



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**PlanLMS
Aplicación multiplataforma
para la gestión de tareas de la
plataforma Moodle
Documentación Técnica**



Presentado por Javier Pampliega García
en Universidad de Burgos — 4 de julio de 2025
Tutor: Raúl Marticorena Sánchez

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	7
Apéndice B Especificación de Requisitos	15
B.1. Introducción	15
B.2. Objetivos generales	15
B.3. Catálogo de requisitos	16
B.4. Especificación de requisitos	18
Apéndice C Especificación de diseño	35
C.1. Introducción	35
C.2. Diseño de datos	35
C.3. Diseño arquitectónico	40
C.4. Diseño procedimental	42
Apéndice D Documentación técnica de programación	47
D.1. Introducción	47
D.2. Estructura de directorios	47
D.3. Manual del programador	48

D.4. Compilación y ejecución del proyecto	50
Apéndice E Documentación de usuario	53
E.1. Introducción	53
E.2. Requisitos de usuarios	53
E.3. Instalación	54
E.4. Manual del usuario	54
Apéndice F Anexo de sostenibilización curricular	67
F.1. Introducción	67
F.2. Competencias de sostenibilidad adquiridas	67
F.3. Sostenibilidad ambiental	68
F.4. Sostenibilidad social y accesibilidad	68
F.5. Sostenibilidad económica y mantenimiento	69
F.6. Conclusión	69
Bibliografía	71

Índice de figuras

B.1. Diagrama general de los casos de uso	20
C.1. Diccionario de datos	36
C.2. Diagrama de Modelo de datos	37
C.3. Diseño arquitectónico del proyecto	41
C.4. Diagrama de secuencia del inicio de sesión	45
C.5. Diagrama de secuencia de la creación de tareas	46
C.6. Diagrama de secuencia de la eliminación de tareas	46
D.1. Botón de acceso a la configuración de emuladores	50
D.2. Botón para añadir nuevos emuladores	50
D.3. Ventana con imágenes del sistema	51
D.4. Ventana de configuración general del emulador	51
D.5. Ventana de opciones de usuario	52
D.6. Lista con todos los emuladores disponibles	52
E.1. Inicio de sesión de PlanLMS	55
E.2. Diálogo para conectarse a un Moodle	56
E.3. Ejemplo de conexión	56
E.4. Pantalla principal de PlanLMS	57
E.5. Diálogo de filtros	58
E.6. Filtrado de cursos	58
E.7. Filtrado de actividades	58
E.8. Pantalla de diagrama de Gantt	60
E.9. Diálogo de actividad en diagrama de Gantt	60
E.10. Filtros del diagrama de Gantt	60
E.11. Filtros estéticos aplicados sobre el diagrama	60
E.12. Sección de actividades del diagrama de Gantt	61

E.13.Ventana con información de la actividad	62
E.14.Sección de tareas pendientes	63
E.15.Sección de tareas completadas	63
E.16.Formulario de creación de tareas personales	63
E.17.Tarea personal en sección de <i>Pendientes</i>	63
E.18.Panel con información de la tarea personal	64
E.19.Opción de borrado de la tarea personal	65

Índice de tablas

A.1. Porcentajes de cotización para empresas	9
A.2. Costes software	10
A.3. Costes hardware	10
A.4. Licencias de uso de software	12
B.1. A01 - Usuario	18
B.2. A02 - API de Moodle	19
B.3. A03 - Supabase	19
B.4. CU-1 Inicio de sesión	20
B.5. CU-2 Configuración de conexión a Moodle	21
B.6. CU-3 Navegación por la pantalla principal	22
B.7. CU-4 Visualización del diagrama de Gantt	23
B.8. CU-4.1 Navegación por la pantalla principal	24
B.9. CU-4.2 Cambiar estética del diagrama de Gantt	25
B.10.CU-5 Visualización de actividades en la sección del diagrama de Gantt	26
B.11.CU-5.1 Visualizar tarjeta de la actividad	27
B.12.CU-6 Visualización de tareas personales	28
B.13.CU-6.1 Visualizar información de la tarea personal	29
B.14.CU-6.2 Cambiar estado de finalización de la tarea personal	30
B.15.CU-6.3 Visualizar información de la tarea personal	31
B.16.CU-7 Creación de tareas personales	32
B.17.CU-8 Filtrado del contenido de la aplicación	33
B.18.CU-9 Cierre de sesión	34
C.1. Modelo UserModel	38
C.2. Modelo Courses	38
C.3. Modelo Assign	38

C.4. Modelo Quiz	39
C.5. Modelo Submission	39
C.6. Modelo QuizGrade	39
C.7. Modelo PersonalTask	40
C.8. Modelo TaskPriority	40

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Este apéndice se centra en explicar la planificación que se ha llevado a cabo para desarrollar el proyecto. Se explorará la planificación temporal, estudio de viabilidad, viabilidad económica y legal. El objetivo es que el proyecto finalice en el plazo establecido, con los recursos necesarios y cumpliendo la normativa.

A.2. Planificación temporal

La planificación temporal del proyecto fue dividida por *sprints* de una duración de una o dos semanas. En estos *sprints* se definían objetivos a completar durante el transcurso de las semanas y se entregaban los resultados en la finalización del *sprint*, donde se realizaba una retroalimentación y se definían los objetivos del siguiente *sprint*.

Sprint 1 (05/02/2025 - 19/02/2025)

En el primer *sprint* se realizó la configuración inicial del proyecto así como se creó el repositorio del proyecto en GitHub. Una vez realizados estos dos pasos, se procedió con la implementación del prototipo trabajado los anteriores meses. En este, se incluía el inicio de sesión de la aplicación, una pantalla principal con un botón y la pantalla del diagrama de Gantt. Además de, la implementación de servicios para obtener datos de la API de Moodle.

A parte de los objetivos anteriores, se propuso la implementación de un filtro para el diagrama de Gantt. En esta primera versión del sistema de filtrado se permitía al usuario filtrar por cursos y por tipo de actividad.

Sprint 2 (19/02/2025 - 06/03/2025)

En el segundo *sprint*, el objetivo era aplicar mejoras sobre la interfaz del sistema de filtrado del diagrama de Gantt. Además, se implementó la posibilidad de que el usuario se pudiese conectar a cualquier servidor Moodle. A su vez, se realizó un cambio de librería de diagrama de Gantt, con el fin de mejorar el aspecto visual así como la representación de actividades.

En este *sprint* se incluyó una librería que iba a ser muy importante en el futuro, *SharedPreferences*. El objetivo en ese momento era almacenar la URL del servidor Moodle al que el usuario quería conectarse.

Sprint 3 (06/03/2025 - 19/03/2025)

En este *sprint* el principal objetivo era arreglar un error de visualización en el diagrama de Gantt. Este error consistía en que el diagrama solo podía mostrar un número limitado de actividades, ya que la altura de este limitaba poder mostrar todas. En la reunión se plantearon dos opciones:

- Modificar la librería o implementar alguna función que calcule la altura.
- Utilizar un *WebView* para tener acceso a un diagrama de Gantt desarrollado en JavaScript.

Finalmente, se optó por crear una función que calculara la altura del diagrama en función del número de actividades que se iban a visualizar.

Por otra parte, se añadieron las siguientes mejoras respecto al diagrama:

- Asignación de colores a los cursos.
- Funciones para asignación de fechas, ordenación de tareas y cuestionarios.
- Admisión de actividades vacías, corrección de fechas, cambio de estilos, etc.

Por último, se añadió un botón que permitía la visualización de la contraseña en el campo de texto de la pantalla de inicio de sesión.

Sprint 4 (19/03/2025 - 02/04/2025)

En el cuarto *sprint* la atención se centró en la resolución de unos *bugs* relacionados con el diagrama y su sección de actividades. Los *bugs* a solucionar eran:

- Mal adjudicada una función que asignaba la fecha errónea.
- Reajuste de los valores a retornar en la función que obtenía las fechas de cierre.

Estos problemas afectaban sobre la visualización de las actividades en el diagrama de Gantt. El otro *bug* a corregir afectaba sobre el filtrado de actividades, que aun teniendo desmarcada la opción del tipo de actividad, se seguían mostrando por pantalla.

Una vez resueltos los *bugs*, se procedió a implementar una línea vertical que indicara la fecha actual, nuevas entidades para la sección de actividades y nuevos datos (tiempo restante y calificación) en las tarjetas de las actividades.

Por último, se esperaba implementar una nueva funcionalidad que permitiera a los usuarios establecer sus propias tareas personales y que estas se almacenaran en una base de datos en la nube.

Sprint 5 (02/04/2025 - 16/04/2025)

En este *sprint*, el primer objetivo era solucionar el error que provocaba Firebase. Este error no permitía la correcta depuración de la aplicación, debido a la ausencia de unos archivos de depuración. Se trato de solucionar este error recurriendo a las experiencias de otros usuarios, pero el resultado fue el mismo, no funcionaba. Por lo tanto, se optó por la segunda opción planteada, cambiar a la librería de Supabase (alternativa de Firebase). Una vez solucionado este error, se procedió con la implementación de la ventana de tareas personales, además de, un ventana con un formulario para la creación de las tareas personales del usuario.

Por otra parte, se añadió un nuevo filtro que permitía filtrar las actividades en función del rango de fechas en el que se situaban. En relación a los filtros, se añadió la persistencia de los mismos, de forma que estos se almacenaban de forma local en el dispositivo del usuario.

Todos estos cambios e implementaciones fueron acompañados con la corrección de *bugs* del *sprint* anterior, y de los nuevos *bugs* que iban apareciendo conforme se implementaban nuevas funciones.

Sprint 6 (16/04/2025 - 30/04/2025)

El objetivo principal de este *sprint* era la mejora de la interfaz de la pantalla de tareas personales. Entre las mejoras a implementar se destacaba:

- Tarjetas para la representación de tareas.
- Distribución de tareas por columnas.
- Botón para cambiar el tipo de ordenación: temporal o por curso.

En este punto de la aplicación, al disponer de dos funcionalidades que requerían de un filtro, se llegó a la conclusión de que la mejor opción era mover el filtro del diagrama a la pantalla de inicio para convertirlo en un filtro general que actuara sobre toda la aplicación.

A parte de mover el filtro a la pantalla principal, se añadió otro filtro que permitía a los usuarios filtrar las actividades en función de las fechas disponibles, es decir, si las actividades estaban configuradas con fecha de inicio y cierre.

Sprint 7 (30/04/2025 - 14/05/2025)

Para este *sprint* los objetivos principales eran implementar dos funcionalidades básicas:

- Cierre de sesión.
- Calificaciones de cuestionarios.

Hasta la fecha, la aplicación no respondía ante fallos imprevistos, por lo tanto, simplemente se bloqueaba y no permitía al usuario avanzar. Es por ello, que en este punto, se planteo la incorporación de robustez en distintas funcionalidades para evitar este problema. Se aplicó robustez en:

- **Inicio de sesión:** comprobación de credenciales de acceso.
- **Conexión a Moodle:** verificación de que la URL sea un servidor Moodle.

Por otro lado, se decidió reorientar la aplicación hacia dispositivos móviles, dejando más de lado el diseño para ordenadores. En primeras instancias se rediseñó:

- Inicio de sesión.
- Pantalla principal.
- Pantalla del diagrama de Gantt.

A su vez, se fue refactorizando el código para ordenarlo en distintos directorios, para facilitar su mantenimiento en el futuro así como su escalabilidad.

Por último, se implementó la obtención y visualización de calificaciones de cuestionarios en la sección de actividades del diagrama de Gantt. Además, se implementó el cierre de sesión desde la pantalla principal.

Sprint 8 (14/05/2025 - 28/05/2025)

En el octavo *sprint*, se implementaron mejoras sobre la pantalla del diagrama de Gantt. Estas mejoras son:

- Panel deslizable con configuraciones del diagrama.
- Diagrama interactivo con posibilidad de hacer *zoom* y moverlo en todas direcciones.
- Cambio en la posición de las líneas temporales.

A nivel general, se fueron moviendo de directorios los distintos componentes de la aplicación con el fin de facilitar su mantenimiento y escalabilidad.

En cuanto a la pantalla de tareas personales, se implementó un formulario que permitía la creación de tareas personales a la vez que se fueron eliminando los anteriores componentes, ya que el diseño original fue completamente modificado. Las características del nuevo diseño eran:

- Visualización de las tareas en formato lista.
- Distintos tipos de pestañas en función del estado de finalización y el tiempo.
- Ordenación de tareas en pestañas de temporalidades.
- Tarjetas deslizables con opción de borrado y, también presionables para ver toda la información de la tarea.

- Implementación de niveles de prioridad.

La implementación de un formulario, obligó a reforzar la robustez de la aplicación. Por ello, se aplicó la obligación de llenar algunos campos del formulario, en el caso de que no fueran llenados no permitiría al usuario la creación de la tarea. Posteriormente se añadió el filtrado a la pantalla de tareas personales y se corrigieron *bugs*.

Finalmente, en este *sprint* se lanzó la primera *pre-release* de la aplicación con las siguientes funcionalidades:

- Filtro general para toda la aplicación.
- Diagrama de Gantt para actividades de Moodle.
- Pantalla de actividades del diagrama de Gantt.
- Creación y gestión de tareas personales.

Sprint 9 (28/05/2025 - 04/06/2025)

En este *sprint*, el objetivo principal era dar solución a los *bugs* de la versión 0.1, estos *bugs* eran:

- Error en la asignación de colores a los cursos del diagrama.
- Error en el filtrado por rango de fechas del diagrama.
- Error en el filtrado de fechas, ya que no se seleccionaban las tareas que se encontraban en el límite.

A parte de la corrección de *bugs*, se realizaron procesos de refactorización del *backend* de la pantalla principal. Además, se llevaron a cabo ciertas modificaciones que mejoraron el aspecto visual de la aplicación.

En este punto, se empezó a dar más atención a la redacción de las memorias y aspectos de documentación.

Sprint 10 (04/06/2025 - 12/06/2025)

A lo largo de esta semana me centré en la redacción de las memorias, se redactaron los apartados más importantes, entre los que se encontraban:

- Conceptos teóricos.
- Aspectos relevantes.
- Trabajos relacionados.
- Conclusiones y Líneas de trabajo futuras.

Por otra parte se empezó a refactorizar el *backend* del inicio de sesión para moverlo a un fichero a parte.

Sprint 11 (12/06/2025 - 23/06/2025)

En este *sprint* se terminó de refactorizar el *backend* del inicio de sesión.

Por otra parte, se continuó la redacción de las memorias y a su vez se comenzó la redacción de los anexos.

Por último, se corrigió un *bug* que no permitía guardar los filtros de múltiples usuarios, y por lo tanto, la aplicación colapsaba. Además, se corrigieron otros *bugs* que se encontraron.

Sprint 12 (23/06/2025 - 02/07/2025)

Para este *sprint*, lo único que quedaba era finalizar los anexos y revisar toda la documentación. Además, se revisó el código en búsqueda de mejoras o errores. Por otra parte, se realizaron cambios en el repositorio del proyecto para mejorar la información (*README* Y licencia) que se proporciona en el mismo.

Sprint 13 (02/07/2025 08/07/2025)

En este último *sprint*, se finalizó toda la documentación y se llevó a cabo una revisión general del proyecto con el objetivo de corregir errores. También, se realizaron las últimas modificaciones de código y se lanzó la *release v1.0*

A.3. Estudio de viabilidad

En este apartado se van a estudiar dos aspectos de viabilidad:

- **Económica:** aspectos económicos que permiten saber si el proyecto es viable económicoamente.

- **Legal:** aspectos legales que permiten saber si el proyecto cumple con todas las leyes.

Si el estudio de viabilidad muestra un resultado positivo, el proyecto puede ser desarrollado sin el temor a no cumplir con los mínimos requeridos.

Viabilidad económica

En este apartado se hará un estudio de viabilidad económica, es decir, se estudiará en profundidad tanto de los costes como de los beneficios. Con el fin de abordar la rentabilidad del proyecto.

El primer paso es estudiar los costes que se pueden generar en torno al desarrollo del proyecto. El objetivo es tener una idea general del coste total del proyecto para posteriormente compararlo con los beneficios y concluir la rentabilidad del desarrollo. Para ello, se van a dividir los costes por grupos con el fin de distinguirlos de forma clara.

Empleados

El primer coste a estudiar, es el coste humano, ya que sin un desarrollador este proyecto no se puede llevar a cabo. En este caso, el desarrollador ha sido el alumno, es decir yo. Por otra parte hay que tener en cuenta al tutor, por la realización de tutorías para guiar al alumno en todas las partes del proyecto. Teniendo localizados a los empleados, se va a dividir y calcular los costes de cada uno.

Empezando por el desarrollador, hay que tener en cuenta el sueldo medio de un programador junior en España, en este caso el sueldo se sitúa entre los 18.000€ y 24.000€, se toma la media entre ambos y sale 21.000€. Otro factor a tener en cuenta son las horas de trabajo llevadas a cabo por el alumno, que son unas 400 horas. En total se han trabajado 5 meses en el proyecto, por lo tanto, 80 horas mensuales.

$$21,000 \frac{\text{€}}{\text{año}} * \frac{1 \text{ año}}{12 \text{ meses}} * \frac{1 \text{ mes}}{160 \text{ horas de trabajo}} = 10,94 \frac{\text{€}}{\text{hora}} \quad (\text{A.1})$$

Teniendo calculado el sueldo por hora de una jornada laboral completa, basta con multiplicarlo por las horas de trabajo empleadas.

$$10,94 \frac{\text{€}}{\text{hora}} * 400 \text{ horas} = 4376 \text{ €} \quad (\text{A.2})$$

En cuanto al sueldo de tutor, se toma como referencia un sueldo medio de 40€/hora y una duración media de una hora por reunión. Por lo tanto, se calcula el gasto de tutor:

$$40 \frac{\text{€}}{\text{reunión}} * 11 \text{ reuniones} = 440 \text{ €} \quad (\text{A.3})$$

Ahora hay que tener en cuenta que como empresa hay que dar de alta a los empleados en la Seguridad Social [5], otro factor a tener en cuenta como coste. Por lo tanto, al empresa tendría que pagar un salario bruto teniendo en cuenta el Régimen General de la Seguridad Social:

Concepto	Empresa
Contingencias	23.60 %
Desempleo	5.50 %
FOGASA	0.20 %
Formación profesional	0.60 %

Tabla A.1: Porcentajes de cotización para empresas

$$4376 \text{ €} + 440 \text{ €} = 4816 \text{ €} \quad (\text{A.4})$$

$$4816 \text{ €} * (1 + (0,236 + 0,055 + 0,002 + 0,006)) = 6255,98 \text{ €} \quad (\text{A.5})$$

Por lo tanto, el coste humano total del proyecto es de 6255.98€.

Software

En cuanto al coste software, el desarrollo del proyecto se ha llevado a cabo con la utilización de herramientas gratuitas y de código abierto. Esto ha permitido reducir significativamente los costes del proyecto. Los costes software han sido:

Software	Licencia	Coste
Visual Studio Code	Gratuita	0€
Postman	Plan gratuito	0€
Android Emulator	Gratuita	0€
Overleaf	Plan gratuito	0€
Dart	Open Source	0€
Flutter SDK	Open Source	0€
Supabase	Plan gratuito	0€
Librerías	Open Source	0€
Total		0€

Tabla A.2: Costes software

En el caso de requerir de más espacio en la nube, se podría considerar la suscripción a los distintos planes que ofrece la plataforma.

El coste total de los productos software ha sido de un total de 0€.

Hardware

En cuanto al coste del *hardware*, hay que distinguir entre los distintos elementos *hardware*:

Hardware	Coste	Coste amortizado
Ordenador portátil	1800€	107.14€
Dispositivo móvil Android	845€	58.68€
Tablet Android	800€	55.55€
Periféricos	50€	4.16€
Total	3495€	225.53€

Tabla A.3: Costes hardware

El coste amortizado se calcula con la siguiente fórmula:

$$\text{Coste Amortizado} = \frac{\text{Coste Adquisición} - \text{Coste Final}}{\text{Vida Útil}} * \text{Meses Proyecto}$$
(A.6)

El uso que se le ha dado a cada componente *hardware* varía en función de la labor:

- **Ordenador portátil:** su uso ha sido fundamental para el uso de herramientas y desarrollo del proyecto.
- **Dispositivo móvil Android:** ha permitido realizar pruebas de depuración en un dispositivo físico.
- **Tablet Android:** ha permitido realizar pruebas de depuración así como su uso de segunda pantalla para agilizar el desarrollo.
- **Periféricos:** los días de desarrollo en casa han sido agilizados por el uso de un teclado y un ratón.

Por lo tanto, el coste total del *hardware* ha sido de 225,53€.

Por otra parte hay que tener en cuenta los costes extras, en este caso se tiene en cuenta el coste de *Internet*, un servicio básico puede tener un coste de 30€ por mes. Teniendo en cuenta los meses de trabajo, se obtiene un gasto de 180€ en *Internet*.

Teniendo en cuenta todos los costes anteriores, obtenemos que el gasto total a cubrir del proyecto es de:

$$6255,98 \text{ €} + 0 \text{ €} + 225,53 \text{ €} + 180 \text{ €} = 6661,51 \text{ €} \quad (\text{A.7})$$

Viabilidad legal

En este apartado del proyecto se contemplan todos los aspectos legales con el fin de cumplirlos, especialmente en aquellos relacionados con el tratamiento de datos y licencias de uso de software.

Protección de datos

Se tiene en cuenta que dentro de la aplicación se accede a información personal del usuario, como cursos, tareas, entregas o calificaciones entre otras, procedentes de su cuenta de Moodle. Por lo tanto, se deben tener en cuenta las siguientes consideraciones:

- Se garantiza que se cumplen con los estándares establecidos en la Reglamento General de Protección de Datos, garantizando la confidencialidad, integridad y disponibilidad de los datos

- Los datos se almacenan localmente en el dispositivo del usuario o en los servidores de Supabase que cumplen con las medidas de seguridad necesarias para la protección de datos

Licencias de uso de software

En este caso se van a estudiar las licencias de uso de las distintas librerías y herramientas utilizadas en el desarrollo del proyecto.

Librería/Herramienta	Licencia
calendar_date_picker2	Apache-2.0
cupertino_icons	MIT
Dart	BSD-3-Clause
Flutter SDK	BSD-3-Clause
flutter_dotenv	MIT
flutter_localizations	BSD-3-Clause
flutter_slidable	MIT
flutter_widget_from_html	MIT
http	BSD-3-Clause
intl	BSD-3-Clause
shared_preferences	BSD-3-Clause
sliding_up_panel	unknown
supabase_flutter	MIT
url_launcher	BSD-3-Clause

Tabla A.4: Licencias de uso de software

A continuación se procede a explicar las licencias que aparecen en la tabla anterior:

- **Apache-2.0:** requiere de la conversación del aviso de derechos de autor y el descargo de responsabilidad. No requiere de la redistribución del código fuente cuando se distribuyen versiones modificadas. Por lo tanto, permite usar para cualquier propósito el software, redistribuirlo, modificarlo y distribuir versiones modificadas [6].
- **BSD-3-Clause:** es una licencia de código abierto permisiva. Permite el libre uso, modificación y distribución del software [2]. Requiere de algunas condiciones:

- Se debe de incluir un exención de responsabilidad en toda distribución
- Se debe incluir un aviso de copyright
- **MIT:** concede el permiso sin cargo a cualquier persona que obtenga una copia del software, para tratar el software sin restricción, incluyendo los derechos a usar, copiar, modificar, fusionar, publicar, distribuir, sublicenciar o vender copias [4]. Todo ello bajo unas condiciones: El aviso de derechos de autor y el aviso de permiso se incluirán en todas las copias o partes sustanciales del software.

Por lo tanto, todas las herramientas empleadas en el proyecto se han usado bajo licencias permisivas, lo que quiere decir que, se permite su uso dentro del marco legal de desarrollo de software.

Por último, la licencia seleccionada para regular el uso, modificación y distribución de este proyecto es la licencia MIT.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice, se profundizará en los objetivos y requisitos que debe de cumplir el proyecto. A continuación se detallarán los siguientes aspectos:

- **Objetivos generales:** se plasman los objetivos generales que se pretenden alcanzar con el proyecto.
- **Catálogo de requisitos:** se listan todos los requisitos que definen el comportamiento de la aplicación.
- **Especificación de requisitos:** se detallan los casos de uso para facilitar la comprensión de las interacciones del usuario con la aplicación.

B.2. Objetivos generales

En este proyecto, se han definido unos objetivos que marcan la trayectoria que se debe de seguir para el desarrollo de este:

- Visualización de las actividades de Moodle en un diagrama de Gantt
- Creación de tareas personales asociadas a un curso del usuario
- Filtrado general de toda la aplicación

B.3. Catálogo de requisitos

En este apartado, se definirán todos los requisitos funcionales y no funcionales del proyecto.

Requisitos funcionales

- **RF1. - Inicio de sesión:** el usuario tiene que poder acceder a la aplicación, mediante el correo y contraseña de su plataforma Moodle. Además, podrá habilitar una opción para guardar su correo. También, se debe de proporcionar una opción de recuperación de contraseña.
 - **RF1.1. - Recuperación de contraseña:** la aplicación tiene que permitir al usuario recuperar su contraseña en caso de olvido.
 - **RF1.2. - Guardar correo de inicio de sesión:** el usuario tiene que poder guardar el correo con el que realizar el inicio de sesión.
- **RF2. - Conexión con plataforma Moodle:** el usuario tiene que poder conectarse a una plataforma Moodle. Además, la aplicación verificará que se trata de un servidor Moodle.
- **RF3. - Cierre de sesión:** el usuario tiene que poder cerrar sesión en la aplicación.
- **RF4. - Acceso a todas las funcionalidades desde la pantalla principal:** el usuario tiene que poder acceder a todas las funcionalidades de la aplicación desde la pantalla principal.
- **RF5. - Visualización de actividades en el diagrama de Gantt:** el usuario tiene que poder visualizar todas sus actividades (filtradas o sin filtrar) en el diagrama de Gantt. También, tendrá que poder ver una información mínima de cada actividad, así como modificar la estética del diagrama con filtros.
 - **RF5.1. - Ver información de la actividad:** la aplicación tiene que permitir al usuario ver una previsualización de la actividad desde el propio diagrama.
 - **RF5.2. - Cambiar estética del diagrama:** la aplicación tiene que permitir al usuario modificar la estética del diagrama de Gantt.

- **RF6. - Visualización de actividades en la sección de actividades del diagrama:** el usuario tiene que poder ver las mismas actividades del diagrama en el sección de actividades, pero con más información que pueda resultar relevante.
 - **RF6.1. - Visualizar tarjeta de la actividad:** la aplicación tiene que permitir al usuario visualizar toda la información disponible de la actividad.
- **RF7. - Visualización de tareas personales:** el usuario tiene que poder ver todas sus tareas personales repartidas en las dos secciones disponibles, así como la información de cada tarea.
 - **RF7.1. - Visualizar información de tarea personal:** la aplicación tiene que permitir al usuario visualizar toda la información de la tarea personal.
 - **RF7.2. - Marcar realización de tarea personal:** la aplicación tiene que permitir al usuario cambiar el estado de finalización de la tarea personal.
 - **RF7.3. - Borrar tarea personal:** la aplicación tiene que permitir al usuario eliminar la tarea personal.
- **RF8. - Creación de tareas personales:** el usuario tiene que poder crear sus propias tareas personales.
- **RF9. - Filtrar contenidos de la aplicación:** el usuario tiene que poder filtrar todos los contenidos de la aplicación, tanto actividades de Moodle como tareas personales.

Requisitos no funcionales

- **RNF1. - Disponibilidad:** la aplicación debe de garantizar su completa disponibilidad, independientemente de la ubicación del usuario.
- **RNF2. - Usabilidad:** la aplicación debe de garantizar al usuario un uso sencillo e intuitivo.
- **RNF3. - Mantenimiento:** la aplicación debe de aportar facilidad en el mantenimiento para la corrección de errores.
- **RNF4. - Escalabilidad:** la aplicación debe de ser capaz de gestionar un crecimiento de usuarios sin afectar negativamente a su rendimiento.

- **RNF5. - Rendimiento:** la aplicación debe de tener tiempos de respuesta y cargas de datos aceptables, para que la experiencia de usuario sea fluida.

- **RNF6. - Seguridad:** la aplicación debe de garantizar la protección de los datos del usuario.

B.4. Especificación de requisitos

Actores del sistema

Actor	A01
Nombre	Usuario
Versión	1.0
Autor	Javier Pampliega García
Descripción	Persona que hace uso de la aplicación para planificar sus actividades Moodle y tareas personales.
Tipo	Usuario
Objetivo	Visualizar y planificar actividades Moodle, y crear y gestionar tareas personales.
Responsabilidades	<ul style="list-style-type: none"> ■ Iniciar sesión para conectarse a Moodle. ■ Usar el sistema de filtrado. ■ Visualizar diagrama de Gantt y actividades. ■ Crear y gestionar tareas personales.
Relaciones con casos de uso	

Tabla B.1: A01 - Usuario

Actor	A02
Nombre	API de Moodle
Versión	1.0
Autor	Javier Pampliega García
Descripción	
Tipo	Sistema
Objetivo	Obtener todos los datos necesarios para el funcionamiento de la aplicación.
Responsabilidades	<ul style="list-style-type: none"> ■ Obtención de los datos de Moodle del usuario del usuario.
Relaciones con casos de uso	

Tabla B.2: A02 - API de Moodle

Actor	A03
Nombre	Supabase
Versión	1.0
Autor	Javier Pampliega García
Descripción	Servicio en la nube que posibilita el almacenamiento de datos en la nube.
Tipo	Sistema
Objetivo	Insertar y borrar tareas personales del usuario en la nube.
Responsabilidades	<ul style="list-style-type: none"> ■ Almacenar las tareas creadas del usuario. ■ Cargar las tareas del usuario. ■ Eliminar del almacenamiento en la nube las tareas del usuario.
Relaciones con casos de uso	

Tabla B.3: A03 - Supabase

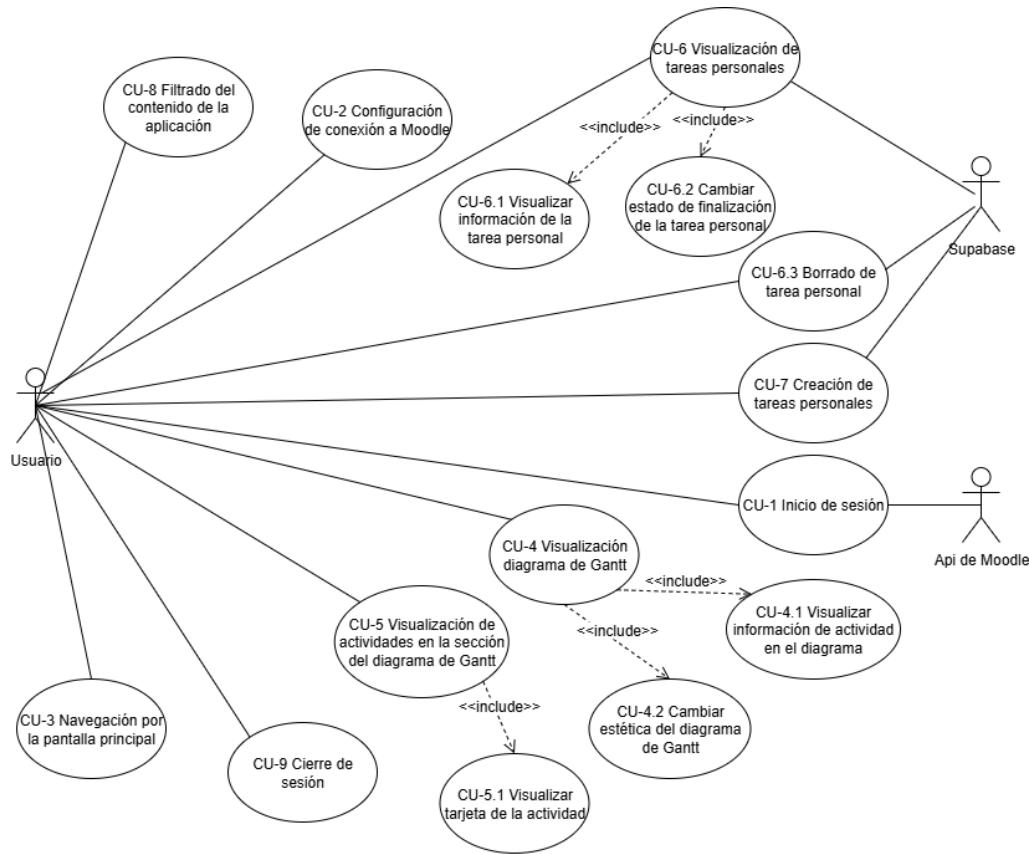


Figura B.1: Diagrama general de los casos de uso

Casos de uso

CU-1	Inicio de sesión
Versión	1.0
Actor	Usuario, API de Moodle
Autor	Javier Pampliega García
Requisitos asociados	RF-1
Descripción	Permitir iniciar sesión para acceder a la aplicación y para cargar los datos de Moodle.
Precondición	La aplicación instalada y abierta.
Acciones	<ol style="list-style-type: none"> 1. Estar conectado a una plataforma Moodle. 2. Introducir las credenciales de usuario. 3. Presionar el botón de inicio de sesión.
Postcondición	Inicio de sesión con la cuenta de Moodle del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Ausencia de plataforma Moodle configurada. ■ Credenciales incorrectas. ■ Ausencia de conexión a Internet.

CU-2 Configuración de conexión a Moodle	
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-2
Descripción	Permitir conectar con un servidor de Moodle.
Precondición	La aplicación instalada y abierta, y en el panel para conectar con un servidor Moodle.
Acciones	<ol style="list-style-type: none"> 1. Introducir la URL del servidor Moodle. 2. Esperar la verificación de la URL. 3. Presionar el botón Guardar.
Postcondición	Inicio de sesión con la cuenta de Moodle del usuario.
Excepciones	<ul style="list-style-type: none"> ▪ Que no se trate de un servidor Moodle. ▪ Ausencia de conexión a Internet.
Importancia	Alta

Tabla B.5: CU-2 Configuración de conexión a Moodle

CU-3	Navegación por la pantalla principal
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-4
Descripción	Permitir acceder a todas las funcionalidades de la aplicación desde la misma pantalla.
Precondición	Sesión iniciada correctamente.
Acciones	<ol style="list-style-type: none"> 1. Pulsar el botón de la funcionalidad a la que se quiera acceder.
Postcondición	Acceso a la funcionalidad seleccionada.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.6: CU-3 Navegación por la pantalla principal

CU-4	Visualización del diagrama de Gantt
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-5
Descripción	Permitir visualizar todas las actividades (filtradas o sin filtrar) en un diagrama de Gantt.
Precondición	Haber accedido desde la pantalla principal.
Acciones	<ol style="list-style-type: none"> 1. Desplazarse por el diagrama para ver todas las actividades.
Postcondición	No hay postcondiciones.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Alta

Tabla B.7: CU-4 Visualización del diagrama de Gantt

CU-4.1	Visualizar información de actividad en el diagrama
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-5, RF-5.1
Descripción	Permitir visualizar la información de una actividad directamente desde el diagrama de Gantt.
Precondición	Haber accedido a la pantalla del diagrama de Gantt.
Acciones	<ol style="list-style-type: none"> 1. Localizar la actividad a visualizar. 2. Presionar sobre el punto que representa a la actividad en el diagrama.
Postcondición	Se visualiza una ventana con la información de la actividad.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.8: CU-4.1 Navegación por la pantalla principal

CU-4.2	Cambiar estética del diagrama de Gantt
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-5, RF-5.2
Descripción	Permitir realizar cambios en la estética del diagrama para mejorar su visualización.
Precondición	Haber accedido a la pantalla del diagrama de Gantt.
Acciones	<ol style="list-style-type: none"> 1. Deslizar hacia arriba el panel inferior de la pantalla. 2. Modificar los filtros según las necesidades del usuario.
Postcondición	Cambios en la estética del diagrama de Gantt.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Baja

Tabla B.9: CU-4.2 Cambiar estética del diagrama de Gantt

CU-5	Visualización de actividades en la sección del diagrama de Gantt
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-6
Descripción	Permitir visualizar en formato tarjeta todas las actividades que aparecen en el diagrama de Gantt.
Precondición	Haber accedido a la pantalla del diagrama de Gantt y a la sección de actividades del diagrama.
Acciones	<ol style="list-style-type: none"> 1. Deslizar la lista de actividades para ver todas las actividades del diagrama.
Postcondición	El usuario visualiza las actividades
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.10: CU-5 Visualización de actividades en la sección del diagrama de Gantt

CU-5.1	Visualizar tarjeta de la actividad
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-6, RF-6.1
Descripción	Permitir visualizar toda la información disponible de la actividad seleccionada.
Precondición	Haber accedido a la pantalla del diagrama de Gantt y a la sección de actividades.
Acciones	<ol style="list-style-type: none"> 1. Presionar sobre la tarjeta de la actividad que se quiera visualizar.
Postcondición	El usuario visualiza la información de la actividad.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Baja

Tabla B.11: CU-5.1 Visualizar tarjeta de la actividad

CU-6	Visualización de tareas personales
Versión	1.0
Actor	Usuario, Supabase
Autor	Javier Pampliega García
Requisitos asociados	RF-7
Descripción	Permitir visualizar todas las tareas personales del usuario.
Precondición	Haber accedido desde la pantalla principal.
Acciones	<ol style="list-style-type: none"> 1. Visualizar la sección de tareas <i>Pendientes</i> y todas sus subsecciones. 2. Visualizar la sección de tareas. <i>Completadas</i> y todas sus subsecciones.
Postcondición	El usuario visualiza todas las tareas personales.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Alta

Tabla B.12: CU-6 Visualización de tareas personales

CU-6.1	Visualizar información de la tarea personal
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-7, RF-7.1
Descripción	Permitir visualizar la información asociada a la tarea que se desea revisar.
Precondición	Haber accedido a la pantalla del diagrama de Gantt.
Acciones	<ol style="list-style-type: none"> 1. Presionar la tarea personal que se desea visualizar.
Postcondición	El usuario visualiza la información de la tarea personal.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.13: CU-6.1 Visualizar información de la tarea personal

CU-6.2	Cambiar estado de finalización de la tarea personal
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-7, RF-7.2
Descripción	Permitir cambiar el estado de finalización de una tarea personal.
Precondición	Haber accedido a la pantalla de tareas personales.
Acciones	<ol style="list-style-type: none"> 1. Presionar la casilla circular para cambiar el estado de finalización de la tarea.
Postcondición	El usuario cambia el estado de finalización y la tarea cambia de sección, <i>Pendiente</i> o <i>Completada</i> .
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.14: CU-6.2 Cambiar estado de finalización de la tarea personal

CU-6.3	Borrado de tarea personal
Versión	1.0
Actor	Usuario, Supabase
Autor	Javier Pampliega García
Requisitos asociados	RF-7, RF-7.3
Descripción	Permitir eliminar tareas personales.
Precondición	Haber accedido a la pantalla de tareas personales.
Acciones	<ol style="list-style-type: none"> 1. Deslizar hacia la izquierda la tarea personal. 2. Presionar el icono que representa una "papelera".
Postcondición	El usuario elimina la tarea personal.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.15: CU-6.3 Visualizar información de la tarea personal

CU-7	Creación de tareas personales
Versión	1.0
Actor	Usuario, Supabase
Autor	Javier Pampliega García
Requisitos asociados	RF-7, RF-8
Descripción	Permitir al usuario la creación de nuevas tareas personales.
Precondición	Haber accedido a la pantalla de tareas personales.
Acciones	<ol style="list-style-type: none"> 1. Presionar el botón de la esquina inferior derecha. 2. Rellenar los campos marcados como obligatorios (*). 3. Presionar el botón de guardar.
Postcondición	El usuario crea una tarea personal.
Excepciones	<ul style="list-style-type: none"> ■ No se rellenan todos los campos obligatorios.
Importancia	Alta

Tabla B.16: CU-7 Creación de tareas personales

CU-8	Filtrado del contenido de la aplicación
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-9
Descripción	Permitir al usuario filtrar todos los contenidos con filtros de <i>Cursos</i> , <i>Actividades</i> , <i>Fechas</i> y <i>Disponibilidad de fechas</i> .
Precondición	Haber iniciado sesión correctamente y estar en la pantalla principal.
Acciones	<ol style="list-style-type: none"> 1. Presionar el botón de la parte inferior derecha de la pantalla principal. 2. Modificar los filtros según las necesidades.
Postcondición	El usuario filtra todos los contenidos de la aplicación.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Alta

Tabla B.17: CU-8 Filtrado del contenido de la aplicación

CU-9	Cierre de sesión
Versión	1.0
Actor	Usuario
Autor	Javier Pampliega García
Requisitos asociados	RF-3
Descripción	Cerrar sesión de la aplicación.
Precondición	Estar situado en la pantalla principal.
Acciones	<ol style="list-style-type: none"> 1. Presionar el botón situado en la esquina superior derecha.
Postcondición	El usuario cierra sesión correctamente en la aplicación.
Excepciones	<ul style="list-style-type: none"> ■ No existen excepciones.
Importancia	Media

Tabla B.18: CU-9 Cierre de sesión

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice, se recoge la especificación de diseño del proyecto. El objetivo es detallar los aspectos de construcción del sistema, en el que se incluyen:

- **Diseño de datos:** se definen entidades con sus atributos y relaciones entre ellas. También, se definen esquemas de persistencia.
- **Diseño arquitectónico:** se define la estructura interna del sistema, así como la comunicación entre los distintos módulos.
- **Diseño procedimental:** se definen los flujos de interacción entre el cliente y el servicio.

C.2. Diseño de datos

En esta sección, se explican los diseños de datos del sistema. A continuación, se explican las entidades que componen el sistema y las relaciones de cada una de ellas.

Diccionario de datos

En este proyecto solo se utiliza una tabla en la base de datos, ya que solo se desea almacenar las tareas personales de un usuario, por lo tanto,

solo se requiere la utilización de una tabla. En caso de que existieran más datos a almacenar en la nube, la cantidad de tablas aumentaría.

PERSONAL_TASKS			
		PK	autoincrement
INTEGER	id		
INTEGER	user_id		NULL
TEXT	moodle_id		NULL
TEXT	name		NULL
TEXT	description		NULL
TEXT	course		NULL
TIMESTAMP	date		NULL
BOOLEAN	done		NULL
TIMESTAMP	finishedat		NULL
TEXT	priority		NULL

Figura C.1: Diccionario de datos

Se explican los atributos del diccionario de datos anterior:

- **id:** representa el número de identificación de la tarea, es autoincremental Fig. C.1.
- **user_id:** representa el número de identificación del usuario dentro de la plataforma en la que se encuentre Fig. C.1.
- **moodle_id:** representa la URL de la plataforma a la que pertenece el usuario Fig. C.1.
- **name:** nombre de la tarea personal Fig. C.1.
- **description:** descripción de la tarea personal Fig. C.1.
- **course:** curso al que pertenece la tarea personal (entre los cursos en los que se encuentra matriculado el usuario propietario de la tarea personal) Fig. C.1.

- **date:** fecha de finalización de la tarea personal Fig. C.1.
- **done:** estado de finalización de la tarea Fig. C.1.
- **finishedat:** fecha en la que se finaliza la tarea Fig. C.1.
- **priority:** prioridad de la tarea Fig. C.1.

Modelos de datos

A continuación, se muestra un diagrama de clases para comprender la relación entre los distintos modelos de datos de la aplicación y la función de cada uno de ellos.

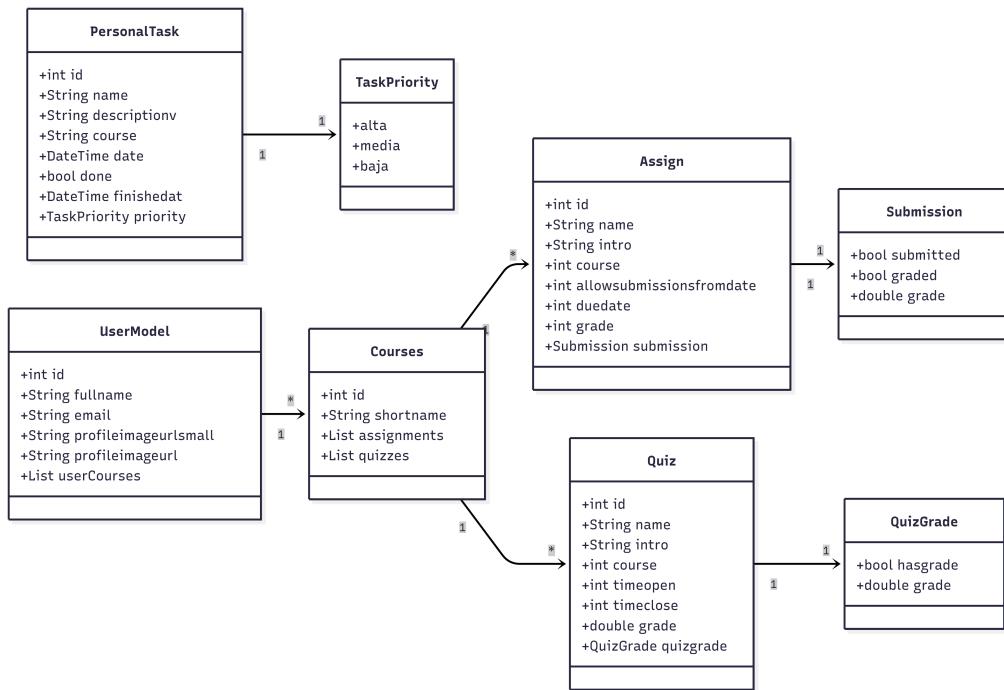


Figura C.2: Diagrama de Modelo de datos

En los modelos hay definidos más atributos, pero en el diagrama C.2 se representan aquellos que han sido empleados en el proyecto. A continuación se detalla cada uno de ellos:

Atributo	Tipo	Contenido
id	int	Número de identificación del usuario.
fullname	String	Nombre completo del usuario.
email	String	Correo electrónico del usuario.
profileimageurlsmall	String	Dirección de imagen de perfil del usuario.
profileimageurl	String	Dirección de imagen de perfil del usuario.
userCourses	Lista	Lista de cursos del usuario.

Tabla C.1: Modelo UserModel

Atributo	Tipo	Contenido
id	int	Número de identificación del curso.
shortname	String	Nombre corto del curso.
assingments	Lista(Assign)	Lista de tareas del curso.
quizzes	Lista(Quiz)	Lista de cuestionarios del curso.

Tabla C.2: Modelo Courses

Atributo	Tipo	Contenido
id	int	Número de identificación de la tarea.
name	String	Nombre de la tarea.
intro	String	Descripción de la tarea.
course	int	Número de identificación del curso de la tarea.
allowsubmissionsfromdate	int	Fecha de apertura de la tarea.
duedate	int	Fecha de cierre de la tarea.
grade	int	Tipo de calificación.
submission	Submission	Entrega de la tarea.

Tabla C.3: Modelo Assign

Atributo	Tipo	Contenido
id	int	Número de identificación del cuestionario.
name	String	Nombre del cuestionario.
intro	String	Descripción del cuestionario.
course	int	Número de identificación del curso del cuestionario.
timeopen	int	Fecha de apertura del cuestionario.
timeclose	int	Fecha de cierre del cuestionario.
grade	double	Tipo de calificación.
quizgrade	QuizGrade	Entrega del cuestionario.

Tabla C.4: Modelo Quiz

Atributo	Tipo	Contenido
submitted	bool	Estado de entrega de la tarea.
graded	bool	Estado de calificación de la entrega.
grade	double	Calificación de la entrega.

Tabla C.5: Modelo Submission

Atributo	Tipo	Contenido
hasgrade	bool	Estado de calificación del cuestionario.
grade	double	Calificación del cuestionario.

Tabla C.6: Modelo QuizGrade

Atributo	Tipo	Contenido
id	int	Número de identificación de la tarea personal.
name	String	Nombre de la tarea personal.
description	String	Descripción de la tarea personal.
course	String	Nombre del curso de la tarea personal.
date	DateTime	Fecha de cierre de la tarea personal.
done	bool	Estado de finalización de la tarea personal.
finishedat	DateTime	Fecha de finalización de la tarea personal.
priority	TaskPriority	Tipo de prioridad de la tarea personal.

Tabla C.7: Modelo PersonalTask

Atributo	Contenido
alta	Prioridad alta.
baja	Prioridad media.
media	Prioridad baja.

Tabla C.8: Modelo TaskPriority

C.3. Diseño arquitectónico

El diseño arquitectónico empleado en PlanLMS se basa en una estructura modular, que persigue lograr una organización clara de las funcionalidades y responsabilidades, con el fin de garantizar la facilidad en el mantenimiento y escalabilidad del proyecto.

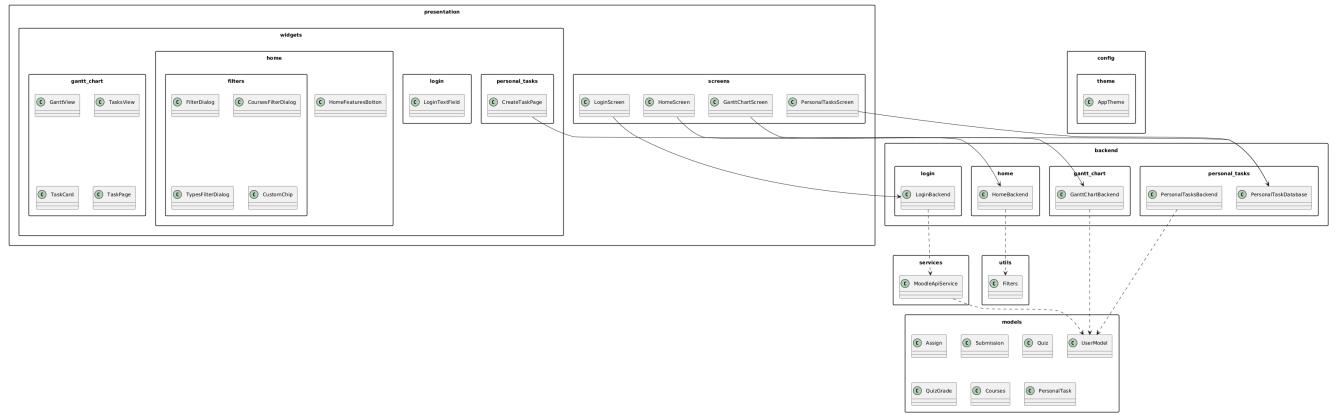


Figura C.3: Diseño arquitectónico del proyecto

A continuación, se detallan los objetivos de cada responsabilidad y funcionalidad:

- **presentation:** contiene todos los elementos de interfaz gráfica Fig. C.3. Este paquete se divide en dos paquetes:
 - **screens:** contiene las pantallas de la aplicación.
 - **widgets:** contiene algunos widgets que componen las pantallas, con el fin de aprovechar la reutilización de estos.
 Estos dos paquetes se dividen en otros más pequeños, cada uno asociado a una funcionalidad de la aplicación.
- **config:** contiene elementos de configuración de temas estéticos de algun componente Fig. C.3.
- **backend:** contiene la lógica de negocio, se encarga de gestionar los datos de cada capa de presentación Fig. C.3.
- **services:** se encarga de la comunicación con lo servicios externos Fig. C.3.
- **utils:** contiene los elementos que son empleados por las diferentes clases de la arquitectura Fig. C.3.
- **models:** contiene todas las entidades y define las estructuras de estas Fig. C.3.

Estrategias de persistencia

La persistencia de los datos dentro del sistema, se reparte en tres formas:

- **SharedPreferences:** se encarga de almacenar los filtros de los usuarios para poder restaurarlos cada vez que se inicia la aplicación. Además, también guarda el correo electrónico de inicio de sesión.
- **Supabase:** se encarga de almacenar las tareas personales de todos los usuarios. Cada vez que se consulta Supabase se retorna la lista más reciente de tareas del usuario.
- **Caché en memoria:** se encarga de almacenar los datos de la sesión del usuario. Algunos de estos datos son los cursos, tareas, cuestionarios, entregas, calificaciones, etc.

Seguridad

En el directorio del proyecto existe un archivo denominado `.env`, que alberga las claves para acceder a la base de datos de Supabase. Estos ficheros se excluyen de la sincronización del repositorio en la nube. Evitando que cualquier usuario no deseado pudiera acceder a los datos.

Por otra parte, el *token* privado que proporciona el *Web Service de Moodle* se mantiene en memoria, además, de que al cerrar sesión este se elimina de memoria.

Por último, todas las peticiones HTTP van por HTTPS, lo que quiere decir, que todas las peticiones emplean un protocolo de comunicación seguro donde los datos no van a ser leídos por terceros.

C.4. Diseño procedimental

A continuación, se muestran diagramas de secuencia que representan el funcionamiento interno de la aplicación.

Inicio de sesión

La Figura C.4 describe el comportamiento del sistema cuando el usuario inicia sesión en PlanLMS.

1. El usuario introduce el correo electrónico y contraseña y presiona el botón "Iniciar sesión".

2. **LoginScreen** le pide a su *backend* que ejecute **loginSetup(email, pass, saveEmail)**, que a su vez, solicita a **MoodleAPIService** que ejecute **login(username, password)**.
3. Una vez obtenido el *token*, se almacena opcionalmente el email con **SharedPreferences**, y solicita los datos del usuario y sus cursos a **MoodleAPIService**, mediante **getUserInfo()** y **getUserCourses(userId)**.
4. Realiza iteraciones para cada curso, obteniendo las tareas (**getCourseAssignments()**) y cuestionarios (**getCourseQuizzes()**) de cada uno, así como los estados de entrega de cada una de las actividades (**getAssignSubmissionStatus()** y **getQuizSubmissionStatus()**).
5. Toda la información extraída, se convierte en objetos que se asignan al **UserModel** y a **Course**, posteriormente se retorna a **LoginScreen** y se navega al **HomeScreen**.

Creación de tareas personales

La Figura C.5 describe el comportamiento del sistema cuando el usuario procede con la creación de una tarea personal.

1. El usuario abre el formulario y lo rellena. Al pulsar el botón *Guardar* se realiza una llamada a **validateFields()** que se encarga de verificar que todos los campos del formulario han sido rellenados.
2. Si la verificación es correcta, se ejecuta **createPersonalTask(PersonalTask)** de **PersonalTaskDatabase**, para realizar la inserción de la nueva tarea en la base de datos en la nube.
3. Cuando se confirma la inserción, se realiza una llamada a **widget.refreshTasks()** que hace referencia a **loadTasks()** de **PersonalTasksScreen**, se actualiza la lista de tareas y se muestra la nueva tarea.
4. En caso de que no se rellene el formulario con los datos necesarios, se muestra un **SnackBar** indicando *Rellene los campos*.

Eliminación de tareas personales

La Figura C.6 describe el comportamiento del sistema cuando el usuario procede con la eliminación de una tarea personal.

1. El usuario desliza la tarjeta de la tarea y pulsa en el ícono de *Eliminar*.
2. **TaskCard** llama a la función `deleteTask(task)` de **PersonalTaskDatabase**, que solicita la base de datos eliminar la tarea personal de la tabla *personal_tasks*.
3. Tras confirmar que se ha eliminado correctamente, se retorna de nuevo a **TaskCard**, que lanza la orden de ejecutar `refreshTasks()`, que repite el mismo proceso que la Figura C.5.

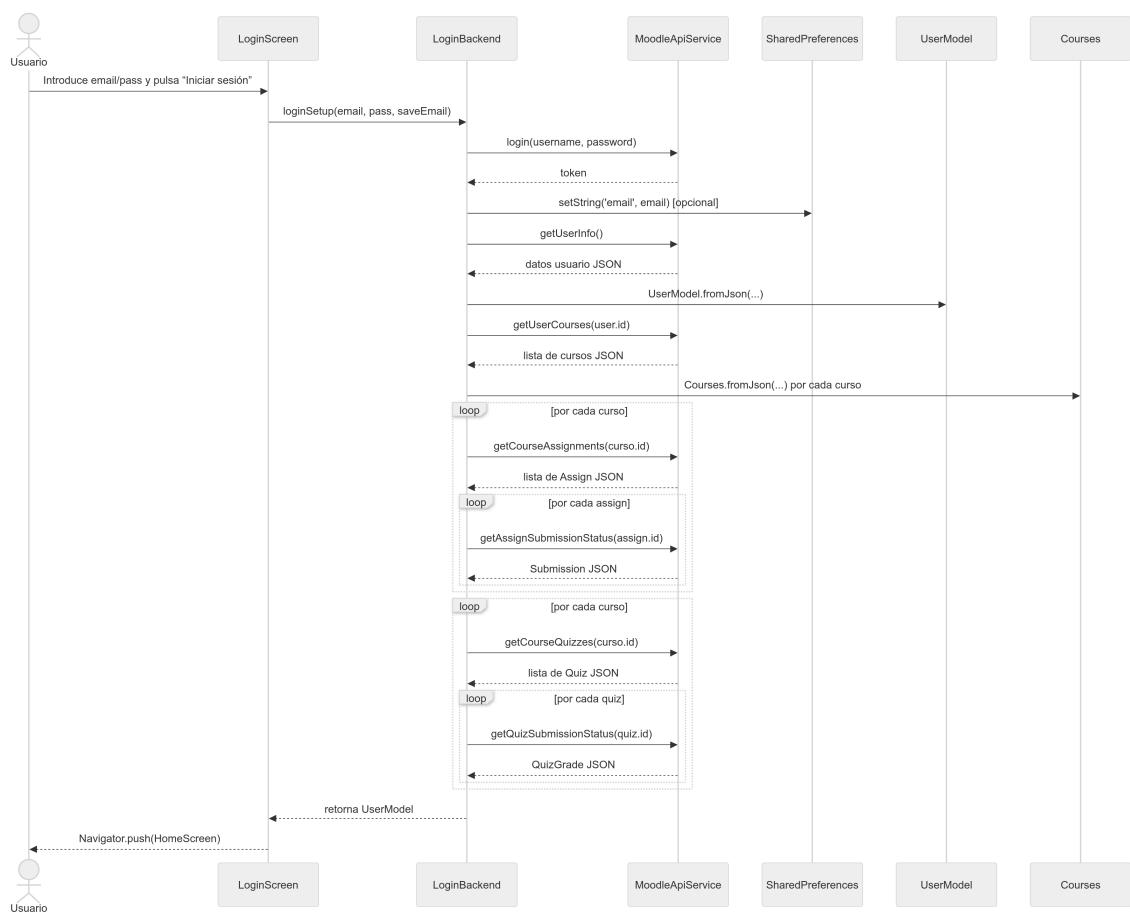


Figura C.4: Diagrama de secuencia del inicio de sesión

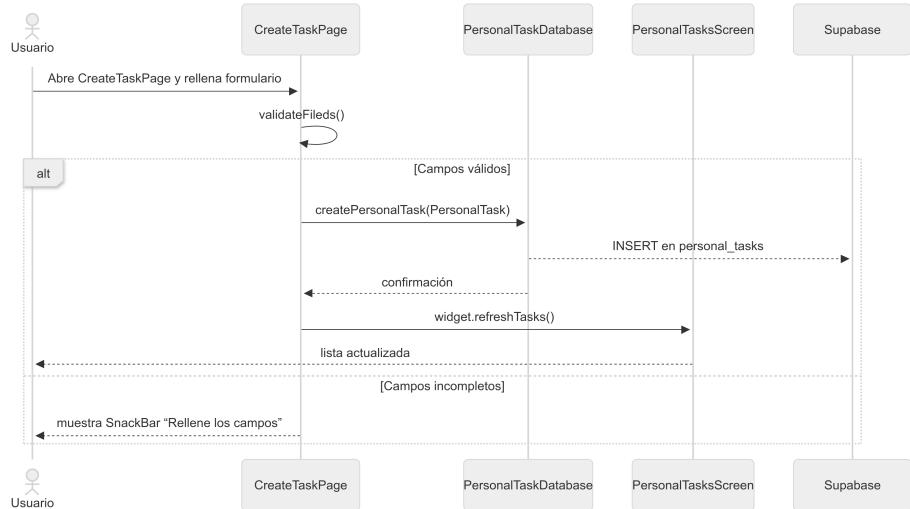


Figura C.5: Diagrama de secuencia de la creación de tareas

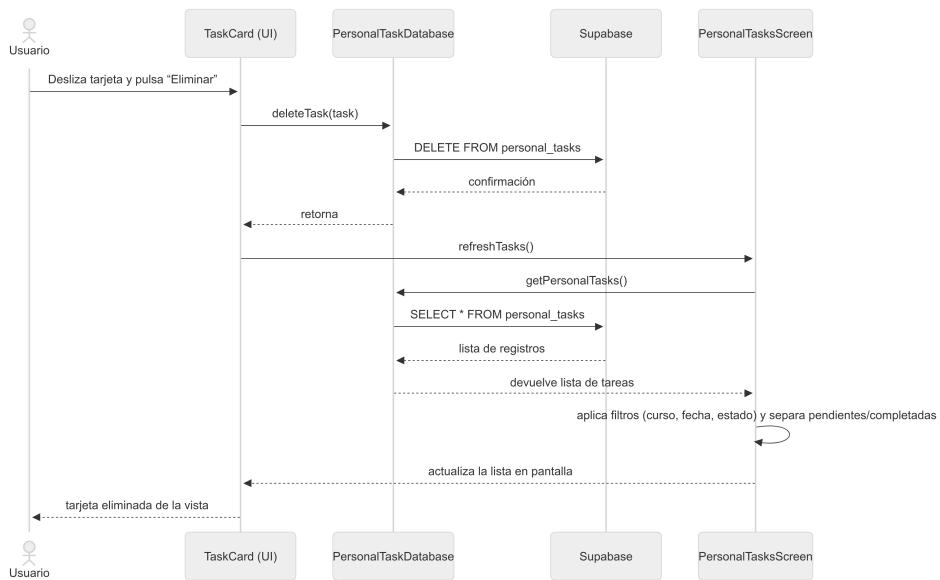


Figura C.6: Diagrama de secuencia de la eliminación de tareas

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este apéndice se centra en proporcionar una visión técnica detallada del desarrollo del proyecto. El objetivo es poder proporcionar una referencia a los futuros desarrolladores que requieran de la comprensión de la estructura del código y los procesos de compilación e instalación.

En las siguientes secciones se describen la organización de directorios del proyecto, un manual del programador con instrucciones para acceder al proyecto y sus respectivas configuraciones, y los procesos de configuración, instalación y ejecución del proyecto.

D.2. Estructura de directorios

A continuación se explican todos los directorios del **repositorio** del proyecto [3]:

- **app/app:** contiene todo el proyecto de Flutter. Se divide en las siguientes subcarpetas:
 - **/android:** directorio generado por Flutter que contiene todos los archivos y carpetas necesarios para ejecutar la aplicación en dispositivos *Android*.

- **/assets:** carpeta que contiene todo el contenido audiovisual de la aplicación, en este caso imágenes.
- **/ios:** directorio que contiene configuraciones, ejecutables, etc, que permiten ejecutar la aplicación en dispositivos *iOS*.
- **/lib:** directorio que contiene todo el código fuente de la aplicación. Dentro de este directorio hay distintos subdirectorios, cada uno de ellos con una función:
 - **/backend:** contiene el backend de cada una de las funcionalidades de la aplicación.
 - **/config:** contiene ficheros de configuración de la aplicación.
 - **/models:** contiene ficheros que representan cada uno de los modelos de la aplicación.
 - **/presentation:** contiene todas las capas de presentación de *UI*, dentro de esta carpeta se distinguen:
 - ◊ **/screens:** representación de las pantallas de la aplicación, donde cada funcionalidad tiene su propio directorio.
 - ◊ **/widgets:** se dividen en directorios por cada funcionalidad, en cada directorio hay *widgets* para mejorar la reutilización.
 - **/services:** contiene el fichero que se encarga de comunicarse con la API de Moodle.
 - **/utils:** contiene un fichero con elementos comunes en toda la aplicación.
- **/linux:** directorio que contiene todo lo relacionado a configuraciones de *Linux* para por ejecutar la aplicación en este entorno.
- **/macos:** directorio que contiene todo lo relacionado al sistema operativo de *MacOS* para por ejecutar la aplicación en este entorno.
- **/windows:** este directorio contiene todos los archivos que permiten ejecutar la aplicación en *Windows*.
- **docs:** contiene todos los documentos L^AT_EX

D.3. Manual del programador

En esta sección se detallan todos los pasos a seguir para instalar y ejecutar localmente el proyecto correctamente.

Configuración del entorno

A continuación se explican todos los pasos a seguir para instalar el proyecto desde el repositorio de *GitHub*.

Antes de instalar el proyecto en el equipo, es necesario tener instalado el editor de código Visual Studio Code, así como Git, posteriormente se indicarán las extensiones recomendables para el uso de *Dart* y *Flutter*.

Por otra parte, es necesario tener instalado en el equipo tanto *Dart* como *Flutter*. En este caso, no es necesario hacer una instalación directa de *Dart*, ya que viene incluido en la instalación de *Flutter*.

El primer paso es acceder a la [página de instalación de Flutter](#) y descargar el instalador del sistema que se esté utilizando. Se recomienda crear una carpeta para almacenar el *SDK*. Se debe de extraer el *SDK* del *zip* en la carpeta creada. Por último, se crea una variable de entorno en el *PATH* haciendo referencia a la ruta de la carpeta: <ruta_carpeta>\flutter\bin [1].

Para comprobar si la instalación se ha realizado con éxito, introduce el siguiente comando:

```
flutter doctor
```

Este comando realiza un diagnóstico de la instalación y configuración del entorno. En caso de que falte algún elemento, se mostrará por pantalla y se deberá de proceder a su instalación.

Por otra parte, en Visual Studio Code se deben de instalar las extensiones *Dart* y *Flutter*, que permiten hacer uso de *Dart* y *Flutter*, además de facilitar el desarrollo al incluir atajos.

Instalación del proyecto

A continuación se procederá a la instalación del proyecto. Para ello, se debe de realizar una clonación del repositorio del proyecto. Para ello, se utiliza una terminal de comandos y se introducirá el siguiente comando:

```
git clone https://github.com/jpg1011/TFG-GestorTareasMoodle.git
```

Posteriormente a la instalación del proyecto, se accederá a la carpeta de proyecto mediante el siguiente comando:

```
cd TFG-GestorTareasMoodle/app/app
```

Por último, ejecutar el siguiente comando, que abrirá el editor de código:

```
code .
```

D.4. Compilación y ejecución del proyecto

En esta sección se describen los pasos a seguir para compilar y ejecutar el proyecto. Se incluye una configuración del emulador para simular la aplicación en entornos Android.

Configuración de simulador

Para poder disponer de un emulador Android, primero es necesario instalar Android Studio, desde el cual se creará el emulador.

Una vez instalado Android Studio, se accede a la sección de emuladores desde el siguiente botón:



Figura D.1: Botón de acceso a la configuración de emuladores

Posteriormente, aparecerá la sección de emuladores y se debe de presionar el siguiente botón para añadir un nuevo emulador.

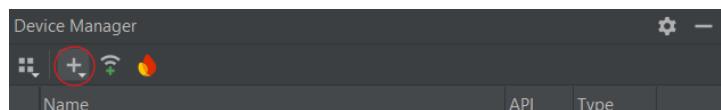


Figura D.2: Botón para añadir nuevos emuladores

Una vez presionado el botón (imagen D.2), se selecciona la opción *Create Virtual Device*.

A continuación, aparecerá una ventana con todas las imágenes de sistema disponibles para instalar en el emulador. Se seleccionará la siguiente imagen:

Release Name	API	xABI	Target
API 35	35	x86_64	Android API 35 (Default Android System Image)
API 35	35	x86_64	Android API 35 (AOSP ATD)
UpsideDownCake	34	x86_64	Android 14.0 (Google APIs ATD)
UpsideDownCake (Extension Level 12)	34	x86_64	Android 14.0 (Google Play)
UpsideDownCake (Extension Level 8)	34	x86_64	Android 14.0 (Google Play)
UpsideDownCake (Extension Level 10)	34	x86_64	Android 14.0 (Google Play)
UpsideDownCake (Extension Level 11)	34	x86_64	Android 14.0 (Google Play)
UpsideDownCake	34	x86_64	Android 14.0 (Google APIs)
UpsideDownCake	34	x86_64	Android 14.0 (Default Android System Image)
UpsideDownCake	34	x86_64	Android 14.0 (AOSP ATD)

Figura D.3: Ventana con imágenes del sistema

Por último, aparecerá una pantalla de configuraciones generales del emulador. En este caso, se dejará todo de la misma forma.

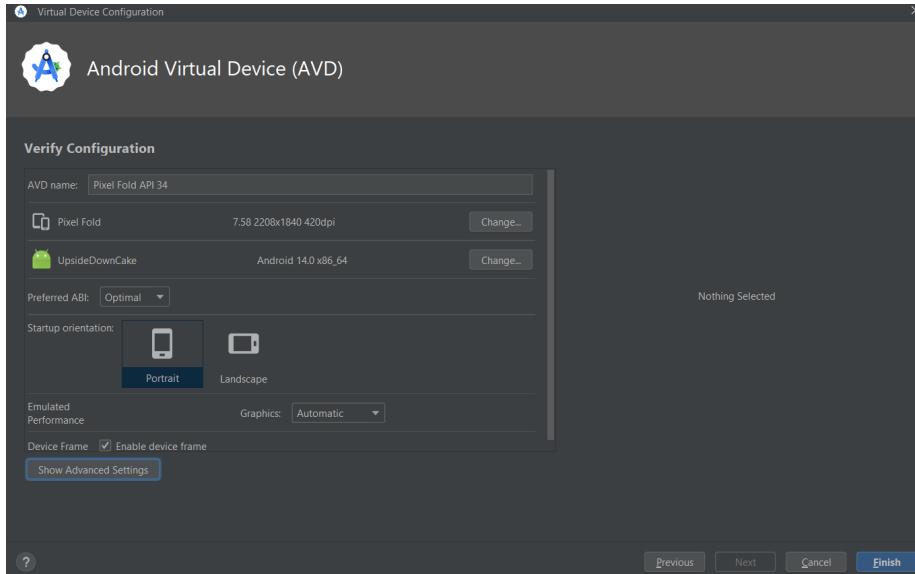


Figura D.4: Ventana de configuración general del emulador

Instalación de dependencias

Para poder ejecutar la aplicación es necesario tener instaladas las dependencias requeridas por la aplicación.

Para ello, será necesario ejecutar por terminal el siguiente comando:

```
flutter pub get
```

Tras la instalación de las dependencias se puede proceder con la ejecución de la aplicación.

Ejecución de la aplicación

A continuación se detallan los pasos necesarios para poder ejecutar la aplicación en el emulador instalador previamente. El primer paso es pulsar la combinación de teclas **Ctrl + Shift + P**. Se mostrará una ventana con un campo de texto en el que hay que introducir: *Flutter: Launch Emulator*.

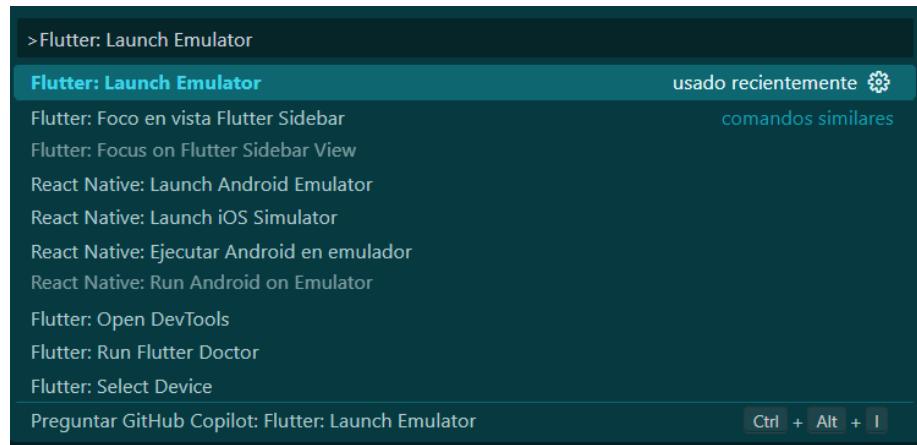


Figura D.5: Ventana de opciones de usuario

Seleccionando la opción anterior, se desplegará una lista con los emuladores disponibles, donde debería de aparecer el emulador creado anteriormente.

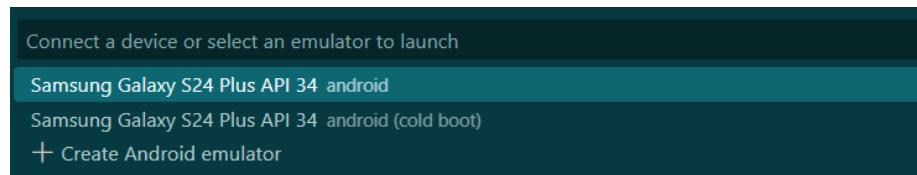


Figura D.6: Lista con todos los emuladores disponibles

Se iniciará el emulador, y el último paso es lanzar la aplicación con el siguiente comando.

```
flutter run
```

Apéndice E

Documentación de usuario

E.1. Introducción

Este apéndice tiene como objetivo brindar a los usuarios los pasos necesarios para instalar y utilizar *PlanLMS*. Se incluye a continuación información detallada sobre los requisitos mínimos para ejecutar la aplicación, pasos que el usuario debe seguir para instalar la aplicación en su dispositivo y una guía de uso.

E.2. Requisitos de usuarios

En esta sección se describen los requisitos mínimos que debe de cumplir el dispositivo del usuario para poder ejecutar correctamente *PlanLMS*.

Requisitos del Sistema

- **Versión mínima Android:** 5.0 (API 21, *Lollipop*)
- **Versión objetivo Android:** 14.0 (API 34, *Upside Down Cake*)

Requisitos Hardware

La aplicación requiere de un dispositivo con acceso a Internet para la obtención de los datos del usuario de Moodle, así como la gestión de tareas personales, es decir, la comunicación con la base de datos.

E.3. Instalación

Actualmente la aplicación dispone de los archivos de instalación para dispositivos Android, estos se encuentran en la sección *Releases* en el repositorio de GitHub.

1. Acceder al siguiente repositorio de GitHub: <https://github.com/jpg1011/TFG-GestorTareasMoodle>.
2. Acceder a la sección de *Releases*, que se encuentra en la parte derecha.
3. Localizar la versión más reciente y acceder a *Assets*.
4. Descargar el archivo con extensión .apk.
5. Una vez descargado el archivo en el dispositivo, abrir el archivo para proceder con la instalación.
Es necesario tener activada la opción *Permitir aplicaciones de fuentes desconocidas*.
6. Seguir las instrucciones del dispositivo para finalizar la instalación.

E.4. Manual del usuario

A continuación, se describen todas las acciones que puede realizar un usuario de PlanLMS. Se detallarán todas las funcionalidades, así como todos los pasos a seguir para poder usar de forma eficiente cada una de las funcionalidades.

Inicio de sesión



Figura E.1: Inicio de sesión de PlanLMS

En la figura E.1 aparece el inicio de sesión que se compone de dos campos para introducir las credenciales de Moodle del usuario. Debajo hay un texto que redirige a la página de recuperación de contraseñas, en caso de no recordar la contraseña. Un *checkbox* que permite recordar el correo del usuario y el botón de acceso a la aplicación.

Por otra parte, esta el texto *Conectarse a Moodle*. Al presionar dicho texto, se mostrará un pequeño diálogo, en el que se podrá introducir la *URL* del servidor a conectar. Además, se incluye un sistema que verifica si la *URL* introducida se trata de una plataforma Moodle.



Figura E.2: Diálogo para conectarse a un Moodle

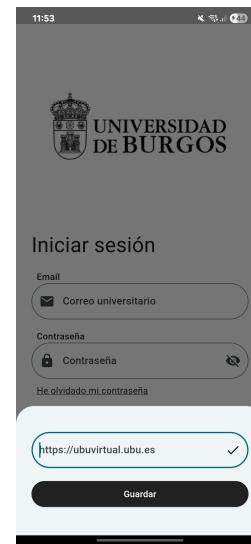


Figura E.3: Ejemplo de conexión

Al presionar el botón *Iniciar sesión*, se empezarán a obtener los siguientes datos de Moodle del usuario:

- Información del usuario
- Cursos del usuario
- Tareas y cuestionarios de los cursos del usuario
- Entregas y calificaciones de las tareas y cuestionarios

La duración del proceso de inicio de sesión puede variar en función de la cantidad de información que se vaya a extraer.

Pantalla principal

La primera pantalla que el usuario visualizará tras iniciar sesión, será la pantalla principal donde el usuario tendrá acceso a la siguientes funcionalidades de *PlanLMS*:

- Diagrama de Gantt
- Tareas personales

- Sistema de filtrado

Además, en la parte superior derecha hay un botón que se trata del cierre de sesión.

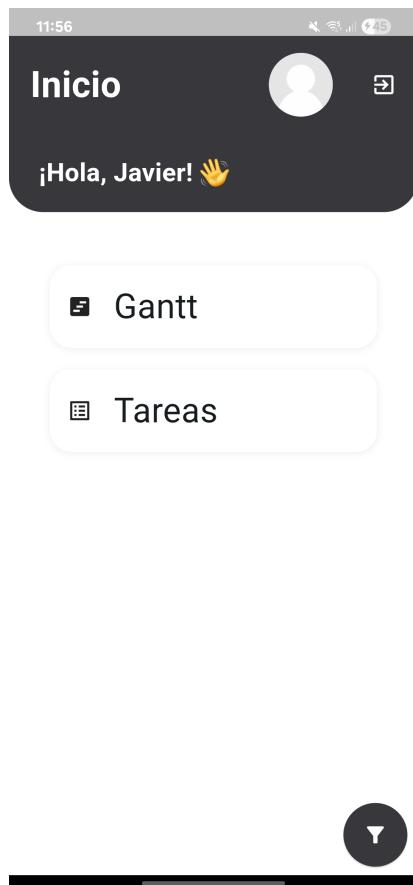


Figura E.4: Pantalla principal de PlanLMS

Sistema de filtrado

Para acceder a esta función basta con pulsar el botón situado en la esquina inferior derecha de la figura E.4. Al ser pulsado, aparecerá un diálogo en medio de la pantalla con todos los filtros disponibles. Algunos filtros solo afectan a algunas funcionalidades de la aplicación:

- **Filtrado de cursos:** Diagrama de Gantt y Tareas Personales.
- **Filtrado de actividades:** Diagrama de Gantt.

- **Filtrado por fechas:** Diagrama de Gantt y Tareas Personales.
- **Filtrado por fechas disponibles:** Diagrama de Gantt.

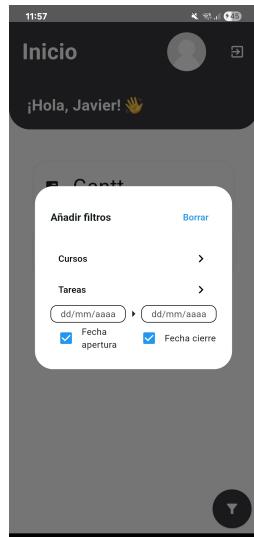


Figura E.5: Diálogo de filtros

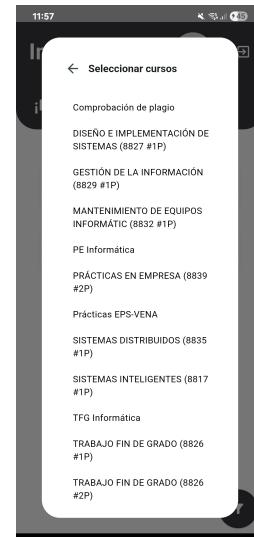


Figura E.6: Filtrado de cursos

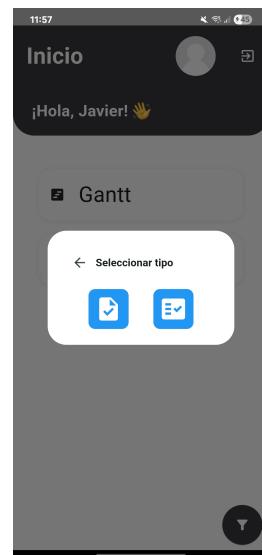


Figura E.7: Filtrado de actividades

Funcionamiento de los filtros

A continuación se detalla el funcionamiento de cada uno de los filtros y como estos afectan a la aplicación.

Filtrado de cursos

Al acceder al diálogo E.6, se muestra una lista con todos los cursos del usuario. Por defecto, se filtrarían todos los cursos (aunque aparezcan como deseleccionados). Si se pulsa en el nombre de cada curso, este se marcaría y se filtrarían solo aquellos cursos que estén seleccionados.

Filtrado de actividades

Al acceder al diálogo E.7, se muestran dos iconos que por defecto se encuentran activados, por lo tanto, se filtrarían ambos tipos de actividades. En el caso de querer descartar alguna, basta con desmarcar la opción.

Filtrado de fechas

En el diálogo de filtros E.5, el tercer filtro disponible permite filtrar las fechas de tres formas diferentes:

- Actividades a partir de una fecha
- Actividades anteriores a una fecha
- Actividades entre dos fechas

Filtrado de fechas disponibles

El último filtro permite filtrar aquellas actividades que por configuración carecen de fecha de apertura o de cierre, o incluso ambas. Consiste en dos casillas que en función de si están marcadas o no, mostrarán actividades con la disponibilidad de esas fechas.

Diagrama de Gantt

Una de las pantallas a las que se puede acceder desde la *Pantalla Principal* E.4 es el diagrama de Gantt. Esta pantalla permite al usuario visualizar las actividades filtradas en un diagrama de Gantt, que además puede ser personalizado con filtros para modificar su apariencia.

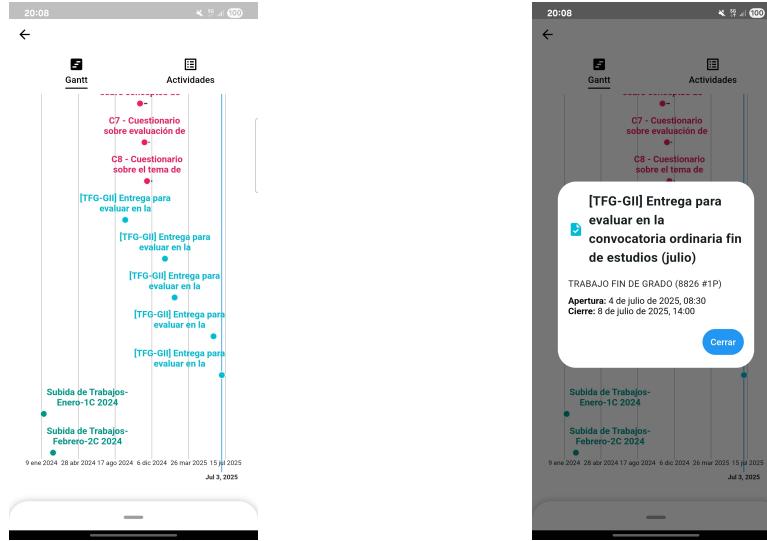


Figura E.8: Pantalla de diagrama de Gantt

Figura E.9: Diálogo de actividad en diagrama de Gantt

Si se presiona cualquier punto de los que aparecen en el diagrama, se abrirá un pequeño diálogo con información de la actividad E.9.

En la parte inferior de la pantalla E.8 se puede observar un pequeño panel que si es deslizado hacia arriba, mostrará dos filtros que permiten modificar la estética del diagrama.

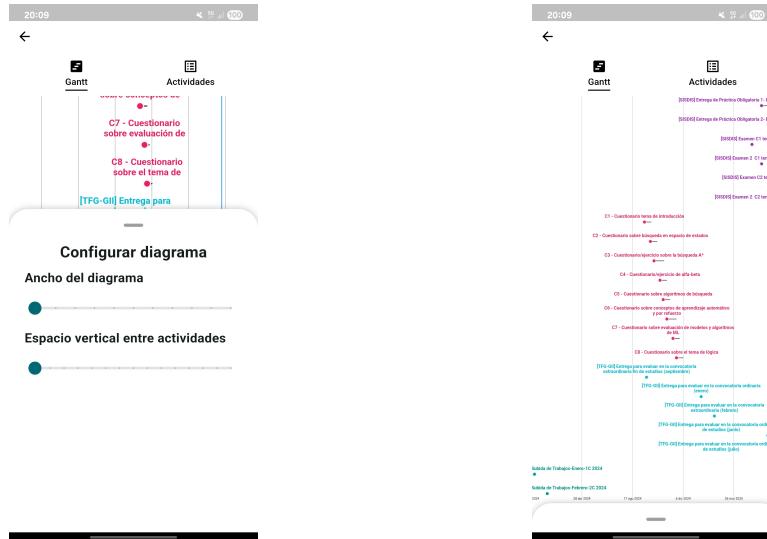
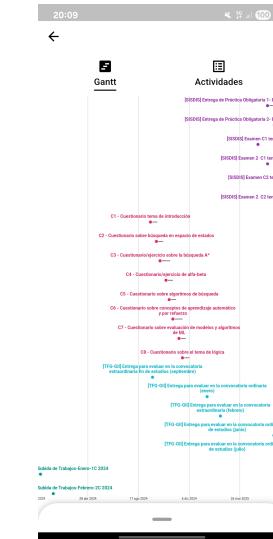


Figura E.10: Filtros del diagrama de Gantt

Figura E.11: Filtros estéticos aplicados sobre el diagrama



Otra funcionalidad del diagrama, es que permite el desplazamiento en todas las direcciones, así como hacer *zoom*, todo mediante gestos en la pantalla con los dedos.

Por otra parte, dentro de la pantalla del diagrama de Gantt, hay una sección de actividades E.12, en la cual aparecen todas las actividades que se encuentren dibujadas en el diagrama de Gantt.

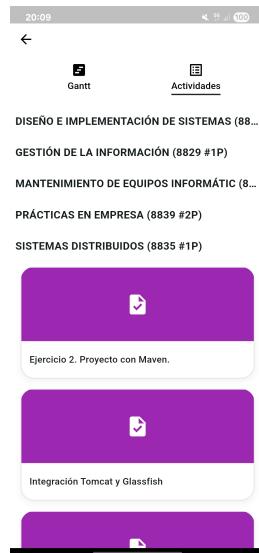
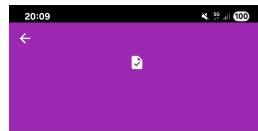


Figura E.12: Sección de actividades del diagrama de Gantt

Al presionar una de las tarjetas que aparecen, se despliega una nueva ventana E.13 con toda la información que se recoge de Moodle y que pueda llegar a ser útil para el usuario.



Ejercicio 2. Proyecto con Maven.
Subir el fichero "POM.xml" obtenido en el Ejercicio 2.
Proyecto con Maven

Fecha de apertura
10 de febrero de 2025, a las 00:00

Fecha de cierre
17 de febrero de 2025, a las 00:00

Estado de entrega
No entregado

Calificación
Sin calificar/10

Figura E.13: Ventana con información de la actividad

Tareas Personales

La última funcionalidad a explicar, son la tareas personales. Esta función tiene acceso desde la pantalla principal [E.4](#) y permite al usuario crear sus propias tareas personales con información personalizada.

Dentro de esta pantalla existen dos secciones: *Pendientes* y *Completadas*. Como su propio nombre indica, hacen referencia al estado de finalización de la tarea. A su vez, dentro de cada sección hay distintas subsecciones:

- **Pendientes:** Hoy, Mañana, Esta Semana y Todos
- **Completadas:** Hoy, Últimos 7d, Este mes y Todos



Figura E.14: Sección de tareas pendientes



Figura E.15: Sección de tareas completadas

En la parte inferior derecha de la pantalla E.14 se encuentra un botón que al ser pulsado abre una ventana con un formulario de *Creación de tareas*. El cual consta de unos campos obligatorios y otros opcionales. Una vez llenados los campos y guardada la tarea, esta se añade directamente a la sección de tareas pendientes.

Figura E.16: Formulario de creación de tareas personales

Figura E.17: Tarea personal en sección de *Pendientes*

Al presionar la tarea personal creada, aparecerá un panel con toda la información de dicha tarea [E.18](#).



Figura E.18: Panel con información de la tarea personal

El círculo que aparece en la tarea de la pantalla [E.17](#), al ser presionado marca la tarea como *Completada* y se moverá directamente a la sección correspondiente.

Por último, si se desliza la tarea de derecha a izquierda, se descubrirá la opción de borrado de la tarea [E.19](#).



Figura E.19: Opción de borrado de la tarea personal

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

En el desarrollo de este Trabajo de Fin de Grado he integrado los principios de sostenibilidad en la metodología de trabajo como en el producto final: una aplicación en Flutter con conexión a Moodle a través de su Web Service. Este trabajo ha reforzado mis competencias técnicas, y a su vez, también me ha servido para mejorar mi responsabilidad ética y mi visión estratégica para desarrollar soluciones tecnológicas que respeten los principios de los seres humanos y el entorno.

F.2. Competencias de sostenibilidad adquiridas

A lo largo de este proyecto he adquirido las siguientes nuevas competencias de sostenibilidad.

Análisis crítico

Capacidad de evaluación en relación a decisiones de diseño y arquitectura, enfocándose en los aspectos de eficiencia, accesibilidad, coste y beneficio a largo plazo.

Responsabilidad social

Priorización en la inclusión digital, asegurando que cualquier ser humano pueda usar la herramienta sin barreras de conocimiento.

Ética profesional

Gestión de los datos aplicando principios de transparencia y confidencialidad. Además, de proporcionar un documentación técnica para futuros desarrollos o revisiones.

Gestión de recursos

Optimización del tiempo, esfuerzo y energía de computación mediante el empleo de técnicas de desarrollo sostenible y automatización de procesos.

F.3. Sostenibilidad ambiental

Con el fin de reducir la huella ecológica del software se han seguido unas prácticas.

Tecnología seleccionada

Se eligió Flutter como tecnología de desarrollo porque tiene la capacidad de compilar a código nativo. Este proceso evita que las capas intermedias de tecnologías híbridas, reduciendo tiempos en el proceso de procesamiento, y por lo tanto, reduciendo la demanda energética.

Estrategias de carga de datos bajo demanda

La aplicación carga únicamente los datos del usuario al iniciar la sesión, y no permite efectuar una actualización de los mismos en ninguna otra pantalla. Esta estrategia evita transferencias de datos innecesarias.

F.4. Sostenibilidad social y accesibilidad

La aplicación refuerza la autonomía de la comunidad educativa al integrar herramientas que promueven la organización de tareas, así como facilitan la participación de todos los perfiles educativos que hagan uso de Moodle.

El diagrama de Gantt muestra de forma visual las actividades, ayudando a planificar y evitando solapamientos. La sección de tareas personales centraliza las responsabilidades académicas individuales en un único espacio, lo que reduce la dispersión de información. Además, el sistema de filtrado general permite al usuario filtrar contenidos, reduciendo la sobrecarga cognitiva y adaptándose a las necesidades de cada usuario.

F.5. Sostenibilidad económica y mantenimiento

Para garantizar la viabilidad del proyecto a largo plazo y optimizar los costes de su evolución, se ha aplicado un diseño modular, separando las distintas responsabilidades del sistema, lo que facilita la integración de nuevas funcionalidades. El código fuente se publica bajo los términos de una licencia libre, además la comunidad puede contribuir, corregir errores y reutilizar componentes para otros proyectos. De este modo, se minimizan los costes y se garantiza que la herramienta se mantenga segura, eficiente y alineada con las necesidades del usuario.

F.6. Conclusión

Este Trabajo de Fin de Grado me ha permitido integrar de forma realista los tres pilares sostenibilidad. He aprendido a optimizar el consumo de recursos evitando procesos innecesarios, a fomentar la inclusión de la comunidad educativa y a garantizar la continuidad y asequibilidad del proyecto. Este tiempo me ha brindado la lección, un enfoque que evalué cada decisión técnica en base a su impacto ambiental, social y económico, permite la creación de *software* práctico y sostenible a largo plazo, y me prepara para afrontar con éxito los retos profesionales.

Bibliografía

- [1] Blup. https://medium.com/@blup-tool/step-by-step-guide-to-installing-flutter-and-dart-on-windows-b30a631e7583. <https://medium.com/@blup-tool/step-by-step-guide-to-installing-flutter-and-dart-on-windows-b30a631e7583>, 2023.
- [2] Sofía González. ¿qué es la licencia bsd? <https://appmaster.io/es/blog/que-es-la-licencia-bsd>, 2023.
- [3] jaimetellezb. Flutter-estructura proyecto. <https://jaimetellezb.com/posts/flutter/2024/quickstart/26-scaffolding/>, 2024.
- [4] Mit License. The mit license (mit). <https://mit-license.org/>.
- [5] Seguridad Social. Bases y tipos de cotización. <https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537>, 2025.
- [6] Wikipedia. Apache license. https://es.wikipedia.org/wiki/Apache_License, 2025.