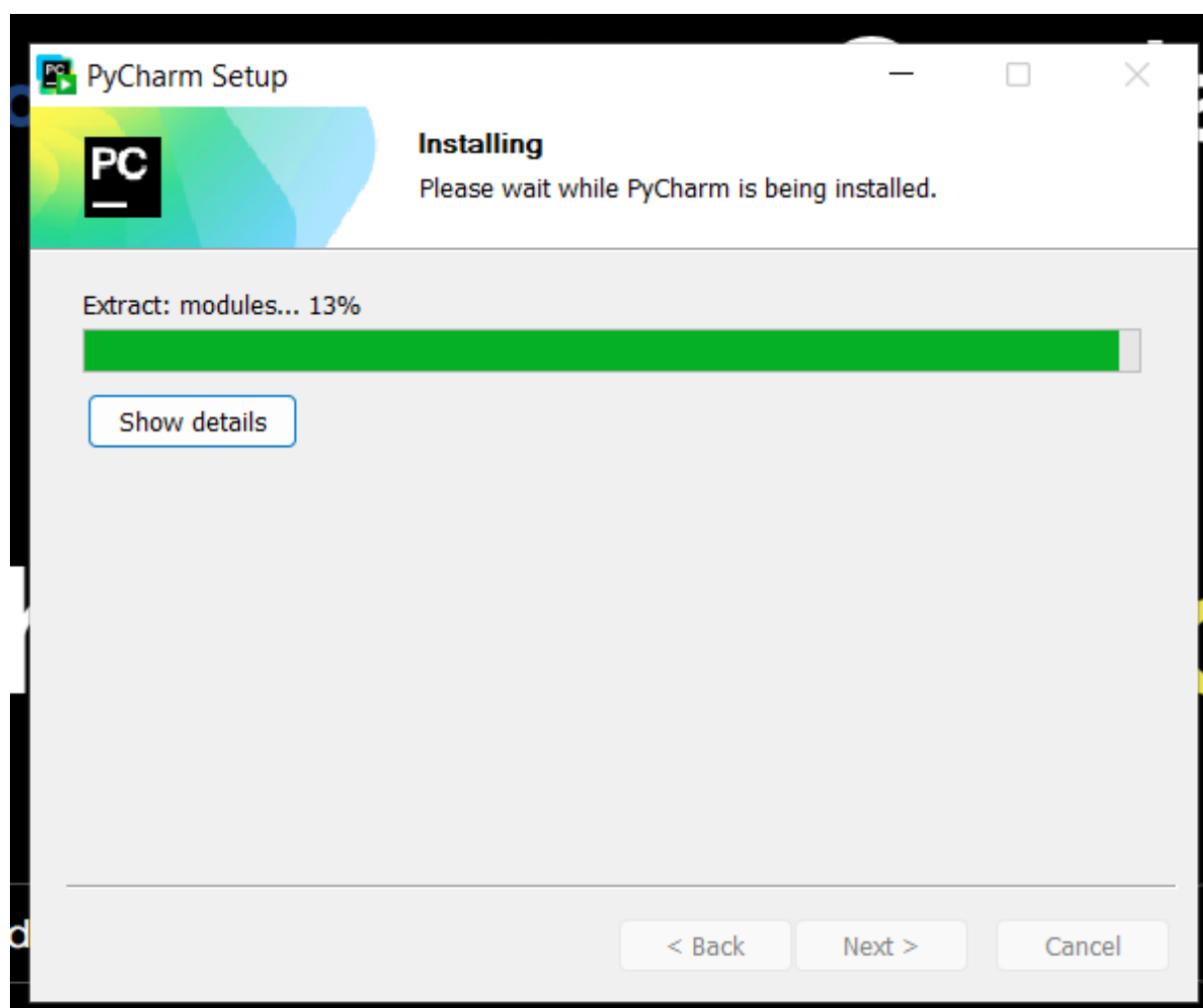


Figura A.1: Proceso de instalación de PyCharm.

Cuando la instalación haya finalizado, tendremos que abrir el IDE, aceptando todos los requisitos de licencia que se requieren.

A.1.2. Python

Se requiere una de las versiones disponibles de Python 3.12 para poder ejecutar la aplicación. Podemos descargar la última versión de Python 3.12 en: <https://www.python.org/downloads/release/python-3128/>.

La instalación es sencilla, deberemos elegir las opciones dadas por defecto en todas las pestañas de la misma. Cabe destacar que es importante tener localizado donde hemos instalado Python 3.12, ya que tendremos que utilizar esa ruta posteriormente para crear el entorno en el que se va a ejecutar el “backend” de la aplicación.

A.2. Preparación del entorno

Cuando hayamos completado la instalación, deberemos ejecutar PyCharm. Con PyCharm abierto, tendríamos que abrir un nuevo proyecto, y seleccionar la carpeta “backend” del mismo para abrirla.

Una vez completado el procedimiento, deberíamos ver lo que se muestra en la Figura A.2 en nuestra pantalla.

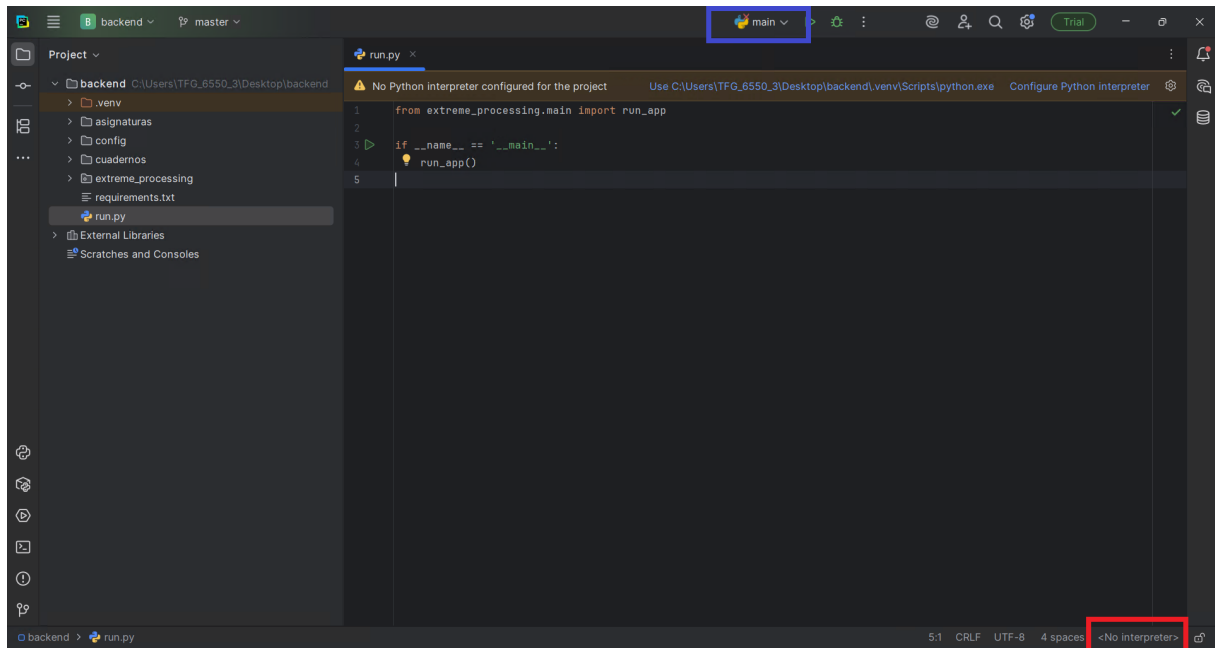


Figura A.2: PyCharm instalado y proyecto abierto.

El siguiente paso será configurar el entorno de en el que se va a ejecutar la aplicación, en concreto, el “backend”.

A.2.1. Crear entorno de Python

En la parte inferior derecha de la Figura A.2 (marcado en rojo) encontramos un apartado en el que podemos seleccionar la versión de Python que queremos utilizar. Al hacer clic sobre esta opción, en concreto en “Add new interpreter” y después en “Add local interpreter”, nos aparecerá una ventana en la que tendremos que seleccionar la versión de Python utilizada, en este caso la 3.12, y la ruta en la que tenemos instalada dicha versión. Esto se puede observar en la Figura A.3.

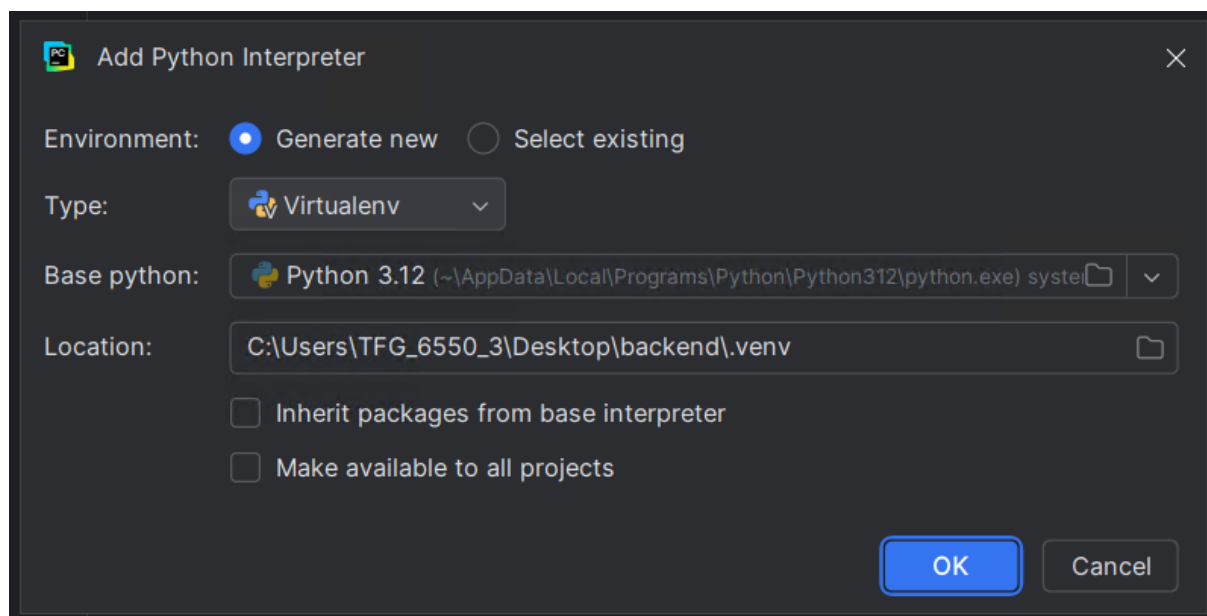


Figura A.3: Selección de versión de Python.

A.2.2. Instalar dependencias

Si el entorno ya está configurado, tendremos que instalar las dependencias definidas en el fichero de requerimientos (`requirements.txt`). Para ello, podemos utilizar el cuadro de diálogo que nos aparece en la parte superior de la pantalla en la que se nos pregunta si queremos instalar los requerimientos, o por el contrario utilizar la terminal contenida en PyCharm. Utilizando esta segunda, deberíamos ejecutar el siguiente comando: `pip install -r requirements.txt`.

A.2.3. Definir configuración de ejecución

Por último, antes de ejecutar la aplicación, debemos definir las variables de entorno necesarias para que la ejecución funcione. En este caso, las credenciales de la base de datos están almacenadas aquí para garantizar una mayor seguridad. En nuestro caso, debemos definir, además de estas variables, el *script* que queremos que comience la ejecución.

Para ello, en la parte superior de la A.2 (marcado en azul) tenemos que editar las configuraciones. Después de seleccionar esta opción, nos aparecerá una pestaña con un “+” en la parte superior izquierda, tendremos que seleccionar al pulsar en él la opción de “Python”. Después de ello nos aparecerá una nueva pestaña. En ella tendremos que configurar el nombre (se recomienda utilizar “main”), el entorno de Python utilizado (seleccionaremos el creado anteriormente), el script que queremos lanzar (en este caso será `run.py`) y el directorio de trabajo (la carpeta global “backend” del proyecto).

Finalmente, en el apartado de variables de entorno, debemos añadir una variable de entorno que se llame `BBDD_PASSWORD` y que contenga la contraseña de la cuenta con la que estemos utilizando la base de datos. Cuando hayamos completado todos estos pasos, nos quedaría de la siguiente forma:

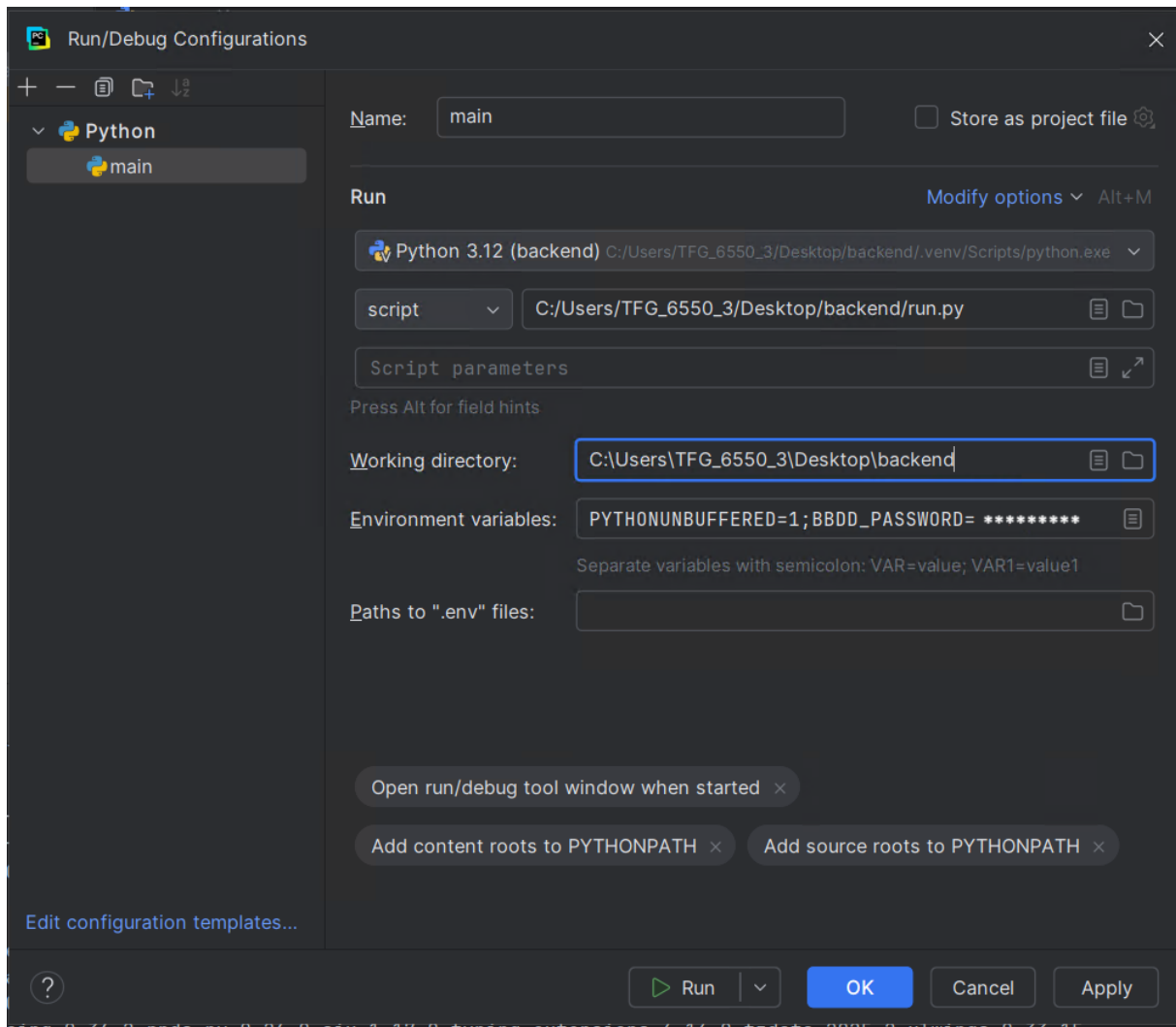


Figura A.4: Definición del entorno de ejecución.

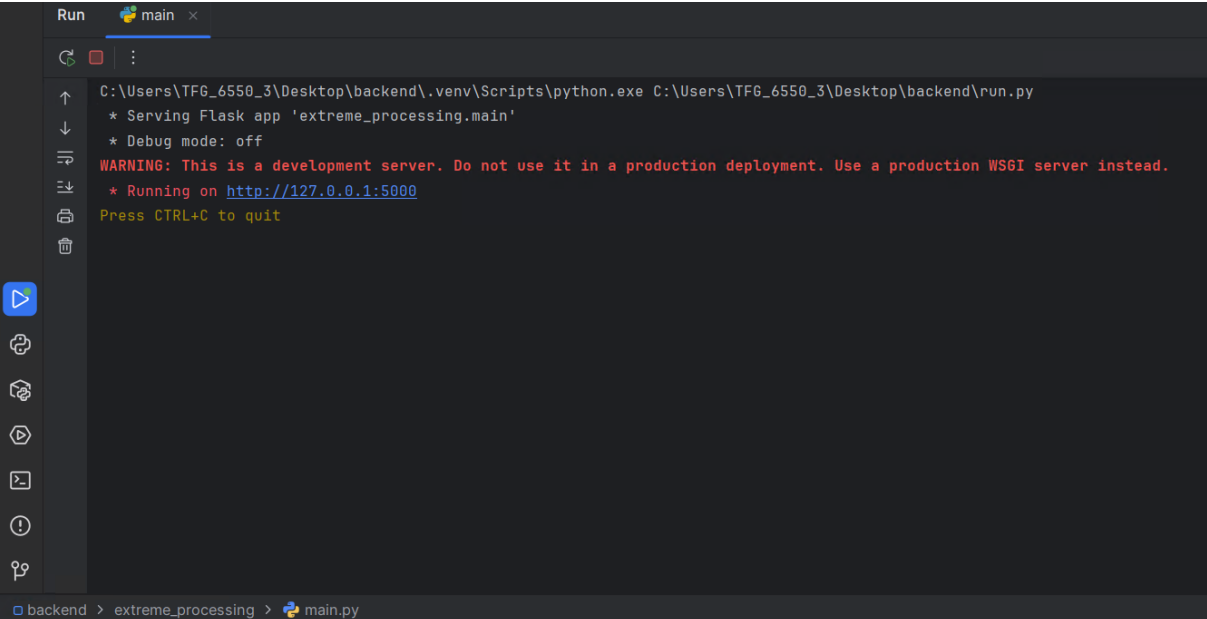
Para terminar, pulsaremos en “Aplicar” y finalmente en “OK” para terminar la configuración.

A.3. Ejecución de la aplicación

Si hemos terminado la instalación y se ha completado la configuración mencionada, estamos en disposición de ejecutar la aplicación para probarla.

A.3.1. Despliegue del backend

Bastaría con ejecutar desde PyCharm en el “play” verde que tenemos en la parte superior. En la terminal nos debería aparecer un mensaje como el que se ve en la siguiente imagen:



```
Run main x
C:\Users\TFG_6550_3\Desktop\backend\.venv\Scripts\python.exe C:\Users\TFG_6550_3\Desktop\backend\run.py
* Serving Flask app 'extreme_processing.main'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Figura A.5: Texto mostrado en terminal al ejecutar.

Esto significa que el backend está ejecutándose en la ruta indicada, que suele ser la siguiente:
`http://127.0.0.1:5000`

A.3.2. Despliegue del frontend

Para desplegar el frontend bastará con abrir el fichero `index.html` que está contenido en la carpeta de “frontend” del proyecto. Esto abrirá la página principal de la aplicación y permitirá utilizarla sin mayor problema, siempre que se tenga el “backend” ejecutándose como hemos explicado en el apartado anterior.

Apéndice B

Manual de Usuario

A continuación se presenta el manual de usuario, que explicará detalladamente cómo se utiliza cada una de las partes que componen la aplicación. Esta aplicación es muy sencilla de utilizar, y tiene varias partes diferenciadas que presentaremos en sendos apartados.

Antes de pasar con dichas partes, vamos a presentar la interfaz inicial de la aplicación, en la que vamos a destacar estas partes que posteriormente explicaremos.

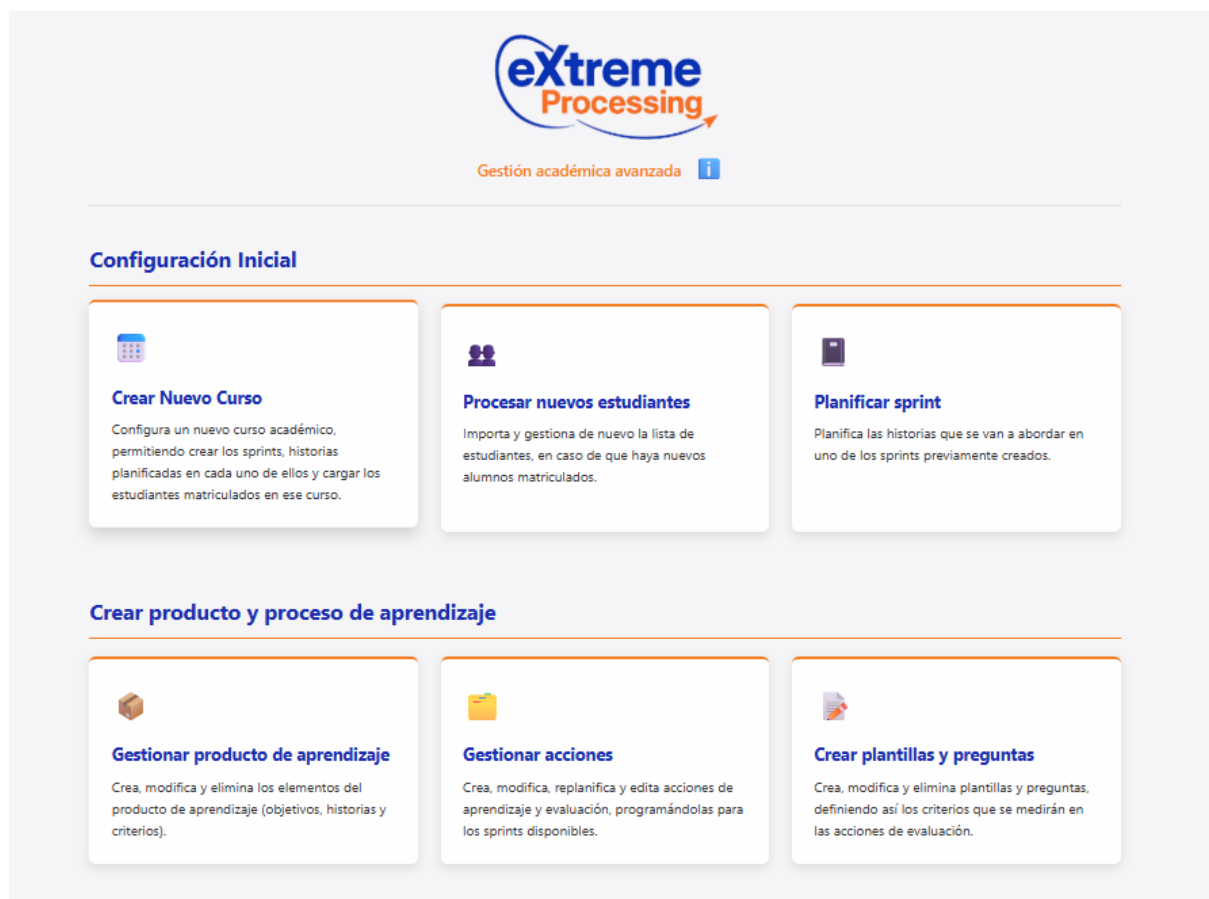


Figura B.1: Primera parte de la página principal.

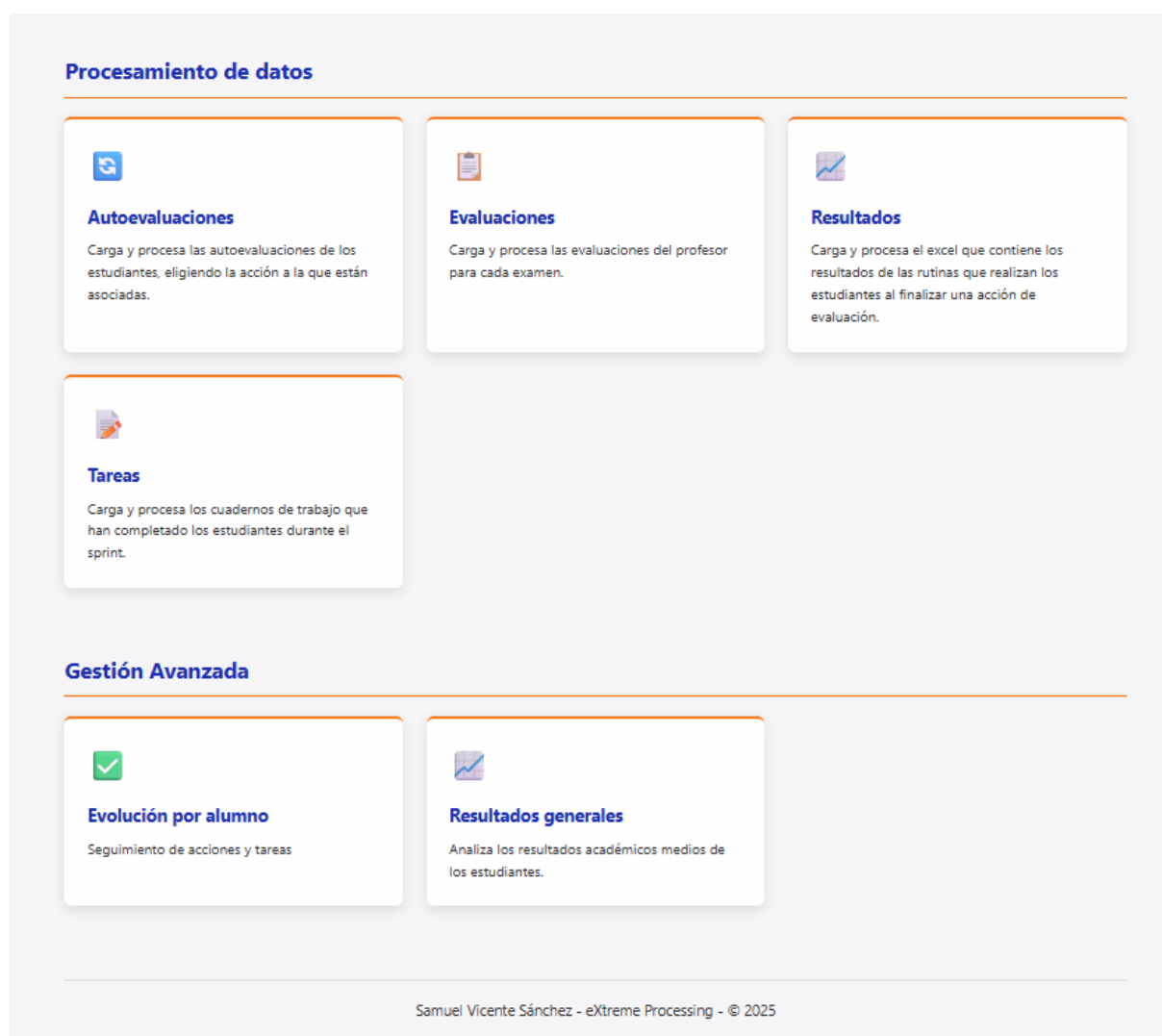


Figura B.2: Segunda parte de la página principal.

En esta interfaz podemos utilizar el botón de información que se ve al lado del logo en la Figura B.1 para obtener más información de la aplicación. El resto de opciones nos llevan a las páginas concretas que vamos a ir explicando por grupos a continuación:

B.1. Configuración inicial

Esta parte de la aplicación permite crear las configuraciones básicas que necesita hacer el profesor al iniciar un nuevo curso académico. La opción de crear curso incluye las otras dos opciones internamente, pero se permite hacerlas por separado por si es necesario realizar cualquier tipo de modificación. Veamos el flujo que sigue crear curso:

1. Cuando elegimos esta opción, la aplicación nos pide introducir, con el formato correcto, los años que abarca el curso y las fechas de inicio y fin. Cabe destacar que en este caso y en

todos los que veamos que conlleven la introducción de fechas se manejan los posibles errores como explicamos en el Capítulo 7. En la imagen B.3 podemos ver la primera pantalla que nos encontramos en este flujo.

El formulario se encuentra dentro de un recuadro con un fondo gris claro. En la parte superior, el título "Gestión de Curso" está en azul oscuro y subrayado. Debajo, el subtítulo "Crear Nuevo Curso" también está en azul oscuro. Hay tres campos de entrada de texto, cada uno con un ejemplo de formato: "Curso (formato 0000-0000):" con "Ej: 2023-2024", "Fecha inicio (formato 00-00-0000):" con "Ej: 12-09-2024", y "Fecha finalización (formato 00-00-0000):" con "Ej: 12-09-2024". Debajo de los campos hay un botón azul con el texto "Continuar". En la parte inferior izquierda hay un botón gris con el texto "← Volver al inicio".

Gestión de Curso

Crear Nuevo Curso

Curso (formato 0000-0000):
Ej: 2023-2024

Fecha inicio (formato 00-00-0000):
Ej: 12-09-2024

Fecha finalización (formato 00-00-0000):
Ej: 12-09-2024

Continuar

← Volver al inicio

Figura B.3: Selección de fechas del curso.

2. Tras completar con unas fechas válidas la primera parte, pasamos a la creación de los *sprints*. Aquí podremos definir cuantos *sprints* tiene el curso, y en que fechas inician y finalizan cada uno de ellos. La Figura B.4 ilustra esta página.

Gestión de Curso

[◀ Atrás](#)

Gestionar Sprints

Sprint 1

Fecha inicio:

Fecha finalización:

+ Añadir Sprint

Continuar

Datos del curso validados correctamente. Ahora puedes planificar los sprints.

[← Volver al inicio](#)

Figura B.4: Creación de los sprints.

3. El siguiente paso será programar las historias que se van a abordar en cada uno de los *sprints* definidos previamente. Cabe destacar que esta acción también se puede hacer desde la opción “Planificar sprint” disponible en la página principal, como veíamos en B.1. Véase la Figura B.5 que muestra la página de selección de historias en un sprint.

Gestión de Curso

[← Atrás](#)

Selección de Historias por Sprint

Sprint 1

- Diseño Conceptual +
- Diseño Lógico +
- Gestión de Datos +
- Manipulación de Datos +
- Sistemas de Gestión de Bases de Datos -

[Seleccionar todo](#)

Fundamentos de Bases de Datos
☐ Conocer los fundamentos conceptuales y tecnológicos de las bases de datos para poder diseñar y construir soluciones que satisfagan los requisitos de información de los sistemas software a desarrollar en el ámbito profesional.

Soft skills +

Continuar

Sprints validados correctamente. Ahora puedes asignar historias a los sprints.

[← Volver al inicio](#)

Figura B.5: Selección de historias asignadas para cada *sprint*.

4. Por último, podremos cargar los estudiantes que se han matriculado en el curso que estamos creando. Para ello, debemos seleccionar el fichero de estudiantes que se puede descargar directamente desde moodle y procesarlo. Además, podemos rehacer esta acción desde “Procesar nuevos estudiantes” disponible en la página principal, como veíamos en B.1, por si fuera necesario incorporar nuevas matrículas que se hayan añadido posteriormente. En la Figura B.6 se muestra esta página de procesamiento de estudiantes.

Gestión de Curso

[◀ Atrás](#)

Carga de Estudiantes

Seleccione el archivo de estudiantes:

No se ha seleccionado ningún archivo

El archivo debe llamarse "estudiantes.xlsx"

Confirmar y Finalizar

Historias asignadas a los sprints correctamente. Ahora puedes cargar el archivo de estudiantes.

[← Volver al inicio](#)

Figura B.6: Carga de estudiantes matriculados en un curso.

B.2. Crear producto y proceso de aprendizaje

En la segunda sección de la aplicación podremos gestionar todos los elementos relacionados con el producto y el proceso de aprendizaje de la asignatura en cuestión. En este caso, podremos realizar todas las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre estos elementos.

Tenemos en la pantalla principal, como vimos en la Figura B.1, tres opciones en este apartado: la primera (Gestionar producto de aprendizaje) que permite la gestión de objetivos, historias y criterios; otra que nos permite gestionar acciones de aprendizaje y evaluación; y una última para la gestión de plantillas y preguntas. A continuación nos centraremos en el funcionamiento de cada una de estas opciones, observando como permiten realizar las operaciones de CRUD sin problemas.

B.2.1. Gestionar producto de aprendizaje

Veamos cómo se pueden hacer todas las operaciones CRUD, para ello, nos basaremos en el ejemplo de los objetivos, siendo análogo para el caso de historias y criterios. Utilizaremos la

siguiente imagen para seguir la explicación:

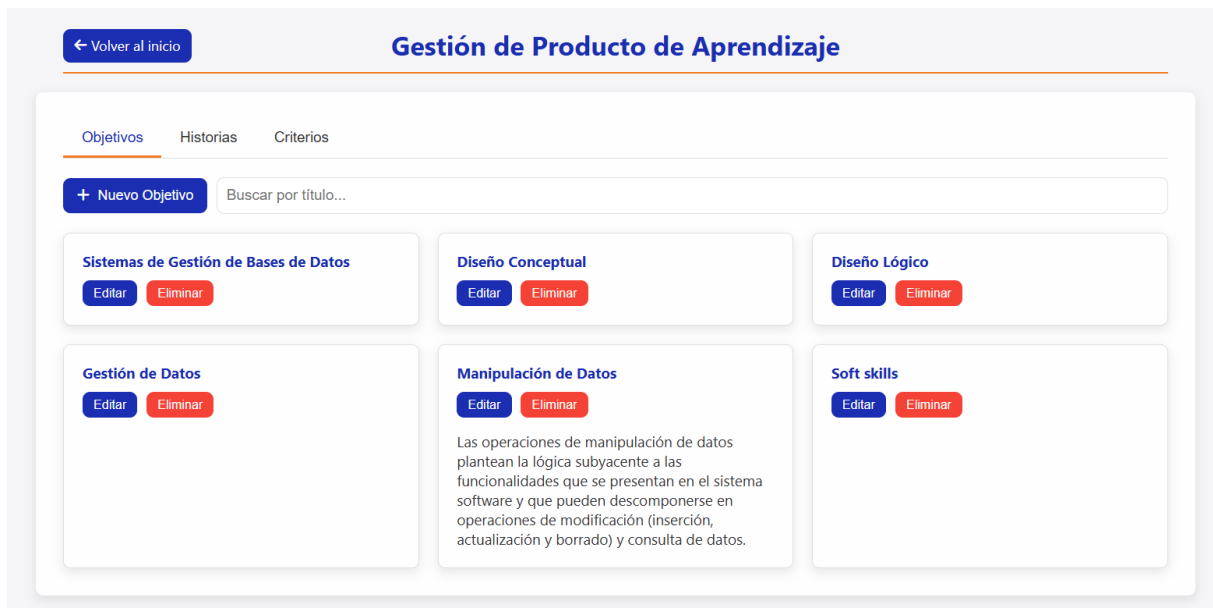


Figura B.7: Página para la gestión de los objetivos.

- **Crear:** Observando la Figura B.7, vemos que tenemos un botón para crear un nuevo objetivo. Pulsando en él nos aparecería la siguiente interfaz, que permite crear un objetivo nuevo.

Figura B.8: Interfaz para crear un nuevo objetivo.

- **Leer:** En la imagen mostrada en B.7 vemos como sin problema podemos encontrar la información de todos los objetivos disponibles. Además, podemos consultar la información adicional pulsando sobre ellos, como por ejemplo con el de la parte inferior central, en el que se muestra la especificación detallada.

- **Actualizar:** Observando de nuevo la Figura B.7 que hemos presentado antes, vemos que cada uno de los objetivos tiene un botón de edición. Esto nos permite actualizar la información de los mismos. La interfaz es igual que la que se encuentra en la Figura B.8 con la salvedad de que el texto estaría completo con la información disponible del objetivo seleccionado.
- **Eliminar:** De nuevo viendo la imagen B.7, de la misma forma que para la actualización, tenemos un botón de eliminar para cada uno de los objetivos que se muestran. Si se pulsa en este botón nos aparecerá un mensaje de alerta preguntando si de verdad queremos eliminarlo, pudiendo elegir si se quiere continuar con la eliminación o cancelar la misma.

B.2.2. Gestionar acciones

El funcionamiento es muy similar al visto en el caso del producto de aprendizaje, por lo que vamos a detallar en este caso las principales diferencias de funcionamiento con el anterior. En este caso, la interfaz que nos encontramos al entrar en este apartado es la siguiente:

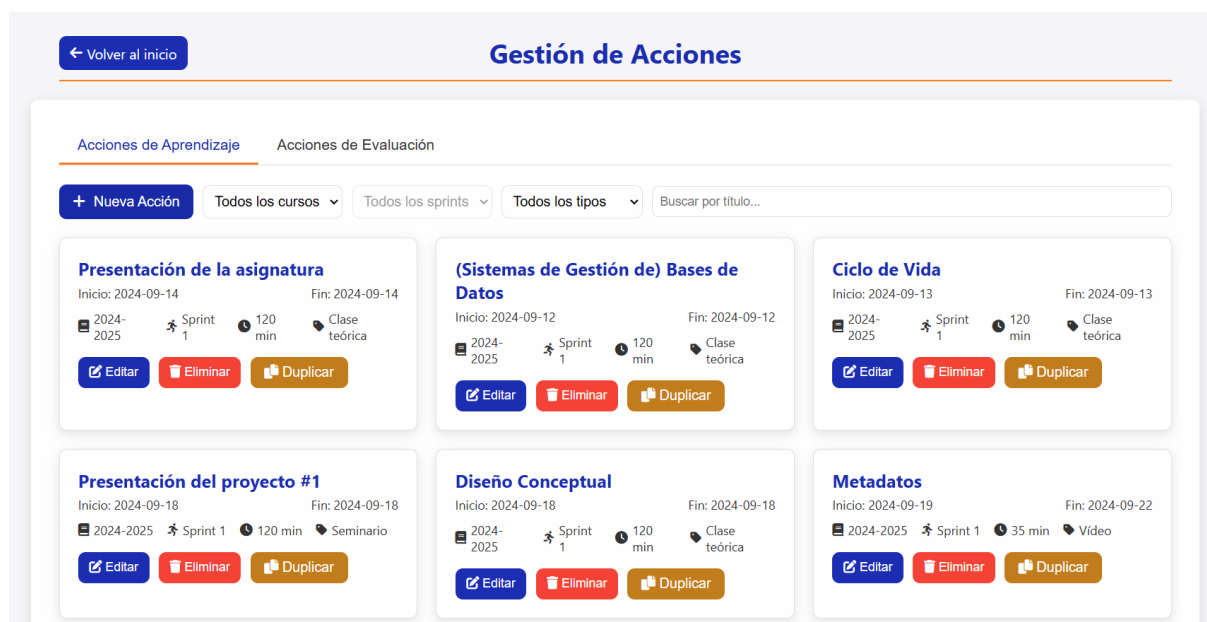
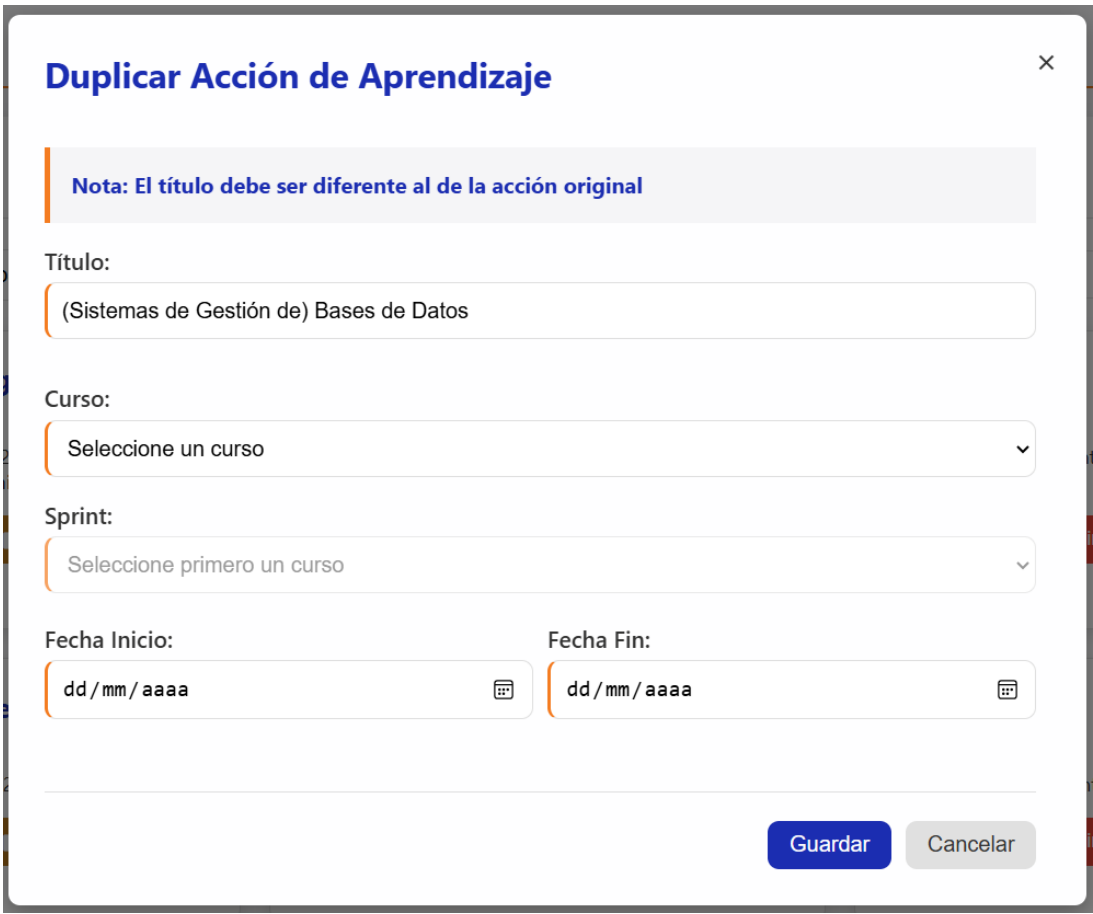


Figura B.9: Página para la gestión de las acciones de aprendizaje.

Los botones de “Nueva acción”, “Editar” y “Eliminar” permiten realizar acciones análogas a las explicadas en el caso anterior. Además, al pulsar en una acción se pueden consultar todos los datos de las mismas, con lo que se completan todas las operaciones de CRUD. Además, para el caso particular de las acciones tenemos el botón de “Duplicar”, que nos permite la creación de acciones que sean de la misma forma que las existentes pero que se quieran programar para un nuevo curso. La interfaz que aparece al pulsar en dicho botón es la siguiente:



Duplicar Acción de Aprendizaje ×

Nota: El título debe ser diferente al de la acción original

Título:
(Sistemas de Gestión de) Bases de Datos

Curso:
Seleccione un curso ▼

Sprint:
Seleccione primero un curso ▼

Fecha Inicio: dd/mm/aaaa 

Fecha Fin: dd/mm/aaaa 

Guardar **Cancelar**

Figura B.10: Duplicar una acción.

B.2.3. Crear plantillas y preguntas

El funcionamiento es totalmente análogo a los dos casos que hemos explicado anteriormente. En las Figuras B.11 y B.12 vemos la interfaz para el caso de las plantillas y preguntas.



Figura B.11: Página para la gestión de plantillas y preguntas.

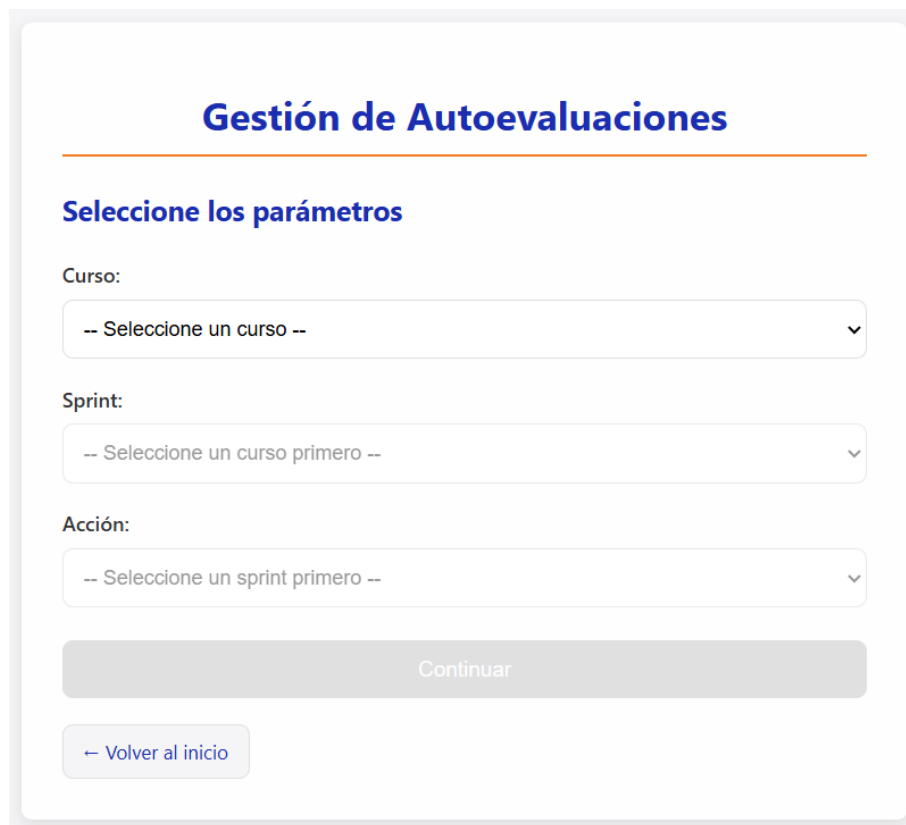
Criterios Evaluados			
Criterio	Umbral	Ponderación	Especificación
CONCEPTUAL-2.2. - ER - entidades	0.5	0.15	Cada entidad incorrecta restará en la misma proporción (1/X) que las entidades correctas.
CONCEPTUAL-2.3. - ER - entidades débiles	0.5	0.05	Cada entidad incorrecta restará en la misma proporción (1/X) que las entidades correctas.
CONCEPTUAL-2.4. - ER - atributos	0.5	0.1	Cada atributo incorrecto restará en la misma proporción (1/Y) que los atributos correctos.
CONCEPTUAL-2.5. - ER - identificadores	0.5	0.05	Cada identificador incorrecto restará en la misma proporción (1/X) que los identificadores correctos y, además invalidará la entidad correspondiente.
CONCEPTUAL-2.6. - ER - relaciones	0.5	0.2	Cada relación incorrecta restará en la misma proporción (1/Z) que las relaciones correctas.
CONCEPTUAL-2.7. - ER - multiplicidades	0.5	0.3	Cada multiplicidad incorrecta restará en la misma proporción (1/2Z) que las multiplicidades correctas.
CONCEPTUAL-2.8. - ER - modelo	1	0.05	-

Figura B.12: Criterios evaluados en una plantilla.

B.3. Procesamiento de datos

En la parte de procesamiento de datos tenemos cuatro opciones, una para cada uno de los archivos que se pueden procesar automáticamente con la aplicación. El funcionamiento de estos cuatro es muy similar, por lo que mostraremos el flujo de uno de ellos.

1. Lo primero que debemos hacer es completar el formulario en el que se nos indica que debemos seleccionar un curso, *sprint* y en algunas ocasiones la acción para la que se quiere realizar el procesamiento de los datos.



The screenshot shows a web form titled "Gestión de Autoevaluaciones" in blue text. Below the title is a section header "Seleccione los parámetros" in blue. There are three dropdown menus: "Curso:" with the placeholder "-- Seleccione un curso --", "Sprint:" with the placeholder "-- Seleccione un curso primero --", and "Acción:" with the placeholder "-- Seleccione un sprint primero --". Each dropdown has a downward arrow icon. Below the dropdowns is a large grey button labeled "Continuar". At the bottom left is a smaller button labeled "← Volver al inicio".

Figura B.13: Primera página del procesamiento de datos.

2. El segundo y último paso para completar el procesamiento de datos es seleccionar el archivo que se va a procesar. Siguiendo las instrucciones que aparecen en pantalla, como podemos ver en la Figura B.14, podemos cargar el archivo que se indica para comenzar con el procesamiento de los datos.

Gestión de Autoevaluaciones

[◀ Atrás](#)

Selección realizada:

- **Curso:** 2024-2025
- **Sprint:** Sprint 1
- **Acción:** Supuestos Prácticos sobre Diseño Conceptual (I) - 2023

Por favor, seleccione el archivo ZIP descargado del campus virtual que contiene las autoevaluaciones. El ZIP debe contener una carpeta por cada alumno, y dentro de cada carpeta los archivos Excel con las autoevaluaciones correspondientes.

Seleccione el archivo ZIP con autoevaluaciones:

Elegir archivo

No se ha seleccionado ningún archivo

El archivo debe ser un ZIP con las autoevaluaciones

Ningún archivo seleccionado

Procesar Autoevaluaciones

[← Volver al inicio](#)

Figura B.14: Segunda página del procesamiento de datos.

B.4. Gestión avanzada

Por último, tenemos la sección de “Gestión avanzada” en la que podemos consultar *dashboards* que se generarán a partir de la información que se ha cargado con las opciones explicadas en el apartado anterior. Tenemos dos opciones disponibles, una de ellas que nos permite ver la evolución de los criterios por alumno y otra en la que se puede consultar un resumen similar pero agregado para todos los estudiantes que han cursado la asignatura en el curso seleccionado.

Ambos casos tienen un funcionamiento similar, por lo que mostraremos el ejemplo de uno de ellos.

1. Para comenzar, debemos seleccionar un curso y una historia entre las disponibles. Se mostrarán todos los resultados de acciones de evaluación realizadas durante la asignatura en dicho curso, y que están asociadas a la historia elegida.

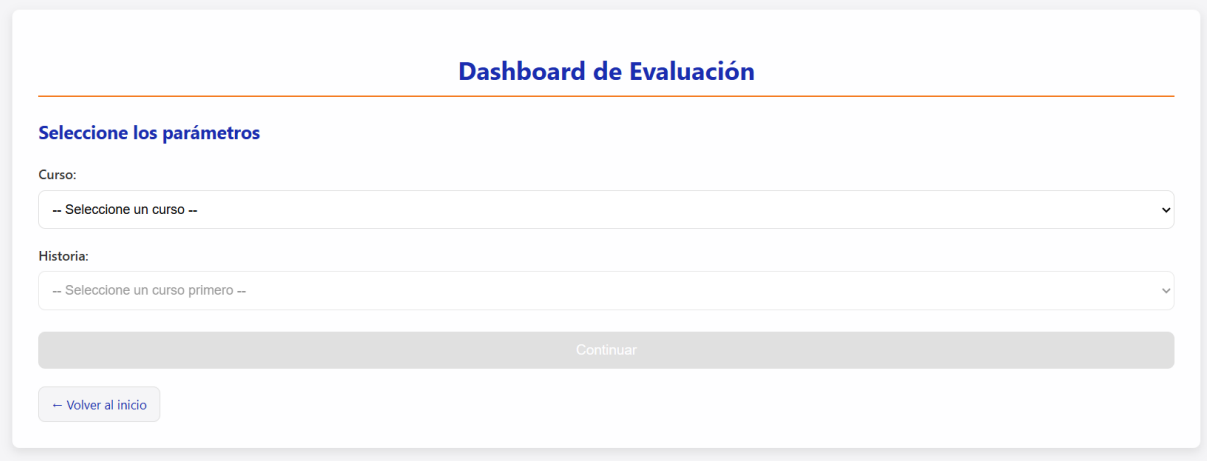


Figura B.15: Selección de curso y acción previas.

2. En la siguiente pantalla que nos aparece, debemos seleccionar uno de los estudiantes que han cursado la asignatura en el curso seleccionado. Una vez seleccionado dicho estudiante, aparecerá automáticamente el resumen de la evolución del estudiante en el aprendizaje de los criterios que forman la historia seleccionada.

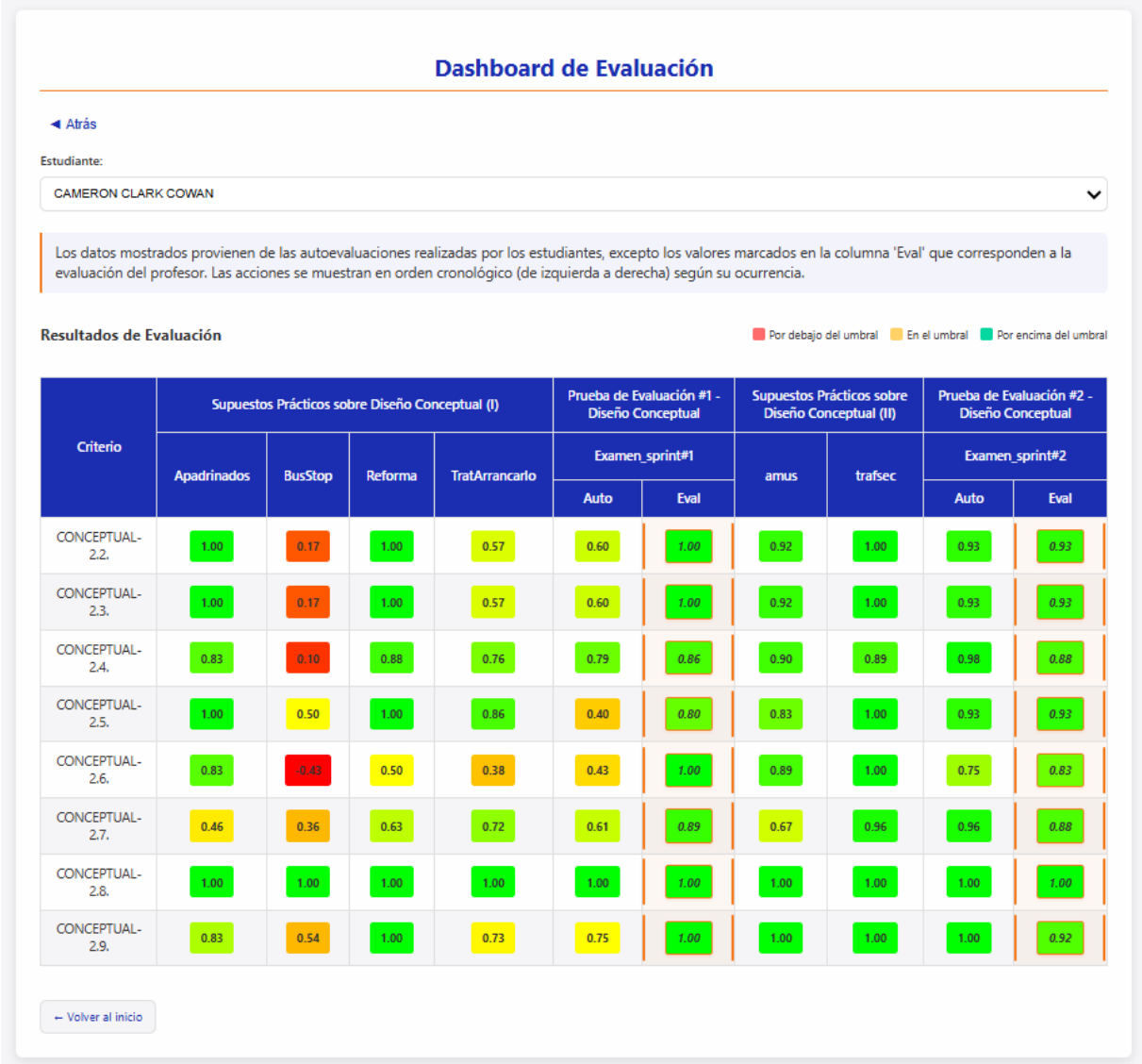


Figura B.16: Selección de estudiante y *dashboard*.