



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO DE FIN DE GRADO

Grado en Ingeniería Informática

Mención en Tecnologías de la Información

**Studlendar: aplicación enfocada en la organización y
gestión de horarios de estudio para estudiantes de
educación superior**

Autor:

Santiago Mateos Fernández

Tutoras:

María Jesús Rodríguez Triana

Alejandra Martínez Monés

Agradecimientos

Muchas gracias María Jesús y Alejandra por su ayuda y dedicación durante todo el proyecto, a mis amigos de la residencia por haber participado en las pruebas, y a mis padres, a José Luis y a Sonia por su apoyo constante.

Resumen

Cuando hablamos de estudios, la gestión del tiempo es un apartado que siempre se deja de lado, siendo un problema para los estudiantes. En respuesta a este problema, se planteó la realización de una aplicación que apoyara a los estudiantes a gestionar su tiempo de estudio y que pudieran mejorar su organización. Para poder crear una aplicación que fuese verdaderamente útil para estudiantes, se optó por aplicar una metodología basada en Diseño Centrado en el Usuario, gracias a la colaboración de un grupo de estudiantes. En este documento se recoge todo este proceso y los beneficios de haber seguido la metodología aplicada.

Abstract

When we talk about studies, time management is a topic that is usually forgotten and left aside, being a problem for students. In response to this problem, I proposed the creation of an application that would help students to manage their study time and improve their organization. In order to create an application that would be truly useful for students, it was decided to apply a methodology based on User-Centered Design, thanks to the collaboration of a group of students. This document describes the whole process and the benefits of having followed the applied methodology.

Índice general

1. Introducción	14
1.1. Contexto y Motivación	14
1.2. Objetivos	14
1.3. Organización del documento	15
2. Plan de proyecto	16
2.1. Metodología	16
2.1.1. Diseño Centrado en el Usuario	16
2.1.2. Diseño Iterativo	16
2.2. Planificación	17
2.2.1. Iteraciones de análisis	18
2.2.2. Iteraciones de desarrollo	19
2.3. Gestión de riesgos	20
2.3.1. Identificación de riesgos	20
2.3.2. Análisis y priorización de riesgos	20
2.3.3. Planificación de riesgos	21
2.4. Seguimiento del proyecto	22
2.4.1. Monitorización de riesgos	22
2.4.2. Planificación inicial vs planificación final	23
2.5. Presupuesto	24
3. Proceso de identificación de requisitos	25
3.1. Proceso de recogida de datos	25
3.1.1. Perfil de los participantes	25
3.1.2. Iteración 1	26
3.1.3. Iteración 2	30
3.1.4. Iteración 3	37
3.1.5. Cambios finales tras las iteraciones	41
3.2. Especificación de requisitos	46
3.2.1. Obtención de requisitos	46
3.2.2. Requisitos funcionales	46
3.2.3. Requisitos funcionales de información	46
3.2.4. Requisitos no funcionales	47
3.3. Diagrama de casos de uso	47
3.4. Descripción de casos de uso	49
3.4.1. Darse de alta en la aplicación	50
3.4.2. Iniciar sesión en la aplicación	50

3.4.3.	Configurar calendario	51
3.4.4.	Crear asignatura	51
3.4.5.	Importar eventos y bloques de clase	52
3.4.6.	Añadir bloque de estudio	52
3.4.7.	Editar bloque de estudio	53
3.4.8.	Borrar bloque de estudio	53
3.4.9.	Visualizar elemento del calendario	54
3.4.10.	Revisar progreso diario	54
3.4.11.	Revisar progreso semanal	55
3.4.12.	Comenzar pomodoro	55
3.5.	Modelo de dominio	56
4.	Fase de diseño	57
4.1.	Introducción	57
4.2.	Arquitectura del sistema	57
4.3.	Diagrama de transiciones de estados o interfaces	58
4.4.	Modelo de datos	59
4.5.	Diagrama del modelo conceptual de la base de datos	61
5.	Fase de Desarrollo	62
5.1.	Lenguajes de programación, entorno de desarrollo y herramientas	62
5.2.	Decisiones de seguridad sobre la base de datos	63
5.3.	Herramientas para accesibilidad	63
5.4.	Elección de la paleta de colores	64
6.	Fase de pruebas	65
6.1.	Pruebas de aceptación de la aplicación	65
6.1.1.	Iteración 1	66
6.1.2.	Iteración 2	69
6.2.	Pruebas de accesibilidad web de la aplicación	71
6.2.1.	Propuestas de mejora	72
7.	Conclusiones y trabajo futuro	73
7.1.	Conclusiones	73
7.2.	Limitaciones	73
7.3.	Trabajo futuro	74
7.4.	Valoración personal	74
	Bibliografía	75
A.	Formularios utilizados con los participantes	77
A.1.	Formulario perteneciente a la iteración de análisis nº2	77
A.2.	Formulario perteneciente a la iteración de análisis nº3	77
A.3.	Formulario perteneciente a la iteración de desarrollo nº1	77
A.4.	Formulario perteneciente a la iteración de desarrollo nº2	77

Índice de figuras

2.1. Visión general del DCU (imagen tomada de GeeksForGeeks)	17
2.2. Proceso seguido en las iteraciones de análisis	17
2.3. Diagrama de Gantt con la planificación inicial	18
2.4. Diagrama del sistema TAM (Figura extraída de TAM)	19
2.5. Diagrama de Gantt con la planificación inicial	23
2.6. Diagrama de Gantt con la planificación final	23
3.1. Gráfico sobre los perfiles de organización de los usuarios involucrados en las iteraciones de análisis	25
3.2. Formulario inicial clásico	27
3.3. Formulario inicial interactivo	27
3.4. Formulario inicial abierto	28
3.5. Grafico de elecciones de interfaz de la iteración 1	28
3.6. Boceto de la interfaz de calendario y notificación del sistema de feedback	29
3.7. Formulario inicial	31
3.8. Ventana emergente con una explicación sobre la Curva del Olvido	31
3.9. Interfaz de configuración de notificaciones	32
3.10. Interfaz de configuración de notificaciones con elección personalizada	32
3.11. Interfaz de calendario	33
3.12. Interfaz de revisión	33
3.13. Interfaz de feedback	34
3.14. Formulario inicial	37
3.15. Configuración de notificaciones	37
3.16. Interfaz de calendario en modo edición y ventana emergente con propuesta de calendario	38
3.17. Interfaz de calendario con notificaciones	38
3.18. Interfaz de modo estudio	39
3.19. Interfaz de revisión diaria	39
3.20. Interfaz de revisión semanal positiva	40
3.21. Interfaz de revisión semanal negativa	40
3.22. Interfaz del formulario inicial	42
3.23. Interfaz para añadir una asignatura	42
3.24. Interfaz para añadir horario a través de un fichero	43
3.25. Interfaz de configuración de notificaciones	43
3.26. Interfaz de calendario en modo edición y con propuesta	44
3.27. Interfaz de calendario	44
3.28. Interfaz de modo estudio	45
3.29. Interfaz de revisión diaria	45
3.30. Interfaz de revisión semanal	46

3.31. Diagrama de casos de uso, parte 1	48
3.32. Diagrama de casos de uso, parte 2	49
3.33. Iconos de las clases UML utilizados en diagramas de secuencia.	49
3.34. Diagrama de secuencia del caso de uso Dar Alta	50
3.35. Diagrama de secuencia del caso de uso Iniciar sesión	50
3.36. Diagrama de secuencia del caso de uso Configurar Calendario	51
3.37. Diagrama de secuencia del caso de uso Crear asignatura	51
3.38. Diagrama de secuencia del caso de uso Importar calendario	52
3.39. Diagrama de secuencia del caso de uso Añadir bloque de estudio	52
3.40. Diagrama de secuencia del caso de uso Editar bloque de estudio	53
3.41. Diagrama de secuencia del caso de uso Borrar bloque de estudio	53
3.42. Diagrama de secuencia del caso de uso Visualizar contenido bloque de estudio	54
3.43. Diagrama de secuencia del caso de uso Revisar progreso diario	54
3.44. Diagrama de secuencia del caso de uso Revisar progreso semanal	55
3.45. Diagrama de secuencia del caso de uso Comenzar pomodoro	55
3.46. Diagrama del modelo de dominio	56
4.1. Diagrama de la arquitectura del sistema	58
4.2. Diagrama de transiciones de interfaces	58
4.3. Modelo conceptual de la base de datos	61
6.1. Diagrama del sistema TAM utilizado en los formularios (Figura extraída de TAM)	65
6.2. Interfaz de propuestas	67
6.3. Interfaz de propuestas en su versión final	70
6.4. Captura de la interfaz de resultados de una auditoría realizada con Google LightHouse	71

Índice de tablas

2.1. Listado de riesgos	20
2.2. Listado de riesgos con reducción y mitigación	21
3.1. Listado de participantes en las iteraciones de análisis	26
3.2. Listado de propuestas de los usuarios de la iteración 1	29
3.3. Listado de propuestas de los usuarios de la iteración 2	34
3.4. Listado de propuestas de la iteración 3	40
6.1. Listado de participantes en las iteraciones de desarrollo	66
6.2. Resultados de los cuestionarios TAM	66
6.3. Listado de propuestas de la iteración de desarrollo nº1	68
6.4. Listado de propuestas de la iteración de desarrollo nº2	69
6.5. Listado de propuestas de la iteración de desarrollo nº2	70

Capítulo 1

Introducción

1.1. Contexto y Motivación

En la actualidad los estudiantes necesitan poder organizar su estudio, sentir motivación por lo que hacen y realizar la gran cantidad de tareas que los estudios de educación superior exigen. Para conseguir esto es necesario tener una buena gestión del tiempo y un calendario que permita a cada estudiante organizarse. También puede ser útil aplicar técnicas como autoseguimiento o la creación de un diario, tal y como se menciona en [14].

Pero no solo eso, los estudiantes deben sentirse realizados, deben sentir que progresan. Como se afirma en este blog [14] y en los artículos [7] y [11], las pequeñas victorias tienen un gran impacto y pueden cambiar la sensación de un mal día por la de uno bueno. También se destaca la importancia del autoseguimiento, utilizando indicadores cuantitativos del progreso diario o semanal.

Sin embargo, la procrastinación está a la orden del día. Así lo afirma un estudio de la Universidad de Guadalajara, México ([22]), en el que se establece una clara correspondencia entre la procrastinación y la motivación entre estudiantes universitarios. En el estudio se afirma que los estudiantes con un alto grado de procrastinación acaban teniendo una baja motivación para continuar sus estudios.

Ante esta necesidad, en la parte de prácticas de la asignatura de Profesión y Sociedad y de forma grupal, nace la idea de crear una aplicación de apoyo a los estudiantes. Esta aplicación no pretende ser una simple herramienta de calendario genérica, sino una herramienta de ayuda que facilite la organización, progreso y desarrollo en los estudios y, en última instancia, motivando a los estudiantes a la consecución de sus metas.

Partiendo de esa idea inicial, y con el permiso de los miembros del grupo, este proyecto retoma esa propuesta, ajustándola al alcance de un Proyecto Fin de Grado, siendo la idea inicial renovada casi en su totalidad, y concretando su diseño en las necesidades y sugerencias de los estudiantes, gracias a la involucración de grupo de estudiantes en el proceso, como se mencionará en el capítulo 3 y sus correspondientes secciones.

1.2. Objetivos

Tras las premisas del apartado anterior, el objetivo principal del proyecto Studlendar es:

- *Diseñar y desarrollar una aplicación web que permita a los estudiantes organizarse y hacer un seguimiento de su plan de trabajo..*

Además de este objetivo principal, el proyecto plantea otro reto importante, saber si la idea es aceptable por los usuarios. Dada la diversidad de estudiantes, la diferencia de conocimientos en materia tecnológica y la variedad de estudios, se pretende integrar a los potenciales usuarios en el proceso de investigación, implementando técnicas de diseño centrado en el usuario. Esto ha dado lugar a un objetivo más:

- *Alinear las funcionalidades de la herramienta con las necesidades y preferencias de los usuarios finales.*

El diseño centrado en el usuario es de vital importancia en una herramienta web dirigida a usuarios finales, tal y como se afirman diversos artículos científicos (por ejemplo [4], [13] y [23]). En éstos se hace hincapié en la utilización de un enfoque centrado en el usuario para la obtención de requisitos, tanto técnicos como prácticos. Además, utilizar un enfoque centrado en el usuario permite un mejor desarrollo del proyecto gracias a las necesidades y propuestas que los usuarios puedan plantear ([16]).

1.3. Organización del documento

Tras el capítulo introductorio, el resto del documento presenta la siguiente estructura:

- **Capítulo 2: Plan de proyecto.** En este capítulo se explican las metodologías, la planificación seguida, los riesgos del proyecto y el presupuesto.
- **Capítulo 3: Fase de análisis.** En este apartado se desarrolla el contenido de las entrevistas realizadas, los prototipos de la aplicación, requisitos y casos de uso.
- **Capítulo 4: Fase de diseño.** En este apartado se detalla todo lo relacionado con la arquitectura del sistema, modelo de datos y proceso de diseño de la aplicación.
- **Capítulo 5: Fase de desarrollo.** En esta parte se mencionan las herramientas utilizadas para el desarrollo de la aplicación.
- **Capítulo 6: Fase de pruebas.** En este capítulo se ilustran las pruebas de la aplicación y de seguridad, comprobando así el correcto funcionamiento de la misma.
- **Capítulo 7: Conclusiones y trabajo futuro.** Aquí se muestran las conclusiones obtenidas al final del proyecto y el trabajo futuro.

Capítulo 2

Plan de proyecto

2.1. Metodología

Como se ha explicado en el capítulo previo, el objetivo principal de este proyecto es desarrollar una aplicación web que permita a los estudiantes organizarse y hacer un seguimiento de su plan de trabajo, teniendo en cuenta que cada estudiante tiene unos niveles de conocimiento en tecnología y una organización diferentes. De ahí que se escoja a un grupo diverso de estudiantes para involucrarlos en la fase de análisis, que es el cimiento del proyecto.

Teniendo esto en cuenta, y siguiendo el libro de gestión de proyectos *Software Project Management, 5th edition* ([8]), la metodología escogida para el desarrollo de este proyecto ha sido una combinación de diseño centrado en el usuario y desarrollo iterativo. A continuación se explicará cada una de las metodologías.

2.1.1. Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario (DCU) es un proceso de diseño iterativo que se centra en los usuarios y sus necesidades en cada fase del proceso de diseño ([10]). En este proceso se utilizan una mezcla de métodos para entender las necesidades del usuario, como entrevistas y cuestionarios. Este tipo de métodos son de utilidad para captar lo que el usuario necesita, y se han utilizado en este proyecto para obtener propuestas sobre la aplicación y cambios sobre los prototipos realizados, de cara a mejorar la experiencia de usuario y acercarse lo máximo posible a lo que los usuarios necesitan, fomentando así su adopción.

En el Capítulo 3 se explican en profundidad las entrevistas realizadas por cada iteración, el grupo de estudiantes seleccionados, los prototipos realizados para cada iteración y los resultados finales.

2.1.2. Diseño Iterativo

El DCU por su propia naturaleza iterativa, es compatible con diferentes metodologías de diseño y desarrollo. Una metodología de proyecto que se basa en iteraciones, y se combina frecuentemente con DCU, es el Diseño Iterativo.

El Diseño Iterativo es una práctica que permite elaborar, refinar y mejorar un proyecto, producto o iniciativa (ver [24]). Una de las principales ventajas de este enfoque junto con el DCU es la posibilidad de mejorar el producto final en cada iteración gracias al feedback obtenido mediante los métodos de DCU previamente mencionados. La siguiente sección presenta cómo se ha implementado, de forma conjunta, DCU con el Diseño Iterativo en este proyecto.

2.2. Planificación

El enfoque seguido para la realización de este trabajo es la combinación de DCU junto con Diseño Iterativo, haciendo especial énfasis en el DCU.

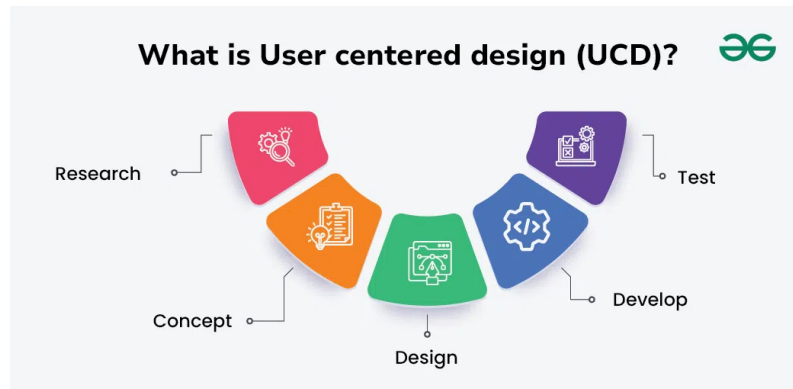


Figura 2.1: Visión general del DCU (imagen tomada de GeeksForGeeks)

Utilizando el DCU en la fase de análisis, se pudo obtener gran cantidad de ideas, propuestas y mejoras de cambio a lo largo de las diversas iteraciones realizadas. Implementando iteraciones (ver Figura 2.2) junto con el DCU en esta fase permitió obtener una serie de requisitos y bocetos más cercanos a las necesidades de los usuarios de cara a las fases de Diseño y Desarrollo.



Figura 2.2: Proceso seguido en las iteraciones de análisis

Durante el Capítulo 3 se detallarán las pruebas de concepto realizadas (elaborando un cuestionario que sirviera como guión), los bocetos utilizados en las mismas y los resultados finales obtenidos al finalizar cada iteración.

Una vez terminada la fase de análisis, se procedió con las fases de Diseño y Desarrollo, utilizando los requisitos y bocetos creados en la fase anterior. Estas dos fases fueron realizadas de forma paralela, principalmente por cuestiones de desarrollo, que propiciaron algunos cambios en el diseño, tal y como se puede ver en el Diagrama 2.5.

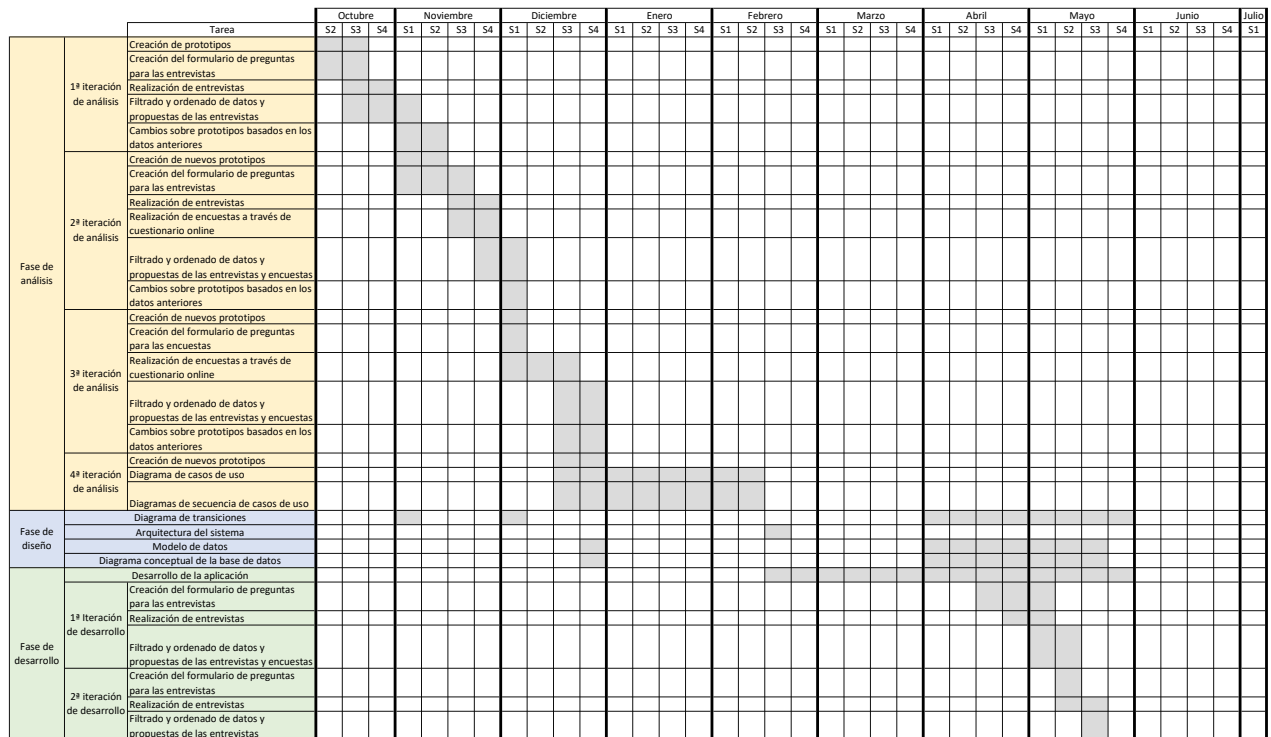


Figura 2.3: Diagrama de Gantt con la planificación inicial

2.2.1. Iteraciones de análisis

En la fase de análisis, se realizaron tres iteraciones, que buscaban implementar técnicas propias de DCU e involucrar a un grupo de personas con diferentes perfiles de organización en los estudios (como se muestra en la Figura 3.1), para obtener ideas desde diferentes perspectivas, obteniendo una idea más refinada de las necesidades de los usuarios en cada iteración. A continuación se describe brevemente el objetivo y contenido de cada iteración:

- **Primera iteración (16/10/2024 - 25/10/2024):** La primera iteración tenía como propósito principal hacer una toma de contacto con los usuarios, plantear la idea básica de la aplicación (una aplicación de calendario para ayudar a los estudiantes a organizarse) y apoyar esa idea con unos bocetos a modo de prueba de concepto que permitieran los usuarios visualizar la aplicación. Permitió obtener propuestas de diverso tipo y entender algunas de las necesidades de los usuario de cara a siguientes iteraciones.
- **Segunda iteración (26/10/2024 - 7/11/2024):** Tras haber refinado los prototipos con las diferentes propuestas de la primera iteración, se procedió a una segunda, con preguntas más concretas sobre diversos prototipos y funcionalidades de la aplicación. El objetivo de esta segunda iteración era añadir más funcionalidad y obtener más propuestas para refinar los prototipos de cara a la última iteración.
- **Tercera iteración (8/11/2024 - 21/11/2024):** La última iteración fue la más importante y la de mayor tamaño a nivel de funcionalidades, puesto que abarcaba desde el formulario inicial hasta la revisión del progreso. Se presentaban unos prototipos más detallados, mostrando así una idea general de como se vería la aplicación implementada y las interacciones aplicación-usuario. El objetivo de esta iteración era implementar cambios y propuestas obtenidas en las iteraciones anteriores en los diferentes bocetos y analizarlas con los

usuarios para definir correctamente lo que sería la aplicación final, siempre teniendo en cuenta el alcance del proyecto.

Tras evaluar los resultados de cada iteración, se crea una tabla de propuestas de cada iteración (Tablas 3.2, 3.3 y 3.4), asignando a cada propuesta un valor de alineación con el proyecto (alto, medio y bajo), y una factibilidad (sí o no). Estos valores fueron asignados por el autor del proyecto, teniendo en cuenta el alcance de la aplicación y el tiempo para la realización del proyecto. Estas tablas sirvieron para seleccionar qué propuestas se llevarían a la siguiente iteración y cuáles quedarían descartadas.

Estas iteraciones en la fase de análisis permitieron tener un modelo claro de la aplicación, con funcionalidades y requisitos, facilitando así las fases posteriores de diseño y desarrollo, y reduciendo algunos riesgos que se explicarán posteriormente en la Tabla 2.1.

2.2.2. Iteraciones de desarrollo

Además de las iteraciones de análisis, y buscando detectar posibles problemas y mejoras en una fase temprana de desarrollo de la aplicación, se optó por realizar dos pruebas de aceptación con los usuarios durante el desarrollo de la aplicación. Se utilizó para ello cuestionarios TAM (Technology Acceptance Model, [19]). Este modelo utiliza tres factores principales para crear una predicción sobre la utilización del sistema actual:

- **Utilidad percibida:** factor que pregunta al usuario si cree que el sistema será fácil o difícil de usar.
- **Facilidad de uso percibida:** factor que evalúa si el usuario cree que el sistema será útil para realizar las tareas necesarias.
- **Actitud hacia el uso:** factor que se refiere a los sentimientos generales del usuario sobre el sistema.

Con estos tres factores, se crea una predicción (intención de uso de la aplicación), que se evalúa junto con los factores anteriores para obtener el uso real del sistema, tal y como se muestra en la Figura 2.4.

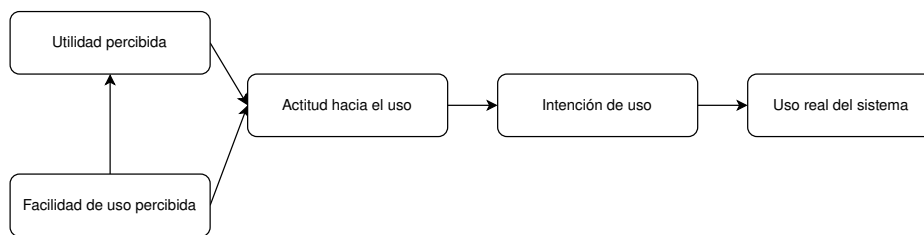


Figura 2.4: Diagrama del sistema TAM (Figura extraída de TAM)

- **Primera iteración (29/03/2025 - 03/06/2025):** La primera iteración en la fase de desarrollo tenía como objetivo recabar información sobre la aplicación en una etapa temprana del desarrollo, siguiendo el modelo TAM (2.4) para diseñar las pruebas de aceptación. Estas encuestas fueron de gran utilidad, puesto que sirvieron para verificar los requisitos obtenidos durante las iteraciones de análisis, encontrar algunos errores y mejoras tanto a nivel visual como a nivel programación, para mejorar así la calidad de la aplicación y asegurar un mayor grado de satisfacción del usuario final.
- **Segunda iteración (04/06/2025 - 24/06/2025):** Habiendo resuelto los problemas encontrados en la primera iteración y continuando con el desarrollo, se procede a realizar una segunda iteración para medir la aceptación de la aplicación por los usuarios, usando de nuevo un cuestionario TAM. Ésta serviría para terminar de cerrar el proceso de desarrollo de este proyecto, arreglando algún fallo encontrado y terminando las funcionalidades más importantes.

Estas dos iteraciones en conjunto permitieron, además de comprobar los requisitos y mejorar las funcionalidades principales, obtener una idea general sobre si la idea del proyecto era aceptada por los usuarios y merecía la pena seguir una línea de trabajo futuro con la aplicación.

2.3. Gestión de riesgos

El análisis de riesgos es un apartado de gran importancia en cualquier proyecto software, tal y como se menciona en el Capítulo 7 de [8]. Además, en este capítulo, se nos indica los pasos a seguir en la planificación de riesgos, tales como *identificación de riesgos*, *análisis y priorización de riesgos* y *planificación de riesgos*. Estos pasos serán explicados en las siguientes subsecciones.

2.3.1. Identificación de riesgos

Según el capítulo previamente mencionado, la identificación de riesgos se puede realizar mediante dos enfoques: *listas* y *brainstorming*. En este proyecto se ha utilizado una mezcla de los dos enfoques para la identificación de riesgos, obteniéndose la Tabla 2.1.

2.3.2. Análisis y priorización de riesgos

Tras haber identificado los riesgos, hay que hacer un análisis de los mismos y asignar prioridad a los mismos. Como se explica en el capítulo 7 de [8], la exposición a un riesgo (RE o Risk Exposure) se calcula como el producto de la probabilidad de ocurrencia de un riesgo (probabilidad de que el riesgo se manifieste) por el impacto de este riesgo en el proyecto (cuánto de grave es el riesgo para el proyecto):

$$RE = \text{probabilidad de ocurrencia} * \text{impacto}$$

En el documento escrito por Barry W. Boehm, "Software Risk Management: Principles and Practices" (ver [1], Sección Risk-identification checklist), se proporciona una lista del top 10 riesgos en proyectos software, y se tendrá en cuenta para la realización de la Tabla 2.1.

En este mismo capítulo, Barry Boehm recomienda que, para estimar de forma precisa estos riesgos, se use una escala relativa del 0 al 10 para la probabilidad de ocurrencia y el impacto. Es la escala que se utiliza en la Tabla 2.1, que se presenta a continuación:

Tabla 2.1: Listado de riesgos

Referencia	Riesgo	Probabilidad	Impacto	Exposición
R1	Problemas en dependencias de terceros (por ejemplo Atlas MongoDB)	9	8	72
R2	Errores de estimación de plazos	8	8	64
R3	Cambios en los requisitos	3	7	21
R4	Implementación incorrecta de funcionalidades	5	7	35
R5	Problemas de accesibilidad web	5	7	35
R6	Errores durante el desarrollo	5	6	30
R7	"Gold plating" ¹	5	6	30

Continúa en la siguiente página

Tabla 2.1: Listado de riesgos (Continuación)

R8	Estancamiento por falta de conocimiento en algún apartado	4	5	20
R9	Problemas con los sistemas del desarrollador	2	4	8
R10	Enfermedad o indisponibilidad del desarrollador	2	2	4

¹ Mala práctica que consiste en la incorporación de más funcionalidades a los entregables del proyecto a pesar de no estar debidamente documentadas en los requisitos.

2.3.3. Planificación de riesgos

Después de haber identificado, analizado y priorizado los diferentes riesgos que pueden afectar al proyecto, hay que planificar como afrontarlos si llegan a manifestarse. Hay cuatro formas de afrontar los riesgos según [8]:

- **Aceptación:** aceptamos el impacto de un riesgo si llega a ocurrir.
- **Evitación:** evitamos el riesgo evitando realizar la actividad que lo puede provocar.
- **Reducción y mitigación:** establecer un plan de contingencia para tratar los impactos si los riesgos se llegan a manifestar.
- **Transferencia:** transferencia del riesgo a otra persona o entidad.

Para este proyecto, utilizaremos la tercera forma, reducción y mitigación, tal y como se muestra en la tabla 2.2. Aunque puedan parecer términos similares, en el contexto de gestión de proyectos no lo son. Reducción de un riesgo es el intento de reducir la probabilidad de que este ocurra, mientras que mitigación se refiere a la acción que se toma cuando el riesgo ocurre, para reducir su impacto en el proyecto.

Tabla 2.2: Listado de riesgos con reducción y mitigación

Referencia	Reducción	Mitigación
R1	Utilización de herramientas de terceros que sean ampliamente utilizadas para evitar problemas o caídas	Valorar un cambio de herramienta si el problema o caída es muy grave, o esperar
R2	Planificación de tareas con tiempo suficiente y dejando un margen de actuación por si se atrasan	Replanificación de las tareas o ajustes en el alcance del proyecto
R3	Definir bien todos los requisitos y el alcance de proyecto. Tanto DCU como Entrega Incremental previenen de que esto ocurra	Revisar todos los requisitos y contactar con los usuarios para acordar los nuevos
R4	Revisar bien todos los requisitos, modelo de dominio y datos. DCU y Entrega Incremental previenen de que esto ocurra	Utilizando los incrementos, la ocurrencia de este riesgo es menor y tiene menor impacto

Continúa en la siguiente página

Tabla 2.2: Listado de riesgos con reducción y mitigación (Continuación)

R5	La utilización de herramientas para la revisión de la accesibilidad web previene este problema	Si el error es detectado antes de avanzar de incremento, arreglarlo. En caso contrario, volver al incremento en el que sí funcionaba y revisar desde ese
R6	Realizar pruebas de calidad en cada incremento para evitar posibles errores antes de avanzar al siguiente incremento	Si el error es detectado antes de avanzar de incremento, arreglarlo. En caso contrario, volver al incremento en el que sí funcionaba y revisar desde ese
R7	DCU previene de este riesgo mediante las entrevistas a los usuarios	Realizar reuniones o encuestas a los usuarios para revisar que la funcionalidad es útil
R8	Antes de empezar el desarrollo, revisar las áreas que son necesarias y formarse en las mismas	Preguntar a alguien con experiencia en la materia y/o revisar documentación
R9	Mantener el contenido del proyecto en un repositorio remoto y crear copias de seguridad periódicas	Utilizar la última copia de seguridad realizada
R10	Planificar con tiempo suficiente las tareas por si se diera tal riesgo	Aprovechar el margen o el tiempo extra asignado a las tareas si llegase a ocurrir

2.4. Seguimiento del proyecto

2.4.1. Monitorización de riesgos

Durante el desarrollo del proyecto y de la aplicación se manifestaron algunos de los riesgos descritos en la Tabla 2.1. A continuación se describirán cuáles fueron y el impacto que provocaron en el proyecto:

1. **Riesgo R1 - Problemas en dependencias de terceros:** Si bien se utilizaron herramientas y aplicaciones ampliamente utilizadas, no se está exento de posibles problemas. Este riesgo se manifestó en contadas ocasiones, y ocurrió principalmente con MongoDB y Atlas. Sus servidores se cayeron en un par de ocasiones, haciendo imposible utilizar la aplicación para gestionar la base de datos en ese periodo.
2. **Riesgo R2 - Errores de estimación de plazos:** De los riesgos que se manifestaron, éste fue el que más a menudo ocurrió, puesto que se subestimó, en más de una ocasión, la complejidad de las tareas a realizar. Este riesgo era bastante previsible, y se asignó el segundo RE más elevado de todos los riesgos (64).
3. **Riesgo R6 - Errores durante el desarrollo:** Durante la fase de desarrollo se encontraron algunos problemas, principalmente con el tratamiento de horas y fechas, que provocaron un retraso en la finalización de la aplicación.
4. **Riesgo R8 - Estancamiento por falta de conocimiento en algún apartado:** Durante el comienzo del desarrollo de la aplicación y a lo largo del mismo se tuvo que revisar la documentación sobre algunos métodos y funciones específicos de React, Remix y JS. Si bien no provocaron un retraso significativo, el riesgo si se manifestó.

2.5. Presupuesto

Este proyecto, bajo el contexto de un trabajo de fin de grado, no tiene gastos reales en términos de remuneración, pero si cuenta con un presupuesto de 300 horas en términos de dedicación. Aún así, para crear un presupuesto remunerado realista, se hace una estimación sobre el coste como si una empresa estuviera realizando este proyecto. Se tienen en cuenta los siguientes apartados:

- **Coste de personal:** El coste que supone un programador junior a una empresa en España es de aproximadamente 20.000€ de salario + 30 % de costes por la Seguridad Social, unos 26.000€. Teniendo en cuenta que se generan 1,8 días de vacaciones por mes trabajado, tendríamos un total de 227 días laborales (descontando una media de 10 días festivos y fines de semana). Si la jornada es de 8 horas diarias, a la empresa le costaría, por hora: $26.000\text{€} / (227 \text{ días} * 8 \text{ horas}) \simeq 15\text{€} / \text{hora}$. Con estos cálculos y una estimación temporal para el proyecto de unas 300 horas, tendríamos que el coste de personal serían $15\text{€} * 300 \text{ horas} = 4500\text{€}$.
- **Coste del hardware:** El hardware utilizado para este proyecto ha consistido de un ordenador de sobremesa que costó en el momento de compra de unos 1500€ y tiene una vida útil estimada de unos ocho años (96 meses). La duración estimada del proyecto es de nueve meses. Para calcular el coste del hardware, utilizaremos la fórmula de la depreciación: $9 / 96 * 1500 \simeq 141\text{€}$.
- **Coste del software/licencias:** El coste por software o licencias de uso es de **0€**, puesto que se han utilizado herramientas de código libre o herramientas en su versión gratuita.

Sumando los costes obtenemos un presupuesto aproximado de $4500\text{€} + 141\text{€} + 0\text{€} = 4641\text{€}$. Si tenemos en cuenta la planificación final del proyecto (Figura 2.6) y su retraso en unas tres semanas, el sobrecoste de personal sería: $8 \text{ horas} * 21 \text{ días} * 15\text{€} / \text{hora} = 2520\text{€}$ de sobrecoste. El presupuesto final sería de $4641\text{€} + 2520\text{€} = 7161\text{€}$.

Capítulo 3

Proceso de identificación de requisitos

3.1. Proceso de recogida de datos

Para la realización de las iteraciones de análisis mencionadas en la Sección 2.2.1, se optó por presentar a los participantes una serie de bocetos en formato digital mediante la herramientas Moqups, que básicamente son un ‘primer modelo de un producto o servicio que se lleva intención de testear’ ([20]). Los bocetos son una forma sencilla de presentar a los participantes una idea de cómo sería la aplicación en funcionamiento, sin tener que programarla, y poder recabar ideas que emerjan de los participantes, lo que los hace idóneos para el desarrollo de este proyecto.

3.1.1. Perfil de los participantes

Los perfiles de los participantes escogidos son variados incluyendo 1 estudiantes universitarios, una persona cursando estudios de educación superior fuera del ámbito universitario (grado superior) y un futuro usuario cursando el bachillerato de adultos. Para conocer los perfiles de organización del grupo de usuarios involucrado en las iteraciones, utilizamos las siguientes definiciones como referencia, y pedimos a los participantes que se identificaran con una de ellas, tal y como se muestra en la Figura 3.1:

- **Organizado:** persona que organiza su jornada y realiza las tareas con tiempo suficiente.
- **Procastinador:** persona que prefiere dejar el trabajo para el último momento y realizar tareas con menos tiempo de margen.
- **Espontáneo y adaptativo:** persona que se adapta a medida que van surgiendo tareas, permutando entre ellas según la necesidad del momento.

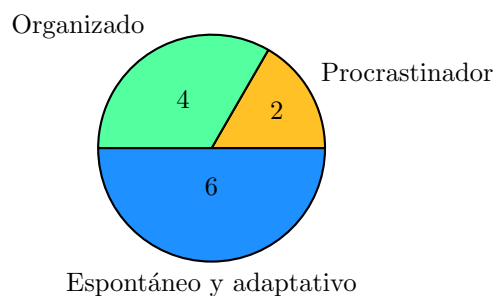


Figura 3.1: Gráfico sobre los perfiles de organización de los usuarios involucrados en las iteraciones de análisis

Cabe destacar que todos los datos tratados y presentados a lo largo de este documento han sido anonimizados para preservar el anonimato de los participantes y dar el mismo valor a toda la información obtenida.

A continuación se detallan, en la Tabla 3.1, los estudios, perfil de organización (ver gráfico 3.1) y participación (representado con una X) de cada uno de los participantes en las diferentes iteraciones:

Tabla 3.1: Listado de participantes en las iteraciones de análisis

Código de participante	Estudios	Perfil de organización	Participación iteración análisis 1	Participación iteración análisis 2	Participación iteración análisis 3
P1	Grado Superior en Sistemas Electrotécnicos y Automatizados	Organizado	X	X	X
P2	Grado en Administración y Dirección de Empresas	Procastinador	X	-	-
P3	Grado en Medicina	Procastinador	X	X	X
P4	Grado en Ingeniería Informática	Organizado	X	X	X
P5	Bachillerato de adultos	Espontáneo y adaptativo	X	X	-
P6	Grado en Ingeniería Industrial	Organizado	X	X	X
P7	Grado en Ingeniería Química	Espontáneo y adaptativo	X	X	-
P8	Grado en Ingeniería Mecánica	Espontáneo y adaptativo	X	X	-
P9	Máster en Arquitectura	Espontáneo y adaptativo	X	X	X
P10	Grado en Matemáticas	Organizado	-	X	X
P11	Doble Grado en Matemáticas y Física	Espontáneo y adaptativo	-	X	X
P12	Grado en Ingeniería Informática	Espontáneo y adaptativo	-	-	X

3.1.2. Iteración 1

El enfoque de esta primera iteración era ubicar y centrar la idea de la aplicación, mostrando unos bocetos sencillos sobre cómo se vería la aplicación y pidiendo retroalimentación a los usuarios. Cabe destacar que los bocetos estaban realizados de forma superficial y sin mucho detalle, de forma que los participantes pudieran aportar sus ideas sobre la aplicación.

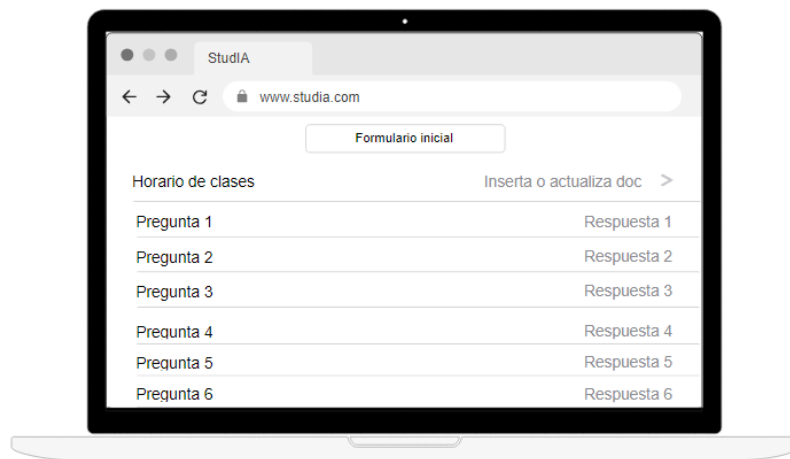
En la Iteración 1 se presentó, de forma presencial e individual, el objetivo del proyecto y la idea de la aplicación a los usuarios de forma oral. Tras una breve presentación de la idea, se preguntó a cada uno de los participantes por sus preferencias respecto al tipo de formulario inicial (i.e., el formulario de creación del calendario y donde se puede configurar la aplicación).

También se preguntó sobre otras interfaces, como la del calendario y las notificaciones, de forma que los participantes pudieran expresar su opinión, lo que no y lo que sí les gustaría ver en la aplicación.

Los estilos del formulario inicial que se presentaron en esta iteración son tres:

- **Formulario clásico:** La idea del formulario clásico es seguir una serie de preguntas clave para poder crear el calendario, tal y como se muestra en la Figura 3.2. Cabe destacar que las preguntas concretas no se presentaron en esta iteración, y por eso el prototipo no muestra preguntas como tal.

Esta interfaz solo fue aceptada por dos participantes. El resto prefirieron el formato interactivo (Figura 3.3) o un formato híbrido entre este y el interactivo.

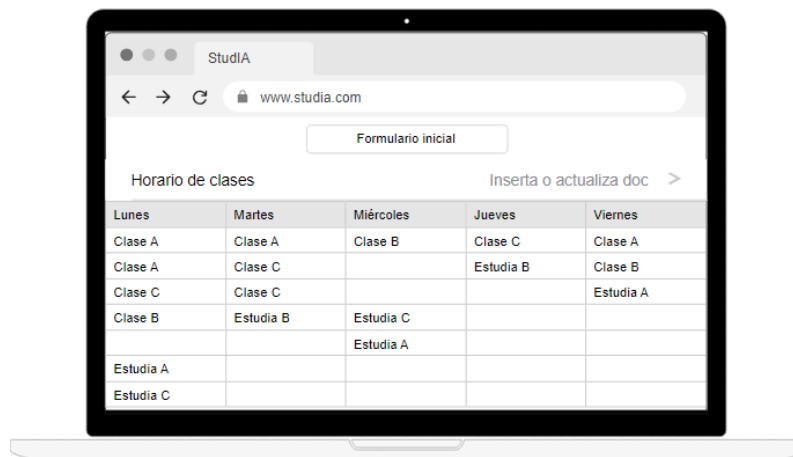


Formulario inicial	
Horario de clases Inserta o actualiza doc >	
Pregunta 1	Respuesta 1
Pregunta 2	Respuesta 2
Pregunta 3	Respuesta 3
Pregunta 4	Respuesta 4
Pregunta 5	Respuesta 5
Pregunta 6	Respuesta 6

Figura 3.2: Formulario inicial clásico

- **Formulario interactivo:** El formulario interactivo pretende mostrar una interfaz más similar a Google Calendar, con un calendario en blanco en formato tabla donde el usuario pueda añadir diferentes bloques que representen sus asignaturas, tal y como se muestra en la Figura 3.3.

Esta interfaz también recibió el apoyo de únicamente dos participantes.



Formulario inicial				
Horario de clases Inserta o actualiza doc >				
Lunes	Martes	Miércoles	Jueves	Viernes
Clase A	Clase A	Clase B	Clase C	Clase A
Clase A	Clase C		Estudia B	Clase B
Clase C	Clase C			Estudia A
Clase B	Estudia B	Estudia C		
Estudia A		Estudia A		
Estudia C				

Figura 3.3: Formulario inicial interactivo

- **Formulario abierto:** El formulario abierto es una interfaz de texto libre, permitiendo al usuario escribir todo lo que necesita en el calendario, asignaturas, horarios y horas de estudio, tal y como se muestra en la Figura 3.4. La entrada de usuario sería procesada posteriormente con inteligencia artificial para extraer la información requerida.

Esta interfaz fue descartada puesto que a ningún participante le gustó.

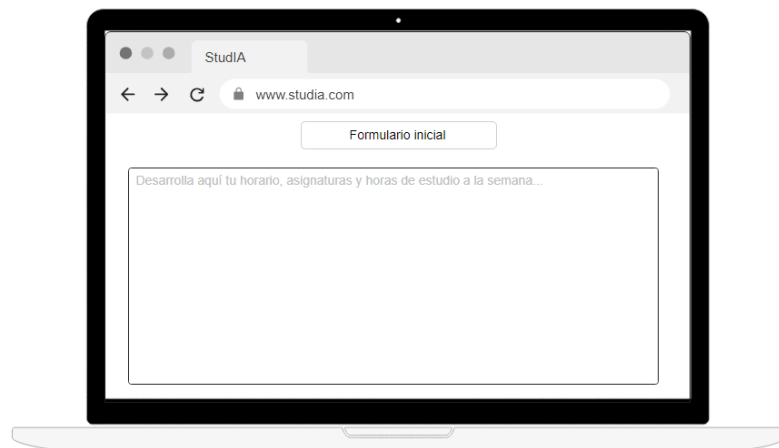


Figura 3.4: Formulario inicial abierto

Los resultados de la elección de interfaz sobre el formulario inicial se presentan en la Figura 3.5, donde los participantes se posicionaron hacia una mezcla entre el formulario clásico y el interactivo. Cabe destacar que los formularios clásico e interactivo se expresan de forma separada porque así se hizo en las pruebas de concepto, aunque después del análisis de la primera iteración queden fusionadas en una sola, representando el formulario híbrido.

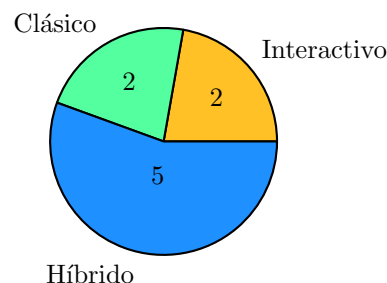


Figura 3.5: Grafico de elecciones de interfaz de la iteración 1

A mayores de la interfaz sobre el formulario inicial, se realizaron bocetos de otras partes de la aplicación, como el calendario y una idea temprana del sistema de feedback (Figura 3.6), y se preguntó a los usuarios por éstas.

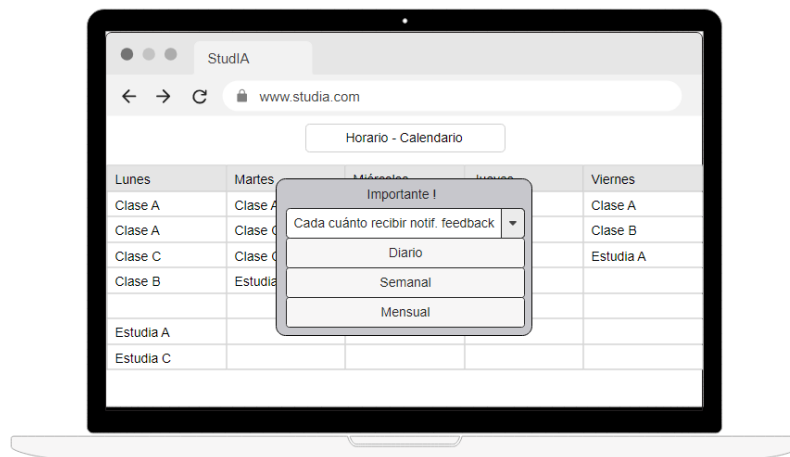


Figura 3.6: Boceto de la interfaz de calendario y notificación del sistema de feedback

También se recogió información extra que los usuarios expresaron, como propuestas de cambio sobre las interfaces mostradas durante la prueba de concepto como nuevas ideas sobre funcionalidades deseadas. Las interfaces, ideas y cambios de esta iteración se encuentran recogidas en la Tabla 3.2.

Las propuestas realizadas por los participantes a lo largo de las diversas iteraciones pueden verse en las Tablas 3.2, 3.3 y 3.4, divididas en cuatro columnas:

- **Columna 1, propuesta:** Nombre de la propuesta.
- **Columna 2, descripción:** Descripción breve de la propuesta.
- **Columna 3, grado de alineación con el proyecto:** Valor sobre la alineación de la propuesta con el proyecto, en un rango de tres valores: alto, medio y bajo.
- **Columna 4, factibilidad:** Factibilidad de la propuesta a nivel de desarrollo, con dos valores: sí o no. Las propuestas con el valor "no" han sido rechazadas por complejidad o porque se salen del alcance actual del proyecto y no han sido implementadas en la aplicación.

Tabla 3.2: Listado de propuestas de los usuarios de la iteración 1

Código de Propuesta	Propuesta	Descripción	Grado de alineación con el proyecto	Factibilidad
IT1.01	Bloques o sesiones	Sesiones de estudio para las diferentes asignaturas que estén en el horario	Alto	Sí
IT1.02	Método pomodoro	Método para la gestión del tiempo para dividir el tiempo en intervalos fijos	Alto	Sí
IT1.03	Notificaciones con consejos	Sistemas de notificaciones con consejos de estudio, organización, salud...	Medio	No
IT1.04	Eventos	Introducción de eventos dentro del calendario, no simplemente bloques o sesiones de estudio	Medio	Sí

Continúa en la siguiente página

Tabla 3.2: Listado de propuestas de los usuarios de la iteración 1 (Continuación)

IT1.05	Opción de "tachar" tareas	Opción que permite marcar tareas como completadas	Medio	Sí
IT1.06	Control del estado emocional	Pregunta sobre el estado emocional del estudiante en la revisión semanal o diaria	Bajo	No
IT1.07	Pregunta sobre fines de semana	Pregunta sobre la gestión del tiempo durante fines de semana (permitir o bloquear fines de semanas para bloques de estudio)	Bajo	No

Los resultados obtenidos de la Iteración 1 fueron significativos y de gran valor para las iteraciones posteriores. Un resultado de gran importancia fue el estilo de formulario, descartando totalmente el formulario abierto y optando por una opción híbrida entre el formulario clásico y el interactivo, tal y como se ve en la Figura 3.5.

Las propuestas de los participantes permitieron centrar y ampliar la idea de la aplicación (ver Tabla 3.2), descartando algunas ideas iniciales (IT1.4), tomando otras ideas (IT1.5, IT1.6, IT1.8 e IT1.10) y refinando los prototipos de cara a las siguientes iteraciones.

En las siguientes iteraciones se trabajó con un formulario inicial híbrido, como se muestra en la Figura 3.5. También se tuvieron en cuenta las propuestas con valor de alineación con el proyecto alto y medio, puesto que son las que más valor aportan a la aplicación, siempre teniendo en cuenta el alcance actual del proyecto.

3.1.3. Iteración 2

Una vez terminada la primera iteración, se procedió a utilizar todos los datos obtenidos para crear una segunda prueba. Se modificó el formulario inicial para crear un formato híbrido y se escogió una serie de propuestas evaluadas como factibles (propuestas IT1.5, IT1.6, IT1.8 e IT1.1 de la Tabla 3.2), que se tuvieron en cuenta para refinar los prototipos, crear algunos nuevos y profundizar en las funcionalidades de la herramienta. Para contrastar los resultados con los usuarios, se planteó una segunda prueba, esta vez de manera híbrida (presencial y online utilizando un formulario como hilo conductual), donde se pedía opinión sobre los nuevos prototipos e identificar cambios que los usuarios realizarían en éstos.

A continuación se mostrarán los bocetos utilizados en esta iteración, que son más detallados que los de la iteración anterior, con el objetivo de identificar nuevas ideas y potenciales cambios propuestos por parte de los participantes.

Un apartado de gran relevancia en esta iteración fue el formulario inicial, modificado para que fuera una combinación entre el formulario clásico e interactivo, esto es, un formulario híbrido. El formulario inicial se acerca más a su forma final, como observamos en la Figura 3.7. Se añadieron preguntas relevantes y algunas que los usuarios habían planteado (i.e., Curva del Olvido, frecuencia de las sesiones):

Figura 3.7: Formulario inicial

Otro apartado relevante que se añadió en esta iteración fueron las ventanas emergentes con mensajes de ayuda o aclaratorios sobre apartados del formulario, como se puede observar en la Figura 3.8. Este tipo de mensajes de ayuda pretende aportar información extra a usuarios que desconozcan ciertos apartados o funcionalidades de la aplicación. Se presentan con un icono de un círculo con una letra ‘i’ en su interior y, tras pulsar en ellos, aparecerá el mensaje con la explicación:

Figura 3.8: Ventana emergente con una explicación sobre la Curva del Olvido

Otras interfaces añadidas en esta iteración y sobre las que se preguntó fueron las interfaces de configuración de notificaciones, calendario, revisión y feedback (Figuras 3.9 3.10, 3.11, 3.12 y 3.13).

En esta primera interfaz sobre la configuración de notificaciones (Figura 3.9), aparecen dos términos de importancia: revisión¹ y feedback². Con este sistema, el usuario puede valorar su día, su semana o mes, permitiéndole así considerar cambios en su organización del estudio, permitiendo a éste un mejor desempeño académico. Ambas interfaces (Figuras 3.9 y 3.10), se crearon para implementar la idea del sistema de feedback propuesto en la primera iteración (ver propuesta IT1.3).

¹Proceso que el usuario realiza de manera diaria y/o semanal que aporta información de diferente valor para la aplicación y para el propio usuario. En esta información se basará la aplicación para proporcionar feedback de utilidad al usuario.

²Estadísticas proporcionadas por la aplicación al usuario.

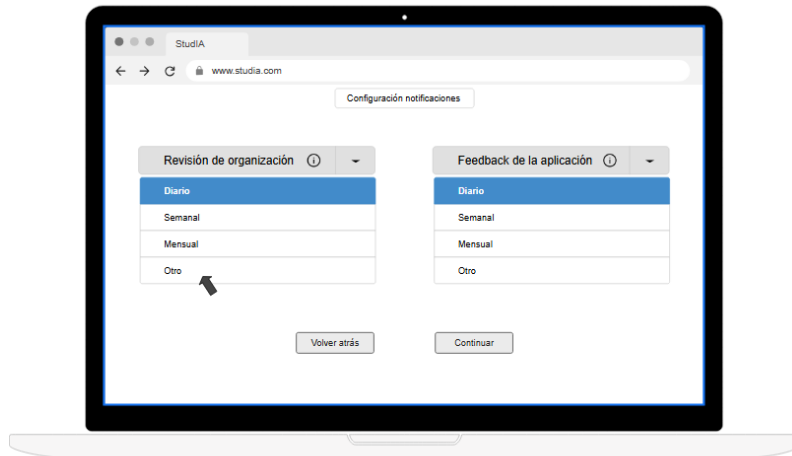


Figura 3.9: Interfaz de configuración de notificaciones

En la interfaz de la Figura 3.10 se pueden observar las opciones que tiene el usuario para poder escoger libremente los días de revisión.

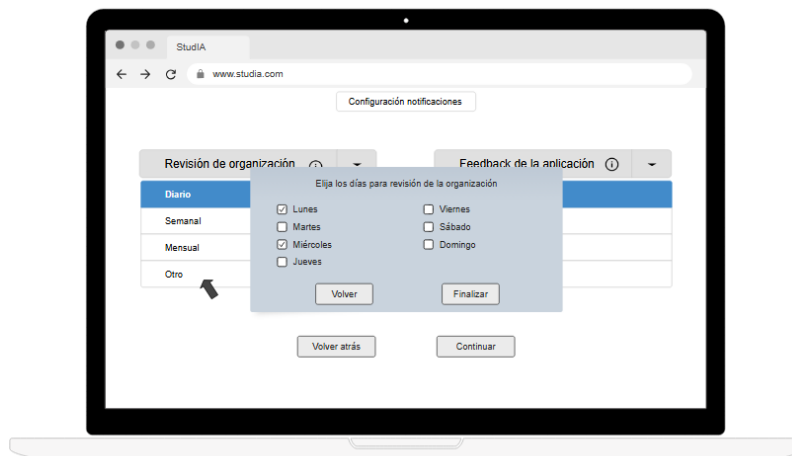


Figura 3.10: Interfaz de configuración de notificaciones con elección personalizada

La Figura 3.11 es la interfaz principal de la aplicación, que se crea a partir de las propuestas IT1.5, IT1.6 e IT1.8 de la primera iteración. Aquí se muestra el calendario con los bloques de estudio y un sistema de colores totalmente personalizable a gusto del usuario. En este caso particular, el sistema de colores representa:

- Verde: bloque completado en su totalidad con éxito
- Naranja: bloque completado de forma parcial
- Rojo: bloque no completado u omitido

A mayores, se pueden observar una ventana emergente de color azul que indica el comienzo de un bloque de estudio, el cual se puede comenzar en ese mismo momento, posponer para más tarde o cancelar.

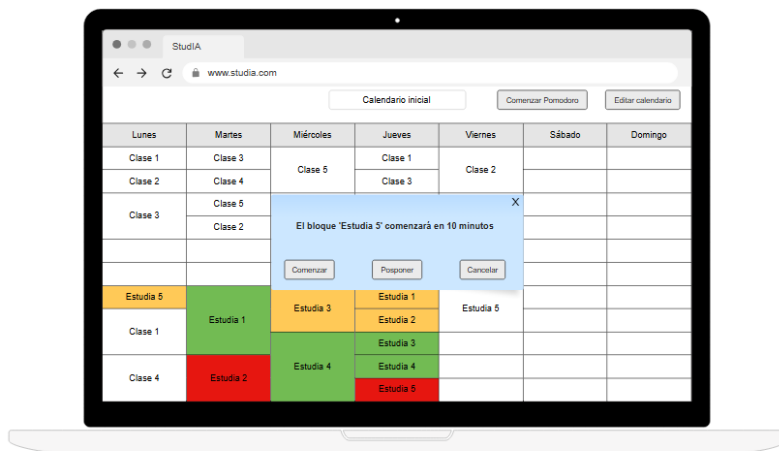


Figura 3.11: Interfaz de calendario

A continuación se presentan las interfaces de revisión y feedback (Figuras 3.12 y 3.13). Estas interfaces, en general, no gustaron los usuarios, además de ser confusas para ellos. Aquí se presenta la revisión que el usuario puede realizar (Figura 3.12):

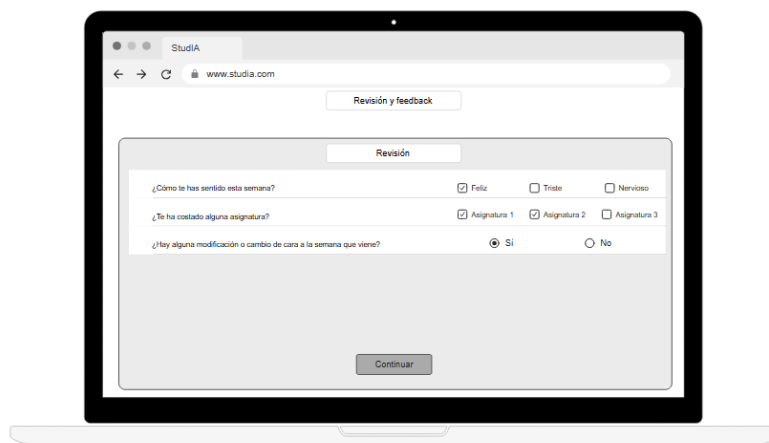


Figura 3.12: Interfaz de revisión

La Figura 3.13 muestra la interfaz que presentaría el feedback al usuario, tras la realización de las revisiones, mostrando posibles cambios y ajustes al usuario para mejorar la semana próxima.

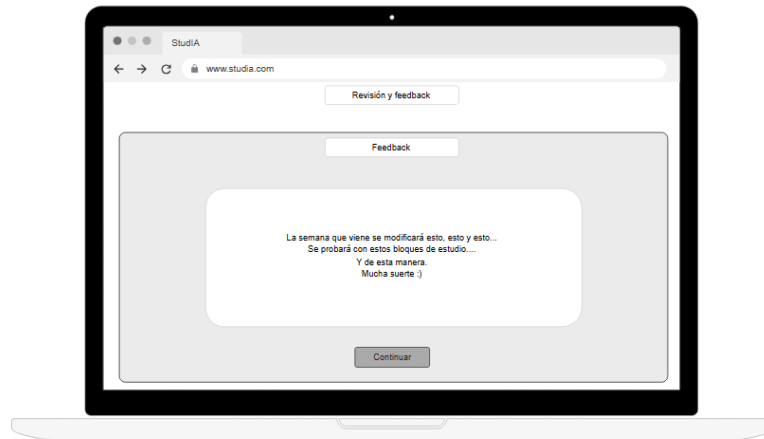


Figura 3.13: Interfaz de feedback

Tras haber recopilado las ideas y cambios de la segunda iteración de pruebas de concepto, se procedió a realizar la siguiente Tabla 3.3:

Tabla 3.3: Listado de propuestas de los usuarios de la iteración 2

Código de Propuesta	Propuesta	Descripción	Grado de alineación con el proyecto	Factibilidad
IT2.01	Método 'Curva del Olvido'	Método de estudio para mantener los conocimientos a lo largo del tiempo, repasando lo estudiado día a día	Alto	No
IT2.02	Grado de satisfacción con el trabajo realizado opcional y configurable, representado con un slider	Opción dentro de las revisiones que permita al usuario expresar su grado de satisfacción con las diferentes asignaturas y el trabajo realizado esa semana	Medio	Sí
IT2.03	Compensar sesiones incompletas o sin realizar	Opción de la aplicación para compensar sesiones incompletas o sesiones sin realizar	Medio	No
IT2.04	Estudiar fuera de los bloques planteados originalmente	Opción que permita al usuario estudiar fuera de las sesiones de estudio planteadas en el horario, añadiéndose al balance semanal de estudio	Medio	Sí
IT2.05	Mostrar asignaturas con código de colores	Opción que permita al usuario establecer colores diferentes para cada asignatura	Medio	Sí

Continúa en la siguiente página

Tabla 3.3: Listado de propuestas de los usuarios de la iteración 2 (Continuación)

IT2.06	Opción de realizar modificaciones del calendario de cara a la siguiente semana (revisión de la organización)	Opción que permita al usuario modificar el calendario de la semana siguiente, en función de las horas completadas de estudio en la semanal anterior	Medio	Sí
IT2.07	Estadísticas	Apartado de estadísticas que el usuario puede revisar para ver sus avances diarios, semanales o por asignatura	Medio	Sí
IT2.08	Gráfico representativo del avance en las asignaturas y rendimiento (feedback)	Sistema de estadísticas y gráficos en la revisión que permita al usuario visualizar sus avances semanales	Medio	Sí
IT2.09	Modo concentración	Modo dentro de la aplicación que permita al usuario concentrarse, limitando las notificaciones, mensajes y distracciones. También puede cambiar el código de colores a escala de grises	Medio	Sí
IT2.10	Ventana de estudio	Modo dentro de la aplicación al que se accede cuando comienza una sesión de estudio, siendo una interfaz diferente al calendario, menos disruptiva para el usuario	Medio	Sí
IT2.11	Varios calendarios simultáneos	Interfaz que la visualización de múltiples calendarios de forma simultánea	Bajo	No
IT2.12	Posponer bloque de estudio	Opción que permita al usuario posponer una sesión de estudio y poder elegir qué hacer con ella	Bajo	No
IT2.13	Modificar el sistema de valoración en la revisión de la organización	Modificación general del sistema de valoración en la revisión, cambiando el enfoque del mismo y las preguntas realizadas en ésta	Bajo	No
IT2.14	Pregunta sobre la carga de estudio en la revisión de la organización	Pregunta enfocada en la carga de trabajo y estudio de las asignaturas, para realizar cambios si fuera necesario	Bajo	No
IT2.15	Elección temporal en la compensación de bloques (misma semana, otra semana...)	Opción que permita al usuario elegir cuando compensar las sesiones incompletas o sin realizar en las revisiones	Bajo	No

Continúa en la siguiente página

Tabla 3.3: Listado de propuestas de los usuarios de la iteración 2 (Continuación)

IT2.16	Personalización de descansos	Opción que permita al usuario personalizar el tiempo de descanso entre sesiones	Bajo	No
IT2.17	Acceso al calendario antes de finalizar el formulario para poder realizar ajustes	Permitir al usuario acceder al calendario antes de finalizar el formulario inicial para poder realizar ajustes de sesiones y de horarios	Bajo	No
IT2.18	Frases motivacionales en el sistema de feedback	Aparición de frases motivacionales y de ayuda aleatorias al usuario como parte del feedback de la aplicación	Bajo	No
IT2.19	Introducción de más notificaciones motivadoras durante los bloques de estudio (ánimo queda poco o similar)	Notificaciones durante las sesiones de estudio que motiven al usuario a completarlas	Bajo	No
IT2.20	Apartado de excepciones dentro del formulario inicial por motivos externos (cita médica, reunión, trabajo...)	Apartado dentro del formulario inicial donde añadir zonas horarias donde no se pueden poner bloques de estudio por motivos externos como reuniones, trabajo, laboratorios...	Bajo	No
IT2.21	Franja horaria sin notificaciones	Opción que permita al usuario establecer una franja horaria donde la aplicación no proporcione notificaciones	Bajo	No
IT2.22	Opción de respuesta libre a modo de agenda personal en la revisión de la organización	Opción que permita al usuario expresarse libremente a modo de agenda o diario en la revisión	Bajo	No

Los resultados obtenidos en la Iteración 2 fueron muy útiles y permitieron refinar y enfocar aún más la aplicación, profundizando en sus componentes, interfaces y funcionalidades, modificando con ésto el flujo de la aplicación.

De esta iteración surgieron muchas ideas de diferente tipo y dificultad, que proporcionan una visión más cercana a lo que sería la aplicación final y que, como se puede observar en el apartado de factibilidad de la Tabla 3.3, serían después implementadas. Todas estas ideas se describen en detalle en la tabla de propuestas previamente mencionada.

El nuevo planteamiento del formulario (Figura 3.7) fue aceptado por todos los usuarios participantes en esta iteración y su diseño ya es prácticamente el definitivo, realizando cambios menores que afectan, mayormente, al flujo de la aplicación.

Las interfaces pertenecientes a las Figuras 3.9, 3.10, 3.12 y 3.13 son las que más cambios sufrirán en iteraciones posteriores. Los usuarios aclararon en las entrevistas que estas interfaces eran confusas y poco claras, por lo que fueron modificadas en su totalidad.

A mayores, algunas propuestas fueron fusionadas, como las propuestas IT2.14 e IT2.15 y otras implementadas de forma conjunta (propuesta IT2.17 junto con el código de colores de la Figura 3.11).

3.1.4. Iteración 3

La última iteración se realizó para refinar y pulir detalles, algunos apartados técnicos y visuales de cara a la aplicación final. En esta iteración se realizó, una prueba de concepto online utilizando de nuevo un formulario creado con Google Forms para guiar las sesiones con cada participante, puesto que resultó ser muy efectivo en la iteración anterior y la recogida de datos más sencilla.

En esta última iteración se puso el foco de atención en el calendario, modo estudio y revisiones, permitiendo a los entrevistados expresarse libremente para cada una de las interfaces, que se mostrarán a continuación (ver Figuras 3.14 a 3.21).

En relación al formulario inicial (ver Figura 3.14), se añadió un flujo extra para modificar calendario, teniendo en cuenta la propuesta IT2.16 de la segunda iteración.

The screenshot shows a web browser window with the URL 'www.studia.com'. The page has a header with 'Formulario inicial' and 'Finalizar formulario' buttons. The main content area contains the following questions and options:

- Horario de clases: Insertar Google Calendar >, Crear calendario de cero >
- ¿Cuántas horas a la semana quieres enfocarte en el estudio? Respuesta: [input field]
- ¿Tamaño de las sesiones de estudio? 1h, 2h, 3h, Otro (radio buttons)
- Organización de las sesiones de estudio: Por día (dropdown menu)
- ¿En qué momento del día eres más productivo? Mañana, Tarde, Noche (checkboxes)
- ¿Quieres enfocar el estudio basado en la Curva del Olvido? Si, No (radio buttons)
- ¿Quieres complementar los bloques con el método Pomodoro? Si, No (radio buttons)

Figura 3.14: Formulario inicial

Para la configuración de las notificaciones de la aplicación, se creó esta interfaz (ver Figura 3.15) teniendo en cuenta las propuestas IT1.7 e IT2.18 de la primera y segunda iteración.

The screenshot shows a web browser window with the URL 'www.studia.com'. The page has a header with 'Configuración notificaciones'. The main content area contains the following notification types and options:

- Notificaciones revisión diaria: Si, No (radio buttons)
- Notificaciones revisión y feedback semanal: Si, No (radio buttons)
- Notificaciones de comienzo de bloques de estudio: Si, No (radio buttons)
- Notificaciones sobre compensación de bloques no completados: Si, No (radio buttons)
- Notificaciones durante periodos de estudio: Si, No (radio buttons)
- Notificaciones estilo VII (descansos, tiempos de estudio...): Si, No (radio buttons)

Figura 3.15: Configuración de notificaciones

La interfaz del calendario en modo edición se creó a partir de la propuesta IT2.16 (ver Tabla 3.3), con una propuesta de estudio de las diferentes asignaturas introducidas en el formulario inicial:

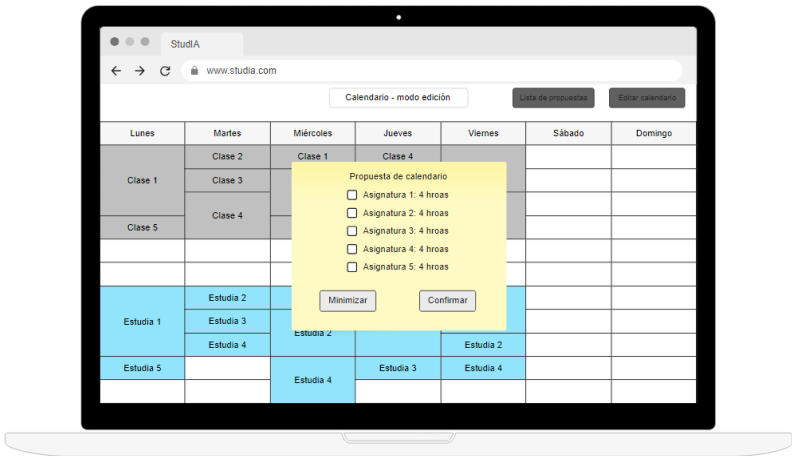


Figura 3.16: Interfaz de calendario en modo edición y ventana emergente con propuesta de calendario

La interfaz de calendario en uso normal no recibió cambios significativos, añadiendo un botón para el modo estudio de la propuestas IT2.15 (ver Tabla 3.3) , con notificaciones en la parte inferior derecha:

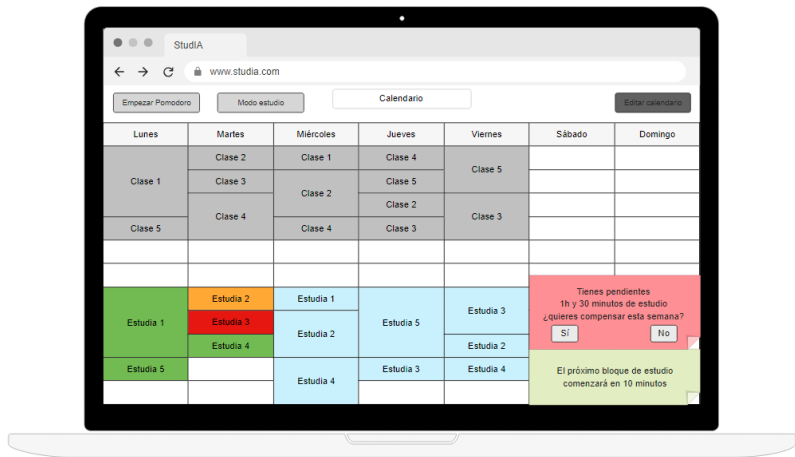


Figura 3.17: Interfaz de calendario con notificaciones

La interfaz de modo estudio de la aplicación se creó a partir de la propuestas IT2.15 (ver Tabla 3.3). Esta interfaz pretende que el usuario sea más productivo y tener menos distracciones. Además, cuenta con un temporizador correspondiente al bloque de estudio en el que se encuentra en ese momento:

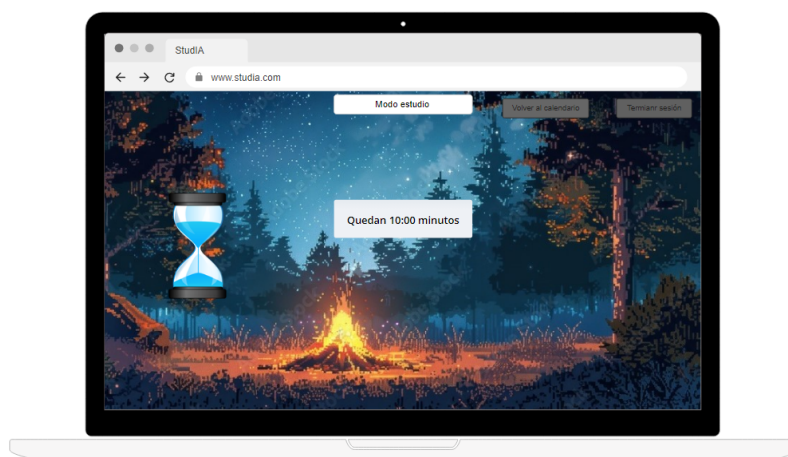


Figura 3.18: Interfaz de modo estudio

La interfaz de revisión diaria se modificó respecto a la Figura 3.12 debido a los comentarios recibidos en la segunda iteración. En este rediseño, el usuario puede elegir si compensar o no las horas pendientes, y en qué día de la semana hacerlo.

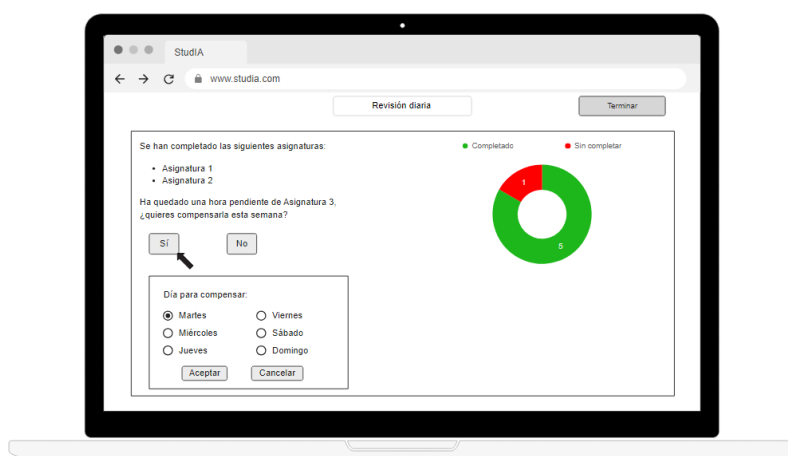


Figura 3.19: Interfaz de revisión diaria

La interfaz de la revisión semanal se creó a partir de la interfaz de feedback de la iteración anterior (Ver Figura 3.13) debido a los comentarios recibidos. Esta interfaz, que presenta un balance positivo (en este momento, se planteaba que el balance era positivo cuando la cantidad de horas pendientes fuera inferior al 50 % de las horas semanales de estudio), permite al usuario ver el progreso semanal y recuperar las horas pendientes en la semana próxima:

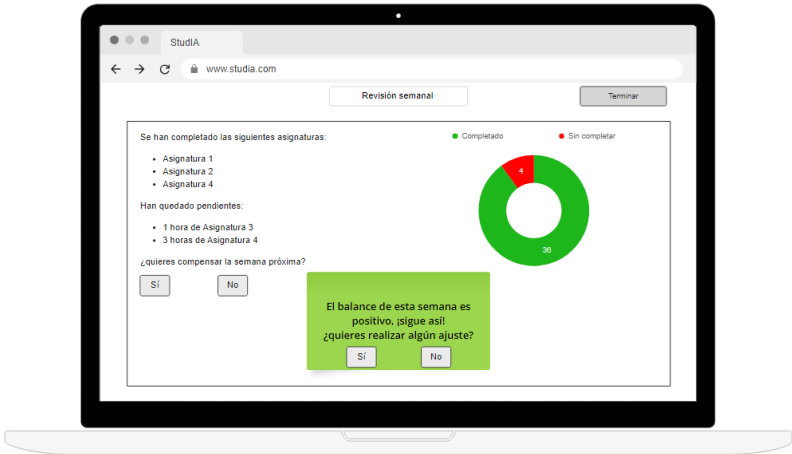


Figura 3.20: Interfaz de revisión semanal positiva

Esta interfaz es idéntica a la anterior (ver Figura 3.20), pero presentado un balance negativo, donde se plantea al usuario la opción de modificar la planificación para la semana próxima y poder recuperar así las horas no realizadas:

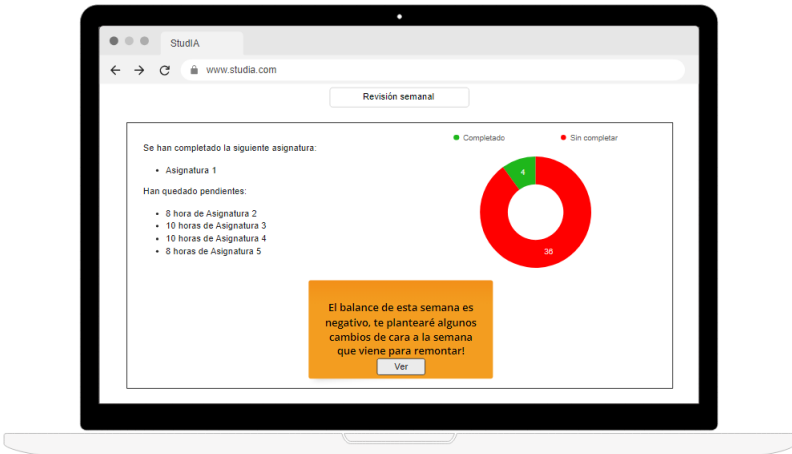


Figura 3.21: Interfaz de revisión semanal negativa

Tabla 3.4: Listado de propuestas de la iteración 3

Código de Propuesta	Propuesta	Descripción	Grado de alineación con el proyecto	Factibilidad
IT3.1	Importación de calendarios a través de ficheros ics	Permitir a los usuarios importar calendarios de otras aplicaciones mediante el formato ics	Alto	Sí

Continúa en la siguiente página

Tabla 3.4: Listado de propuestas de la iteración 3 (Continuación)

IT3.2	Revisión diaria y semanal a elección del usuario	Permitir al usuario elegir si quiere o no realizar las revisiones	Medio	Sí
IT3.3	Apartado de información estilo ventana emergente de cada bloque de estudio con información, tareas o exámenes	Apartado dentro de cada bloque de estudio donde el usuario pueda escribir información que éste necesite	Medio	Sí
IT3.4	Poder añadir una marca (estrella o similar) a tareas de mayor importancia	Opción que permita marcar determinados bloques de estudio o eventos de forma destacada respecto al resto	Medio	No
IT3.5	Plantilla de estudio base en el formulario (plantilla con bloques predefinidos respecto a una carrera, grado o tipo de estudio)	Opción en el formulario inicial que permita establecer una plantilla de estudio por horas a la semana	Bajo	No
IT3.6	Botón para activar y desactivar todas las notificaciones a la vez	Botón que permita activar y desactivar todos los tipos de notificaciones en esta sección de la configuración	Bajo	No
IT3.7	Vistas del calendario para ver días sin actividades o no verlos	Vista que permita quitar días en los que no haya ningún tipo de sesión de estudio	Bajo	No

Los resultados de la Iteración 3 fueron muy positivos. La gran mayoría de participantes aceptaron los cambios respecto a las iteraciones anteriores y dieron una valoración positiva sobre la misma. A mayores, se plantearon algunas ideas con funcionalidad añadida que se presentan en la Tabla 3.4.

Tras esta iteración y una vez analizados los resultados, se procede a realizar una serie de ajustes en los prototipos y crear algunos extra para facilitar las labores de diseño y desarrollo.

3.1.5. Cambios finales tras las iteraciones

En esta sección se mostrarán los cambios finales sobre los bocetos de la última iteración, teniendo en cuenta las opiniones proporcionadas por los participantes durante las tres iteraciones. Aparte de las opiniones y respuestas de los participantes, en esta última fase, se tuvieron en cuenta cuestiones de análisis y diseño que motivaron algunos de los cambios y nuevos prototipos presentados en esta subsección. Un ejemplo de ello se muestra en los prototipos de las Figuras 3.22, 3.23 y 3.24, donde se separaron las diferentes partes del formulario para facilitar al usuario el entendimiento de las mismas y no sobrecargar una sola página.

Cabe destacar que la aplicación final tendrá más interfaces y se verá de mejor manera que los bocetos presentados a continuación, puesto que la aplicación utilizada para realizarlos (Moqups) era limitada, permitiendo hasta solo hasta 400 objetos (unas 10 interfaces aproximadamente).

Un cambio de importancia realizado tras la evaluación de la información recogida en las iteraciones ha sido la interfaz del formulario inicial. Ahora presenta una nueva forma de añadir asignaturas para la organización, con un

botón propio y nuevas interfaces extra (3.22, 3.23 y 3.24). Esta separación es de importancia puesto que permite diferenciar claramente los apartados del formulario, facilitando así la contextualización de la parte que se está haciendo, sin sobrecargar una única interfaz con todo.

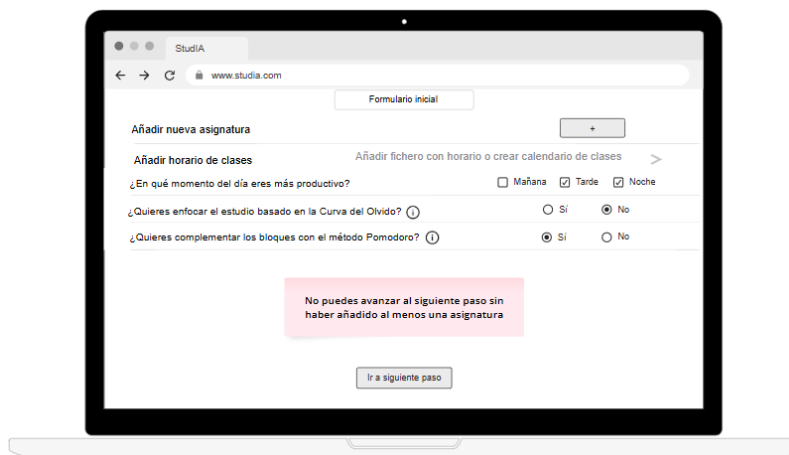


Figura 3.22: Interfaz del formulario inicial

Una nueva interfaz asociada al formulario inicial es la de crear una asignatura nueva (ver Figura 3.23). Se optó por separar el formulario para hacerlo más sencillo y no tener una interfaz muy recargada de preguntas. Aquí se puede especificar la cantidad de horas dedicadas a la semana, el tamaño de los bloques de estudio y la organización de estas sesiones o bloques.

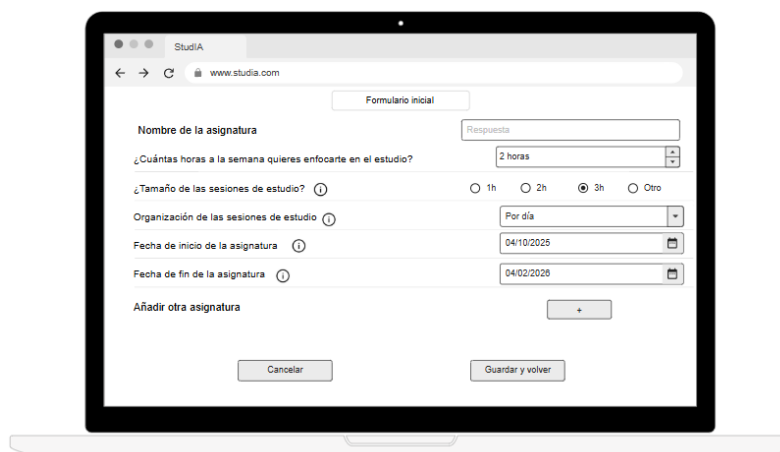


Figura 3.23: Interfaz para añadir una asignatura

Otra interfaz creada, es la de añadir el horario a través de un fichero (ver Figura 3.24). En esta interfaz, se puede optar por dos tipos de horario:

- Horario con eventos de una asignatura: horario que proporciona los eventos de una asignatura y los asocia a una de las ya creadas.
- Horario de clase: horario con horas de clase, laboratorios y otras actividades donde no establecer bloques de estudio.

Cabe destacar que, para el correcto funcionamiento del calendario, se debe integrar un fichero con formato ICS. Este formato pertenece al estándar iCalendar (Estándar [5]) para el intercambio de información de calendarios.

Este estándar es utilizado por grandes aplicaciones como Microsoft Outlook y Google Calendar, permitiendo la importación y exportación de calendarios de forma sencilla, además de facilitar la integración entre diferentes aplicaciones de este tipo.

Además, a parte de la integración con otras aplicaciones de calendario, todas las plataformas que implementan como arquitectura base el Learning Management System (LMS) Moodle, permiten la exportación de calendarios de cursos, eventos y asignaturas con ficheros en el formato previamente mencionado, facilitando al alumno poder integrar los eventos de una asignatura en la aplicación Studlendar.

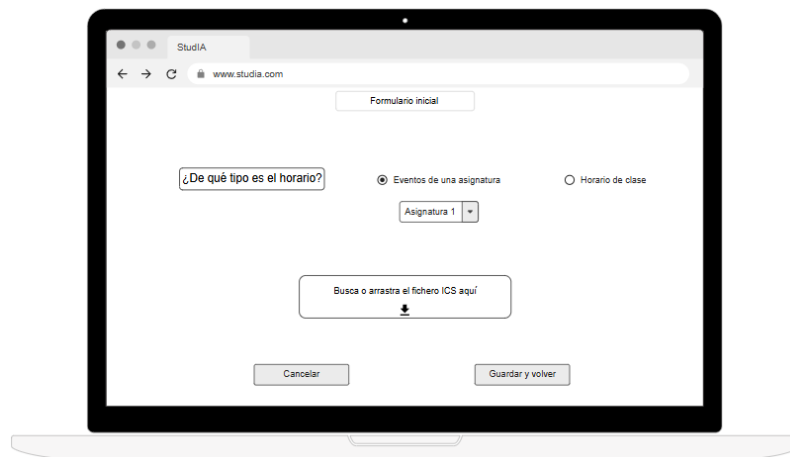


Figura 3.24: Interfaz para añadir horario a través de un fichero

Interfaces como la de configuración de notificaciones (Figura 3.25) recibieron cambios menores, como la adición de algunos botones extra para facilitar a los usuarios tareas como habilitar y deshabilitar todas las opciones:

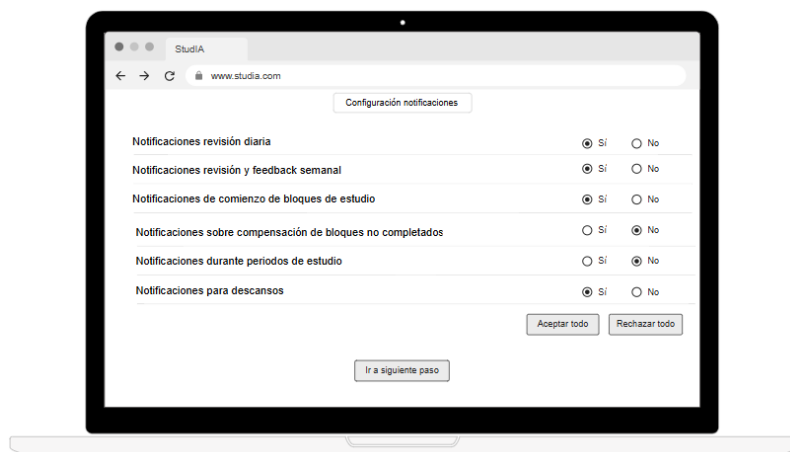


Figura 3.25: Interfaz de configuración de notificaciones

En la interfaz de la Figura 3.26 podemos observar la propuesta de bloques de estudio de las diferentes asignaturas añadidas durante el formulario inicial. En este apartado podremos escoger las propuestas de nuestra preferencia.

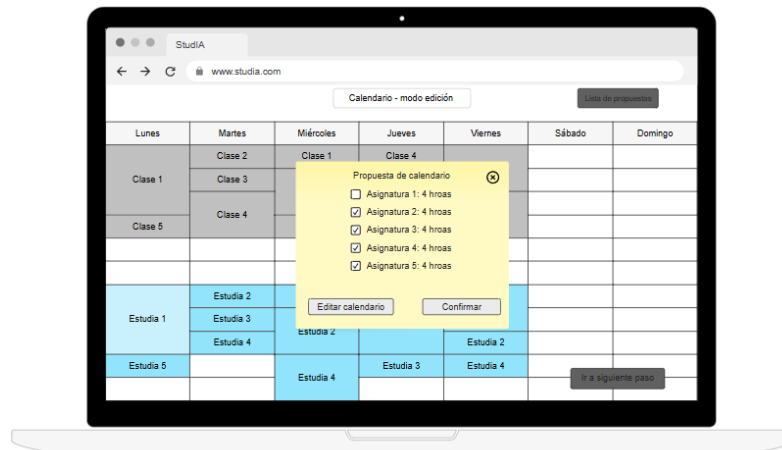


Figura 3.26: Interfaz de calendario en modo edición y con propuesta

La interfaz de calendario (Figura 3.27) presenta nuevos botones para empezar un pomodoro, el modo estudio, configuración y edición del calendario.

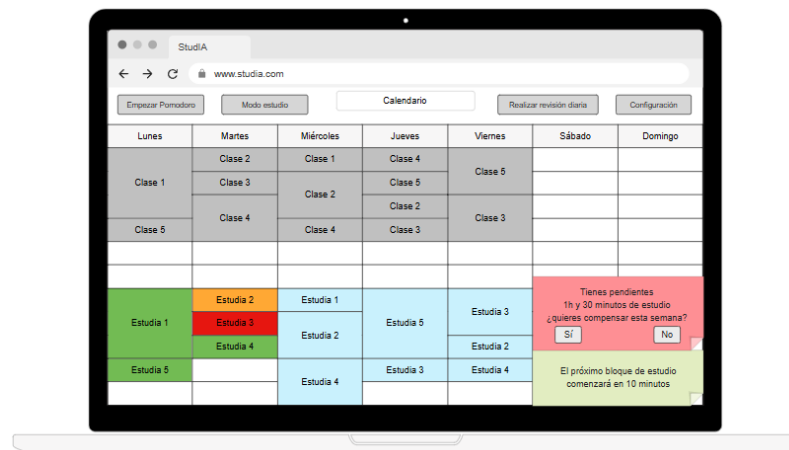


Figura 3.27: Interfaz de calendario

El modo estudio (Figura 3.28) también presenta nuevos botones, para pausar la sesión, terminarla o poder visualizar el calendario sin parar la sesión:

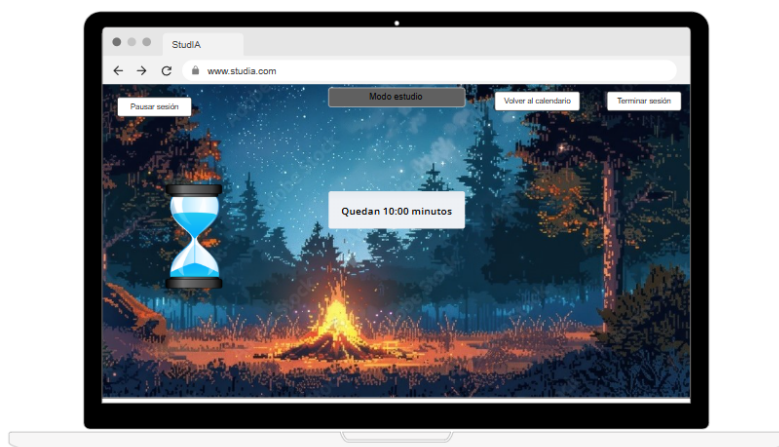


Figura 3.28: Interfaz de modo estudio

La interfaz de revisión diaria (Figura 3.29) se ha simplificado, permitiendo al usuario situar libremente la sesión o sesiones pendientes en el calendario, sin que se tenga que hacer en esta interfaz:

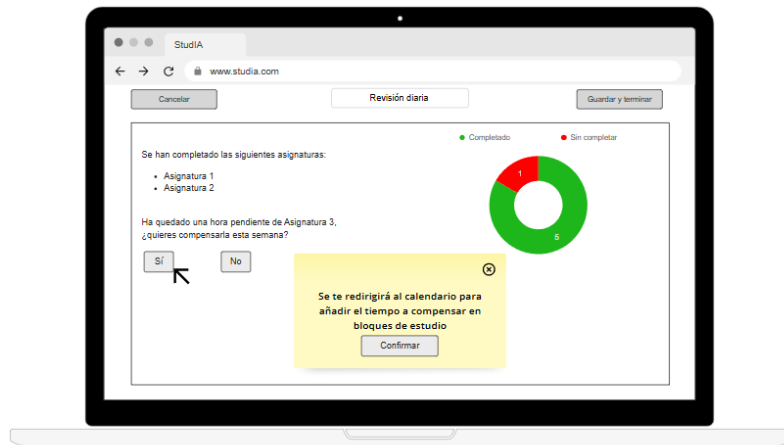


Figura 3.29: Interfaz de revisión diaria

La interfaz de revisión semanal (Figura 3.30) ha cambiado en gran medida, habiendo eliminado la valoración positiva o negativa, puesto que a los participantes no les gustó esa idea. Por otro lado, se estableció otra métrica de medición, "satisfacción con el trabajo", lo que permite al usuario valorar de forma independiente el trabajo realizado en cada asignatura durante la semana.

A mayores, a parte de establecer estadísticas sobre las horas completadas y no completadas, se añaden posibles horas extra que se hayan realizado de una o varias asignaturas.

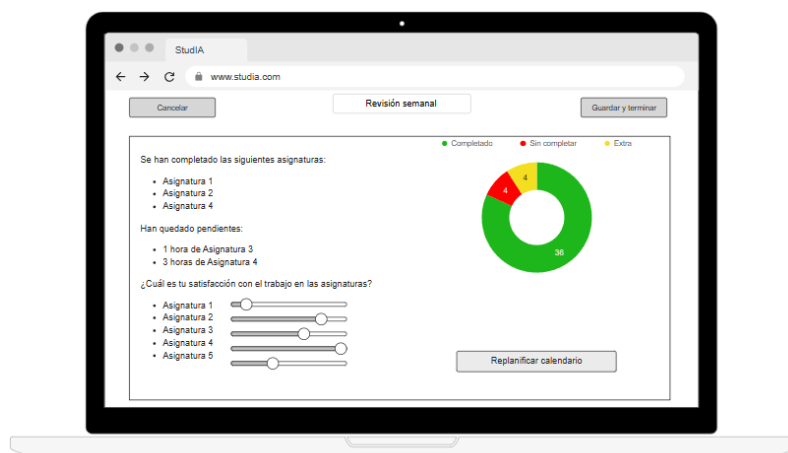


Figura 3.30: Interfaz de revisión semanal

3.2. Especificación de requisitos

3.2.1. Obtención de requisitos

La obtención de requisitos de la aplicación se ha realizado progresivamente teniendo en cuenta las tablas de propuestas y los prototipos creados para cada iteración de análisis. Todo esto permitió encontrar y depurar los diferentes requisitos de la aplicación hasta los listados que se adjuntan a continuación.

3.2.2. Requisitos funcionales

- **RF-01:** El sistema debe permitir al usuario configurar el calendario indicando sus preferencias de estudio.
- **RF-02:** El sistema debe permitir al usuario crear una asignatura con parámetros como el nombre, las horas de estudio, el tamaño de las sesiones de estudio, la fecha de inicio y la fecha de fin.
- **RF-03:** El sistema debe generar una serie de propuestas de estudio por asignatura, conforme a las preferencias de estudio y los parámetros de esa asignatura.
- **RF-04:** El sistema debe permitir al usuario poder importar eventos y bloques de clase en formato ICS para cada asignatura creada.
- **RF-05:** El sistema debe permitir al usuario seleccionar las propuestas de estudio que quiere configurar.
- **RF-06:** El sistema debe permitir al usuario crear ítems en el calendario.
- **RF-07:** El sistema debe permitir al usuario editar ítems del calendario.
- **RF-08:** El sistema debe permitir al usuario borrar ítems del calendario.
- **RF-09:** El sistema debe proporcionar estadísticas al usuario sobre su progreso.

3.2.3. Requisitos funcionales de información

El sistema deberá almacenar la siguiente información:

- **RFI-01:** Preferencias del usuario para la creación de propuestas de estudio.

- **RFI-02:** Eventos y bloque de clase importados con formato ICS.
- **RFI-03:** Bloques de estudio seleccionados por el usuario y propuestos por la aplicación.
- **RFI-04:** Bloques de clase, bloques de estudio y eventos creados por el usuario.
- **RFI-05:** Ediciones que el usuario realice sobre el calendario y los ítems creados (bloques de clase, estudio y eventos).
- **RFI-06:** Grado de satisfacción semanal con las asignaturas.
- **RFI-07:** Estadísticas del usuario (ratio de completitud de cada bloque de estudio y completitud de cada evento).

3.2.4. Requisitos no funcionales

- **RNF-01:** La aplicación debe ser compatible con las versiones más recientes de los navegadores más utilizados en la actualidad.
- **RNF-02:** La interfaz de la aplicación debe ser compatible con diferentes tamaños de pantalla.
- **RNF-03:** La aplicación debe cumplir con la normativa vigente de accesibilidad web, establecida por la W3C, teniendo en cuenta el alcance del proyecto.

3.3. Diagrama de casos de uso

Los casos de uso son un conjunto de escenarios relacionados por el resultado que el usuario espera obtener y describen cómo funciona el sistema, facilitando su comprensión ([3]). La realización de casos de uso tiene como objetivo principal identificar y definir requisitos del sistema, y sirven como base para el diseño del sistema y creación de escenarios de prueba.

En las Figuras 3.31 y 3.32 se muestran los diferentes casos de uso de la aplicación y se pueden dividir en diferentes temáticas:

1. Creación y actualización de cuentas:

- Iniciar sesión en la aplicación.
- Darse de alta en la aplicación.
- Cerrar sesión.
- Recuperar contraseña.

2. Creación y modificación del calendario:

- Configurar calendario
- Importar calendario ICS

3. Creación y edición de bloques y eventos:

- Crear bloque de estudio
- Editar bloque de estudio
- Crear bloque de clase
- Editar bloque de clase

- Crear evento
- Editar evento

4. Revisión y estadísticas:

- Revisión del progreso diario
- Revisión del progreso semanal
- Revisión del progreso por asignatura

5. Configuración de la herramienta:

- Configurar notificaciones
- Crear asignatura

6. Otras funcionalidades:

- Comenzar pomodoro
- Apuntar horas de estudio

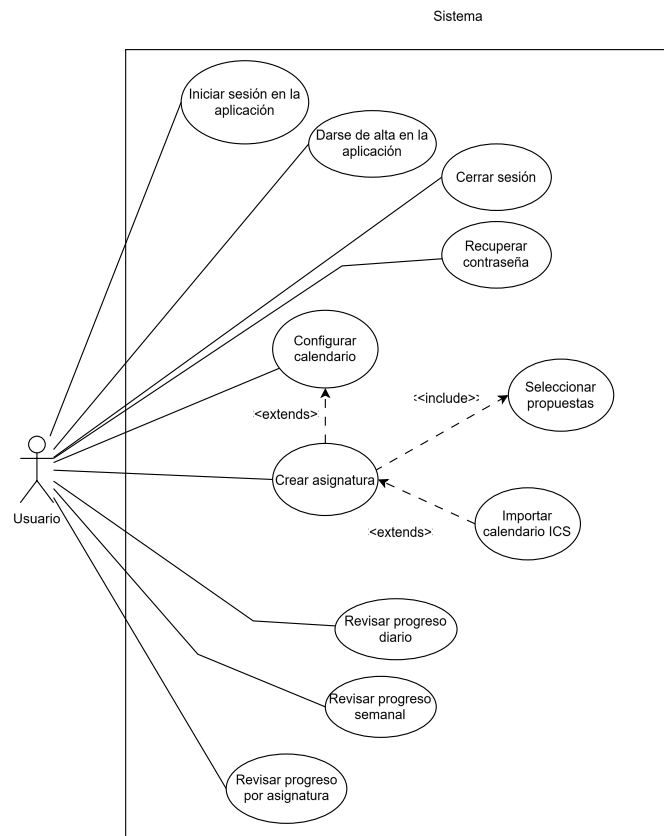


Figura 3.31: Diagrama de casos de uso, parte 1

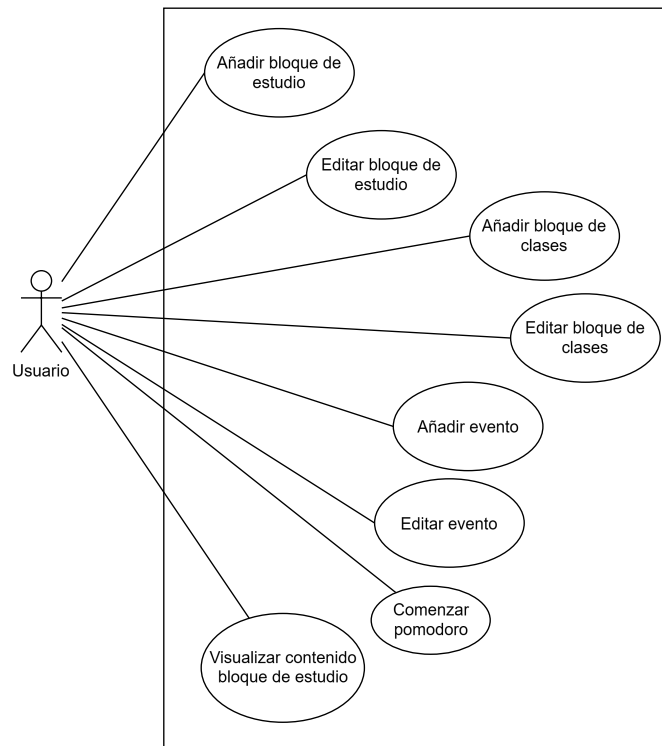


Figura 3.32: Diagrama de casos de uso, parte 2

3.4. Descripción de casos de uso

Los casos de uso, presentados en el diagrama anterior, se explicarán de forma textual, junto con su diagrama de secuencia, a continuación. Estos diagramas de secuencia corresponden con los que se realizarían en una fase de análisis y contienen tres clases principales:

- **Clases Boundary:** clases mediadoras entre el sistema y su entorno.
- **Clases Control:** clases que encapsulan el comportamiento descrito en los casos de uso.
- **Clases Entity:** clases que modelan la estructura lógica del sistema. Pueden ser persistentes o no.

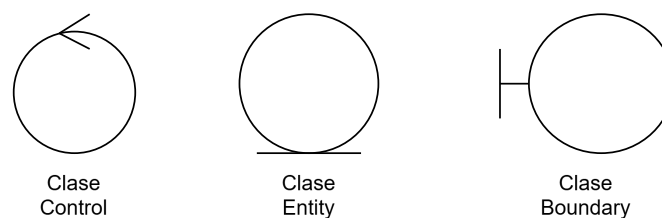


Figura 3.33: Iconos de las clases UML utilizados en diagramas de secuencia.

3.4.1. Darse de alta en la aplicación

El usuario quiere darse de alta. Accede por primera vez a la aplicación. La aplicación muestra un formulario de registro. El usuario completa el formulario de registro, la aplicación comprueba nombre, contraseña y correo electrónico y se guarda.

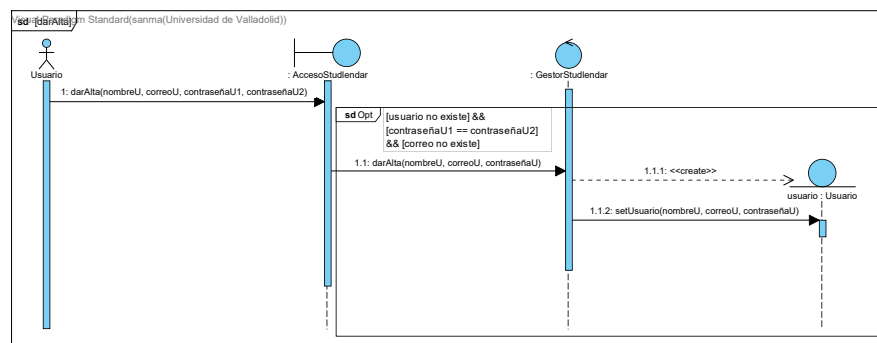


Figura 3.34: Diagrama de secuencia del caso de uso Dar Alta

3.4.2. Iniciar sesión en la aplicación

El usuario quiere iniciar sesión. Accede a la aplicación. La aplicación pide que se identifique, el usuario introduce su usuario y contraseña, la aplicación verifica las credenciales y el usuario accede a la aplicación.

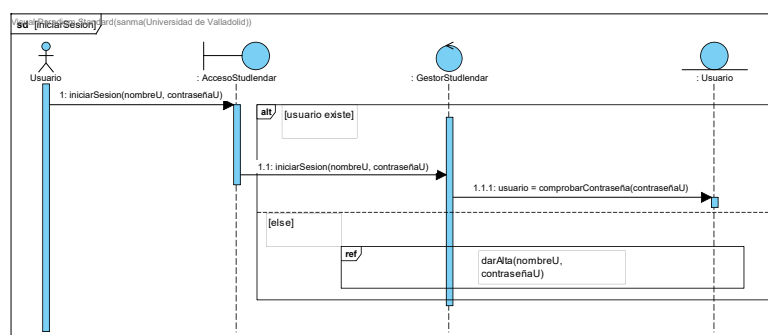


Figura 3.35: Diagrama de secuencia del caso de uso Iniciar sesión

3.4.3. Configurar calendario

El usuario quiere configurar su calendario por primera vez. Inicia sesión en la aplicación. La aplicación le muestra el formulario inicial que éste debe realizar. El usuario rellena el formulario con sus preferencias de estudio y la aplicación guarda los cambios.

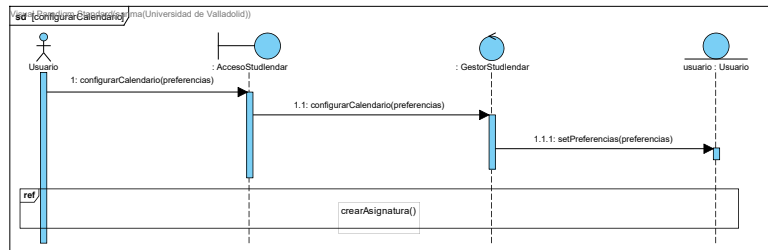


Figura 3.36: Diagrama de secuencia del caso de uso Configurar Calendario

3.4.4. Crear asignatura

El usuario quiere crear una nueva asignatura. Accede a la sección para ello y rellena el formulario. La aplicación verifica los campos y se guardan los cambios.

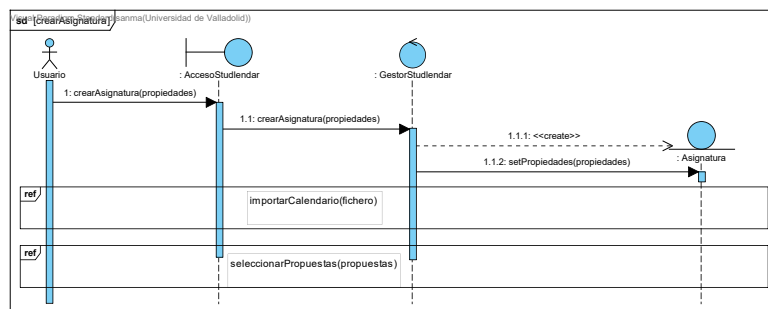


Figura 3.37: Diagrama de secuencia del caso de uso Crear asignatura

3.4.5. Importar eventos y bloques de clase

El usuario quiere importar un calendario de eventos y horarios de clase tras crear una asignatura. Accede a la sección para importar eventos y bloques. Escoge el fichero en formato ICS, la aplicación lee el fichero, el usuario selecciona los eventos y bloques que quiere y se guardan los cambios.

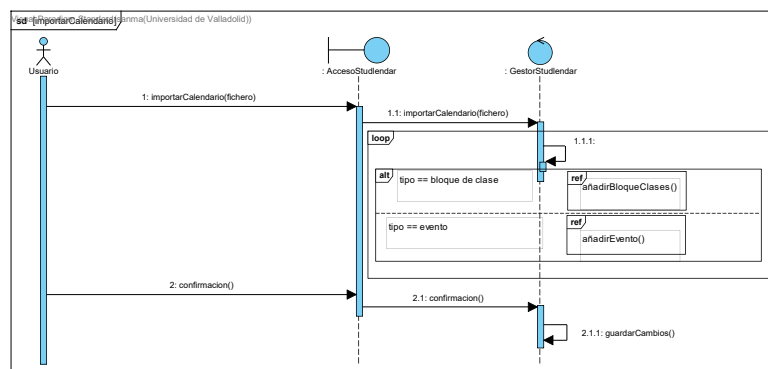


Figura 3.38: Diagrama de secuencia del caso de uso Importar calendario

3.4.6. Añadir bloque de estudio

El usuario quiere añadir un bloque correspondiente a estudio de una asignatura. Selecciona la opción para añadir un bloque, añade los atributos correspondientes (asignatura, nombre, duración, fecha y notas), la aplicación verifica lo datos y se guarda.

Para el caso de uso ‘**Añadir bloque de clase**’, la secuencia es igual a la mostrada en la Figura 3.39.

Para el caso de uso ‘**Añadir evento**’, la secuencia es igual a la mostrada en la Figura 3.39, con el matiz de que los atributos son asignatura, nombre, fecha y notas.

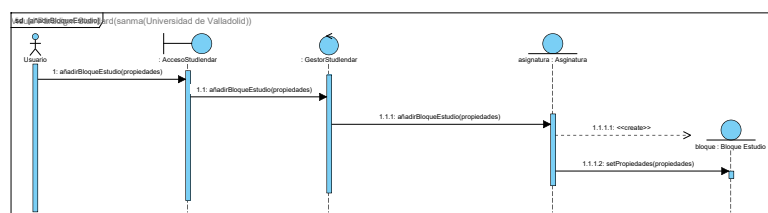


Figura 3.39: Diagrama de secuencia del caso de uso Añadir bloque de estudio

3.4.7. Editar bloque de estudio

El usuario quiere editar un bloque de estudio. Pulsa el bloque quiere editar, modifica los atributos que se pueden editar (nombre y valor ‘completado’), la aplicación verifica los cambios y se guardan.

Para los casos de uso de ‘**Editar de bloque de clase**’ y ‘**Editar evento**’, la secuencia es igual a la mostrada en la Figura 3.40.

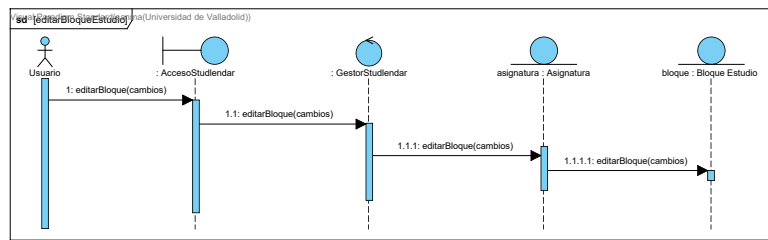


Figura 3.40: Diagrama de secuencia del caso de uso Editar bloque de estudio

3.4.8. Borrar bloque de estudio

El usuario quiere borrar un bloque de estudio. Pulsa el bloque que quiere borrar, pulsa el botón de borrar y el bloque de estudio se borra.

Para los casos de uso de ‘**Editar de bloque de clase**’ y ‘**Editar evento**’, la secuencia es igual a la mostrada en la Figura 3.41.

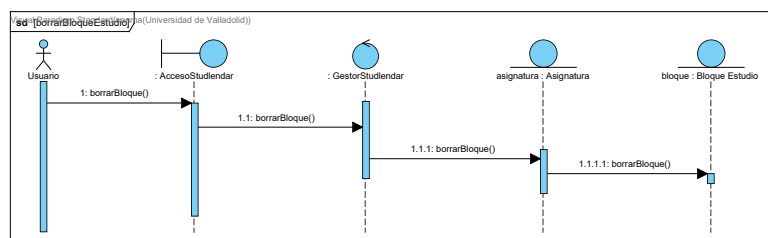


Figura 3.41: Diagrama de secuencia del caso de uso Borrar bloque de estudio

3.4.9. Visualizar elemento del calendario

El usuario quiere ver el contenido de un bloque de estudio. Accede a uno de los bloques pertenecientes y el sistema muestra su contenido.

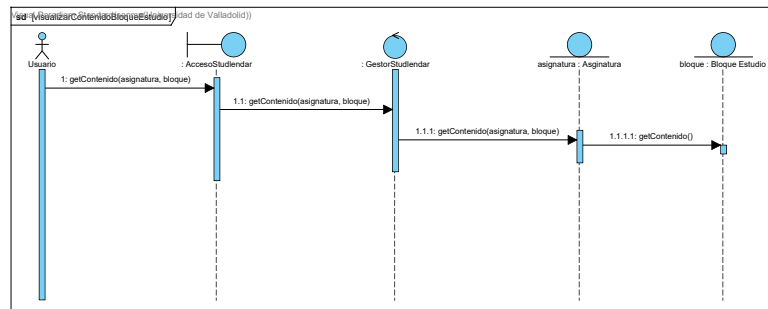


Figura 3.42: Diagrama de secuencia del caso de uso Visualizar contenido bloque de estudio

3.4.10. Revisar progreso diario

El usuario quiere realizar su progreso diario. Accede a la sección de progreso diario, y la aplicación muestra las estadísticas diarias.

Para el caso de uso 'Revisar progreso por asignatura', la secuencia es igual a la mostrada en la Figura 3.43, con el matiz de que se muestran las estadísticas mensuales por asignatura.

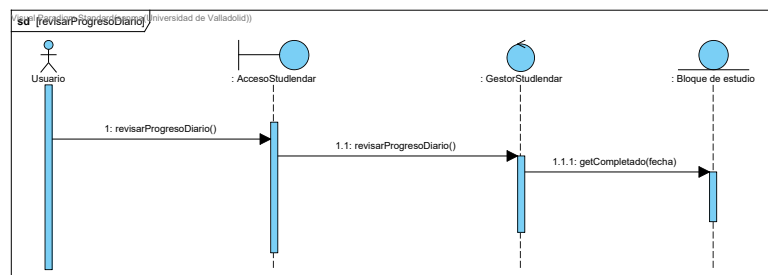


Figura 3.43: Diagrama de secuencia del caso de uso Revisar progreso diario

3.4.11. Revisar progreso semanal

El usuario quiere revisar su progreso semanal. El usuario accede a la sección de progreso semanal y la aplicación muestra las estadísticas semanales. El usuario reporta su satisfacción con cada una de las asignatura y se guardan los cambios.

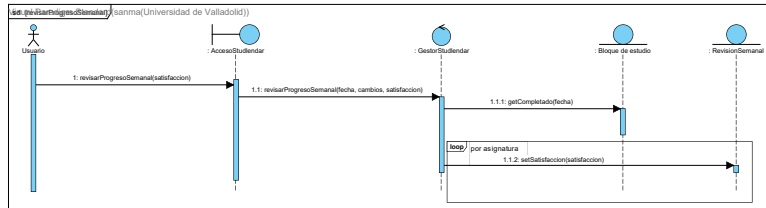


Figura 3.44: Diagrama de secuencia del caso de uso Revisar progreso semanal

3.4.12. Comenzar pomodoro

El usuario quiere empezar un bloque de estudio. Pasa a modo estudio, selecciona la duración del pomodoro, de la pausa corta y la pausa larga que quiere, y comienza la cuenta atrás.

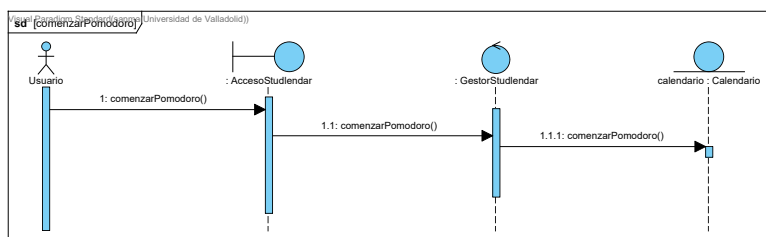


Figura 3.45: Diagrama de secuencia del caso de uso Comenzar pomodoro

3.5. Modelo de dominio

El modelo de dominio que se muestra a continuación describe el sistema de gestión de la aplicación Studlendar, que abarca las asignaturas, bloques de estudio, bloques de clase, eventos de las asignaturas y revisiones semanales.

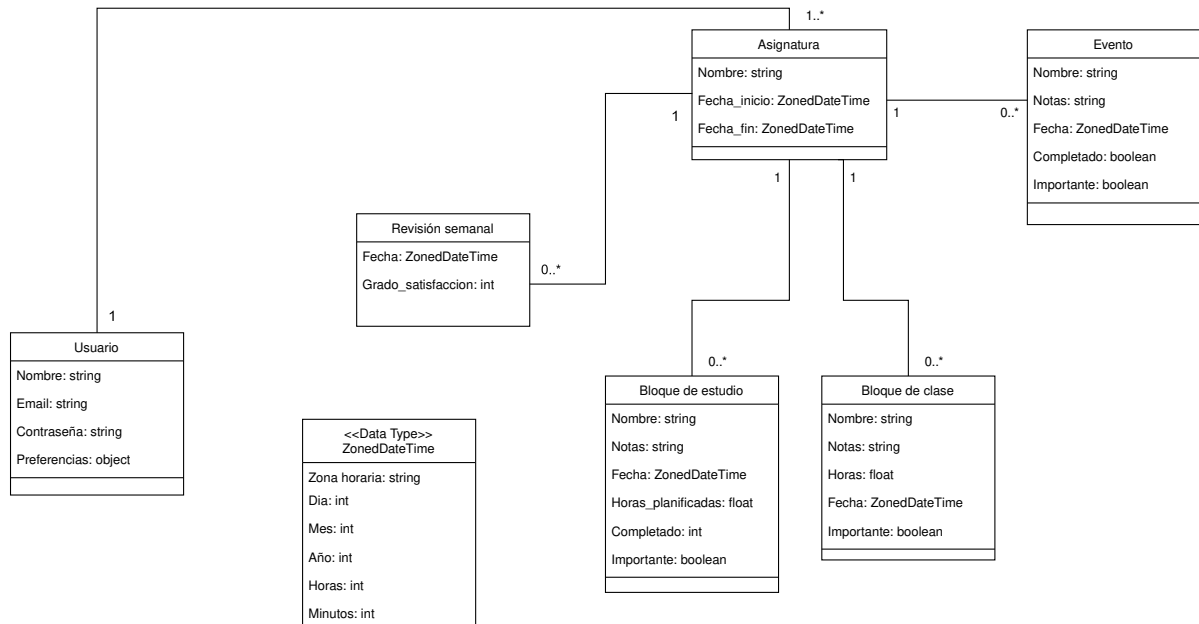


Figura 3.46: Diagrama del modelo de dominio

Capítulo 4

Fase de diseño

4.1. Introducción

Este capítulo se centra en el diseño software de la aplicación, entrando en detalle de la arquitectura de la misma, un diagrama de transiciones de interfaces, el modelo de datos y el diagrama del modelo conceptual de la base de datos. El diseño de la interfaz se ha ido realizando de forma iterativa mediante los prototipos, y junto con las necesidades descritas en el capítulo anterior.

4.2. Arquitectura del sistema

El patrón arquitectónico utilizado para este proyecto es el patrón MVC (Modelo-Vista-Controlador). Este patrón divide una aplicación en tres partes con responsabilidades diferentes, tal y como se observa en la Figura 4.1:

- **Modelo:** parte que contiene los datos y funcionalidad de la aplicación.
- **Vista:** parte que muestra la información al usuario.
- **Controlador:** parte que maneja las interacciones y entradas del usuario.

Estas partes con diferentes responsabilidades se comunican entre sí, para proporcionarse los diferentes datos e información necesarias para el correcto funcionamiento de la aplicación.

Actualmente, es uno de los patrones arquitectónicos más utilizado en el desarrollo de aplicaciones web ([21]), puesto que la gran mayoría poseen una base de datos de la que obtienen información (modelo) y se la proporcionan al usuario (controlador) mediante una interfaz (vista).

Si bien algunos frameworks, como Angular o Rails, se considera que implementan un MVC puro, no es el caso de ReactJS (que directamente afirma no ser un framework MVC). Sin embargo, uno de sus frameworks asociados, Remix, el utilizado en este proyecto, gestiona la Vista y el Controlador, dejando el Modelo al programador.

Para la parte del Controlador, Remix utiliza las funciones Loader y Action. La función Loader se encarga de cargar datos obtenidos de la base de datos y la función Action se encarga de tratar y enviar datos a la base de datos. También se encarga de realizar redirecciones entre las diferentes rutas de la aplicación.

Para la parte de la Vista, se utilizan las rutas de Remix, que son, en esencia, páginas web. Un apartado positivo de estas rutas es que pueden estar anidadas, por lo que es más fácil fragmentar la aplicación en diferentes apartados e integrarlos en una sola página o ruta principal.

Para el almacenamiento de datos, se implementó una base de datos NoSQL con MongoDB. Se escogió esta base de datos por la flexibilidad en el almacenamiento de datos y las facilidades a la hora de gestionarla. Junto

con MongoDB, y para asegurar que todo se almacenaba de forma correcta, se implementó Prisma ORM con las funciones CRUD (Create, Read, Update y Delete).

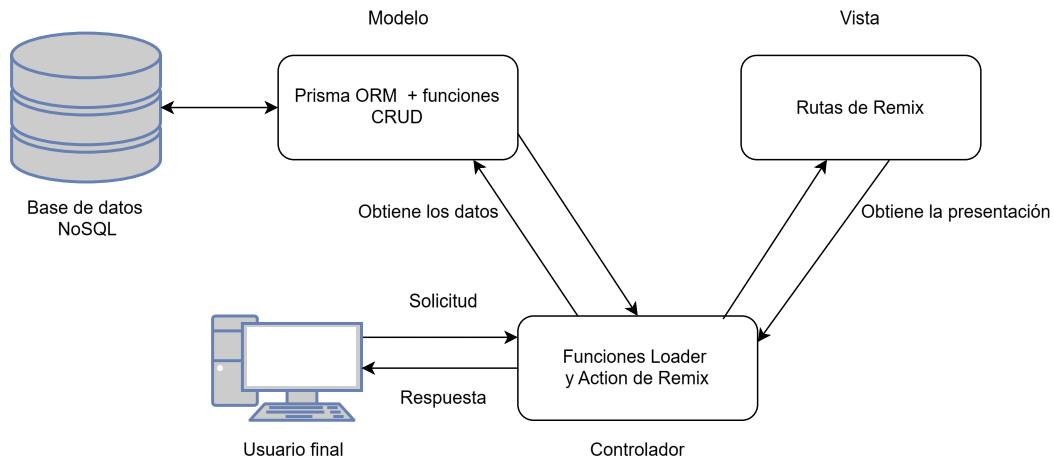


Figura 4.1: Diagrama de la arquitectura del sistema

4.3. Diagrama de transiciones de estados o interfaces

Para facilitar el desarrollo y poder mantener coherencia y orden entre las diferentes interfaces, se ha realizado un diagrama de transiciones de estados o interfaces, que se muestra en la Figura 4.2.

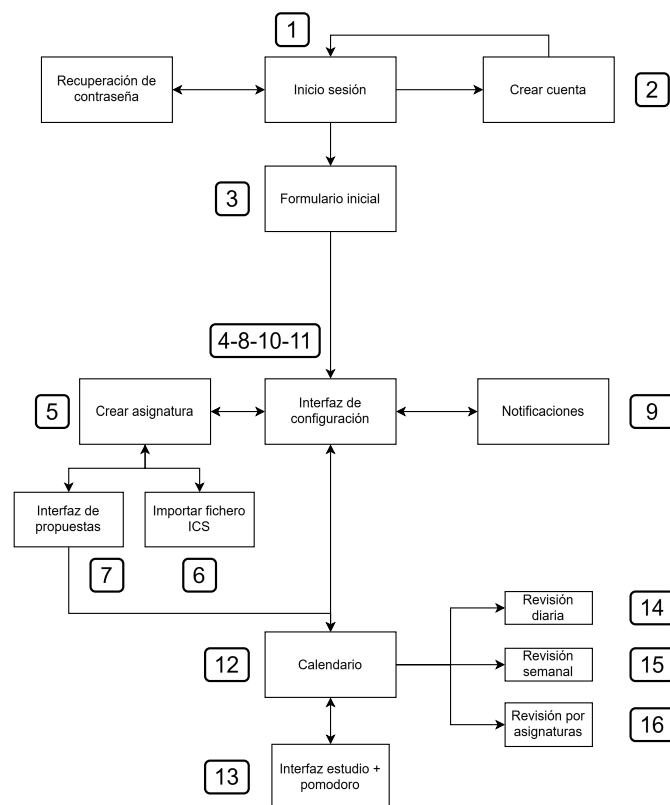


Figura 4.2: Diagrama de transiciones de interfaces

La numeración permite visualizar el flujo que realizaría un usuario que accede por primera vez a la aplicación. Cabe destacar que las interfaces 6, 9, 13, 14, 15 y 16 son optativas para el usuario y puede utilizar las funcionalidades que éstas ofrecen o no.

La interfaz ‘Recuperación de contraseña’ no tiene número asignado porque no corresponde a un flujo normal del usuario. Se trata de un flujo excepcional en el que el usuario olvida su contraseña.

4.4. Modelo de datos

El modelo de datos consta de cinco clases con sus respectivos atributos:

1. Usuario:

- ID - ObjectID: String único que representa a cada usuario.
- Correo electrónico - String: correo electrónico del usuario.
- Nombre - String: nombre de usuario para iniciar sesión.
- Contraseña - String: contraseña para el inicio de sesión.
- Preferencias - Object: objeto que contiene los diferentes días de la semana y las preferencias del usuario sobre cuándo prefiere establecer los bloques de estudio (aplica en la generación de propuestas por parte de la aplicación).

2. Revisión semanal:

- ID - ObjectID: String único que representa a cada revisión semanal.
- authorId - String: usuario que ha realizado la revisión.
- Fecha - Date: fecha de la revisión semanal (solo se realizará una revisión por semana).
- Grado satisfacción - Object: valor del 0 al 10 que permite al usuario visualizar su grado de satisfacción con el trabajo realizado sobre esa asignatura durante la semana junto con el id de la asignatura correspondiente.

3. Asignatura:

- ID - ObjectID: String único que representa a cada asignatura.
- authorId - ObjectID: usuario que ha creado la asignatura.
- Nnombre - String: nombre de la asignatura.
- Horas - Float: número de horas que el usuario quisiera incluir en la planificación semanal en la asignatura.
- SessionSize - Float: tamaño de los bloques de estudio pertenecientes a esta asignatura (aplica en la generación de propuestas por parte de la aplicación).
- Fecha de inicio - Date: fecha y hora de inicio de estudio de la asignatura.
- Fecha de fin - Date: fecha y hora de fin de estudio de la asignatura.
- Color - String (#... valor hex): valor que representa el color con el que se visualizará la asignatura en el calendario.

4. Bloque de estudio:

- ID - String: String único que identifica el bloque de estudio.
- SubjectID - ObjectID: ID de la asignatura a la que pertenece el bloque.

- **SubjectName** - String: nombre de la asignatura a la que pertenece el bloque.
- **BlockID** - Int: ID del elemento de la matriz del calendario en el que se encuentra el bloque (se utiliza para reducir cálculos).
- **Nombre** - String: nombre único que identifica a cada bloque.
- **Notas** - String: notas breves sobre el bloque de estudio
- **Fecha** - Date: fecha y hora de inicio del bloque.
- **Duración** - float: duración real en horas del bloque (permite saber si el bloque de estudio se ha completado o no).
- **Completado** - Int: valor entre 0 y 100 con paso de 25 para saber cuánto se ha completado del bloque.

5. Bloque de clase:

- **ID** - String: String único que identifica el bloque de clase.
- **SubjectID** - ObjectID: ID de la asignatura a la que pertenece el bloque.
- **SubjectName**: nombre de la asignatura a la que pertenece el bloque.
- **BlockID**: ID del elemento de la matriz del calendario en el que se encuentra el bloque (se utiliza para reducir cálculos).
- **Nombre** - string: nombre único que identifica a cada evento.
- **Notas** - string: notas breves sobre el bloque de clase.
- **Fecha** - Date: fecha y hora de inicio del bloque.
- **Duración** - float: duración real en horas del bloque. (permite saber si el bloque de estudio se ha completado o no).

6. Evento:

- **ID** - String: String único que identifica al evento.
- **SubjectID** - ObjectID: ID de la asignatura a la que pertenece el evento.
- **SubjectName** - String: nombre de la asignatura a la que pertenece el evento.
- **BlockID**: ID del elemento de la matriz del calendario en el que se encuentra el evento (se utiliza para reducir cálculos).
- **Nombre** - String: nombre del evento.
- **Notas** - String: notas breves sobre el evento.
- **Fecha** - Date: fecha y hora de finalización del evento.
- **Completado** - Boolean: representa si el evento ha sido completada o no.

4.5. Diagrama del modelo conceptual de la base de datos

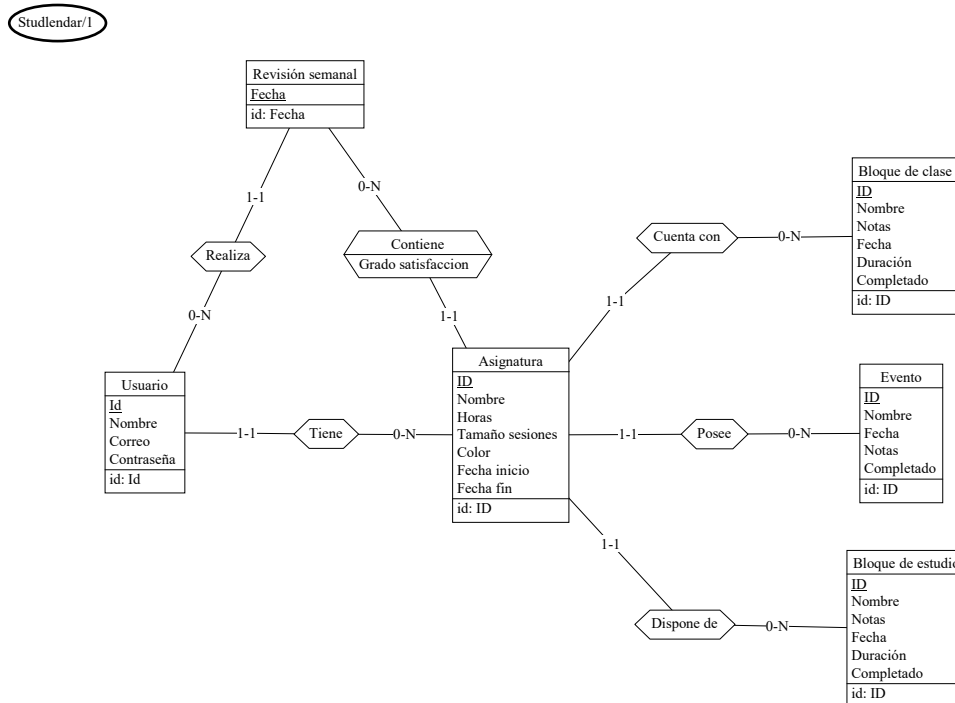


Figura 4.3: Modelo conceptual de la base de datos

Capítulo 5

Fase de Desarrollo

En este capítulo se explican las herramientas que han sido utilizadas para el desarrollo de la aplicación y la interacción entre las mismas.

5.1. Lenguajes de programación, entorno de desarrollo y herramientas

La aplicación web se ha desarrollado utilizando principalmente Javascript y Typescript. También se ha utilizado HTML para la base de la aplicación y CSS para proporcionar estilos a la misma.

Para facilitar las labores de desarrollo, se ha utilizado un Framework que implementa React (una biblioteca de Javascript), el Framework Remix. Se ha elegido este Framework por las facilidades en el enrutado de las diferentes páginas, la carga de datos junto con Prisma ORM, el tratamiento de datos de los formularios y la sencillez para implementar elementos y componentes dinámicos.

Tras escoger el Framework, se debían escoger una serie de herramientas para facilitar el desarrollo, como el editor de código, gestión de la base de datos, gestión de transacciones y control de versiones. A continuación se detallan las herramientas utilizadas en este proyecto:

- **Visual Studio Code:** La herramienta Visual Studio Code es un editor de código fuente que se ha utilizado para el desarrollo, depuración y control de versiones del proyecto, siendo muy útil por la compatibilidad con todos los lenguajes utilizados en este proyecto.
- **Atlas Database:** La herramienta de gestión utilizada para la base de datos es Atlas Database, que permite visualizar la base de datos, transacciones y conexiones en tiempo real. Esta herramienta permite realizar escalado de forma rápida para mayor flujo de datos y conexiones simultáneas.
- **Prisma ORM:** Para simplificar las lecturas, actualizaciones y escrituras de datos, se optó por implementar la herramienta Prisma ORM (object-relational mapping) para gestionar estas transacciones. Esta herramienta facilita las transacciones mediante la creación de modelos que definen la base de datos y sus colecciones. Además, presenta sus propias consultas predefinidas, lo que facilita la realización de todo tipo de transacciones y filtrado de datos.
- **Control de versiones, GitHub:** Como cualquier proyecto de software, Studlendar se tenía que alojar en alguna plataforma. Esa plataforma es GitHub, que aparte de servir como alojamiento para el proyecto, proporciona un sistema de control de versiones que facilita el seguimiento y mantenimiento del proyecto.

5.2. Decisiones de seguridad sobre la base de datos

La base de datos es uno de los activos más importantes que un proyecto o aplicación tiene. Se debe tener máximo precaución en los formularios, para evitar posibles ataques de inyección NoSQL, escalado de privilegios y robo de datos.

La utilización de Prisma ORM permite, mediante los modelos creados en su configuración, mantener la coherencia de la base de datos, permitiendo solo los tipos de datos definidos en los modelos, y sirviendo como una primera capa de filtrado.

Otra de las medidas principales fue la encriptación de contraseñas utilizando la biblioteca Bcrypt. Esta biblioteca de NODE permite crear un hash de un string.

A mayores, a parte de utilizar el módulo previamente mencionado, se hacen comprobaciones lógicas manuales sobre los datos introducidos en los diversos formularios como, por ejemplo:

- No repetir nombre ni correo electrónico al registrarse (cada usuario tendrá un nombre y correo únicos).
- La introducción de fechas en la creación de asignaturas debe ser siempre superior a la fecha y hora actuales.
- No puede haber dos bloques de estudio de la misma asignatura de forma concurrente en la misma fecha y hora.

5.3. Herramientas para accesibilidad

Un apartado de gran importancia en cualquier página web es la accesibilidad. Un sitio web tiene accesibilidad web si puede ser usado por personas con discapacidades de diverso tipo (auditivas, cognitivas, neurológicas, físicas...). Dicha accesibilidad también beneficia al resto de personas, puesto que la navegación y visibilidad es mejor y más sencilla.

La accesibilidad web se divide en tres niveles principales, cada cual más exigente y mejor que el anterior en esta materia:

1. **WCAG Level A:** accesibilidad mínima o esencial. Contiene lo mínimo para que una persona con discapacidad pueda utilizar la web. Puesto que los criterios de accesibilidad del primer nivel ascienden a 30 en múltiples apartados, se ponen algunos ejemplos:
 - La información no se debe mostrar únicamente a través de color, forma o tamaño.
 - Los formularios deben incluir 'label' o instrucciones.
2. **WCAG Level AA:** accesibilidad recomendada. Contiene todos los criterios del nivel A y nuevos criterios para mejorar la accesibilidad a la gran mayoría de usuarios. A los criterios anteriores se suman otros 20, como por ejemplo:
 - Contraste de colores adecuado entre texto y fondo.
 - Orden lógico del contenido, de arriba a abajo.
3. **WCAG Level AAA:** accesibilidad excelente. Se considera el nivel ideal, es muy exigente y puede llegar a ser complicado de obtener según el proyecto. A los criterios del nivel anterior, se suman otros 28, como por ejemplo:
 - El ratio de contraste entre texto y fondo debe ser de, al menos, 7 a 1.
 - Los encabezados (elementos H1-6) se utilizan para organizar el contenido.

Además, en muchos casos, las accesibilidad web se convierte en un requisito legal y se realizan evaluaciones de accesibilidad de toda la página web. Por ello, es un apartado que no debe dejarse a un lado y se debe tener en cuenta desde el principio del desarrollo, para evitar tener que rehacer trabajo en un futuro.

En este proyecto, algunas herramientas utilizadas para mejorar la accesibilidad web han sido:

- Colour Contrast checker: herramienta para ver el contraste entre colores y verificar si pasa los diferentes niveles de accesibilidad (WCAG A, AA y AAA).
- ARIA in HTML: guías de desarrollo de la W3C sobre accesibilidad.

5.4. Elección de la paleta de colores

Estamos rodeados de colores y todos ellos surgen de los colores primarios (rojo, verde y azul). El color, tal y como se afirma en [2], afecta a nuestras emociones, comportamiento y reacciones. Por esto, se realizó el estudio [6], para comprobar los efectos de estos colores en el desempeño de tareas. En este estudio se afirma que el color azul está comúnmente asociado con "transparencia, paz y tranquilidad", provocando una mejora en tareas creativas, aunque se quedaba atrás en las tareas orientadas al detalle, donde ganaba el color rojo. Los resultados finales afirmaban que el color afectaba en función de la naturaleza de la tarea.

Por otro lado, el estudio [2] afirma que el color rojo se asocia con palabras relacionadas con el fracaso ([12]) y que los profesores que utilizaban el color rojo tienden a corregir más negativamente que con colores azules ([9]).

Con todos estos estudios, se decidió escoger, por defecto, una paleta de colores con diversos tonos de azules, integrando también el blanco que "ayuda a nuestra mente a concentrarse y contribuye a la organización" ([17]). Esta paleta de colores se utiliza en toda la aplicación por defecto y, a mayores, el usuario puede cambiar el color asociado a cada asignatura, que se utilizará a la hora de mostrar sus bloques y eventos asociados.

Capítulo 6

Fase de pruebas

En este capítulo se muestran las pruebas que se han realizado para comprobar el correcto funcionamiento de la aplicación. El capítulo se divide en dos secciones que corresponden a:

- Pruebas de funcionalidad de la aplicación.
- Pruebas de accesibilidad web de la aplicación.

6.1. Pruebas de aceptación de la aplicación

Para comprobar si la aplicación cumplía con las necesidades de los usuarios, se realizaron dos iteraciones de pruebas de aceptación de la aplicación, utilizando el modelo TAM (ver Figura 6.1) para diseñar los formularios utilizados en cada una.

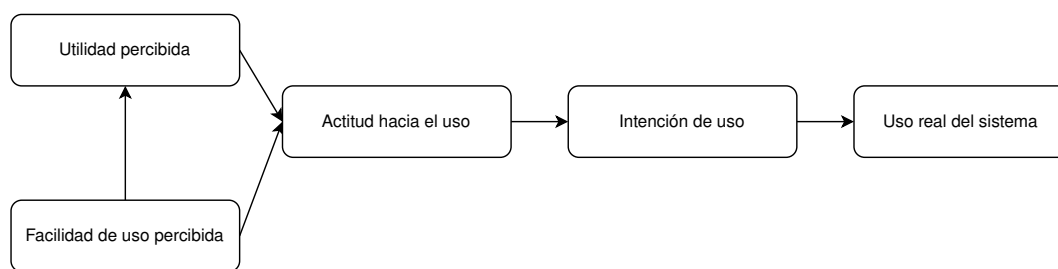


Figura 6.1: Diagrama del sistema TAM utilizado en los formularios (Figura extraída de TAM)

La realización de dos pruebas de aceptación permitió encontrar fallos en funcionalidades de la aplicación y mejoras de cara a la aplicación final.

Como estas pruebas tenían mayor duración que las pruebas de concepto realizadas en las iteraciones de análisis (aproximadamente las pruebas de concepto tenían una duración de entre 15 y 40 minutos, mientras que estas pruebas se alargaron, en algunos casos, hasta más de una hora), se redujo el número de usuarios que participaron en las mismas dada la limitada disponibilidad de muchos de los participantes (ver Tabla 6.1). A pesar del menor número de resultados, la información recogida fue amplia y de gran utilidad para seguir con el desarrollo de la aplicación.

Tabla 6.1: Listado de participantes en las iteraciones de desarrollo

Código de participante	Estudios	Participación iter. desarrollo 1	Participación iter. desarrollo 2
P1	Grado superior	X	-
P2	Grado en ADE	-	-
P3	Grado en Medicina	-	-
P4	Grado en Ingeniería Informática	X	X
P5	Bachillerato de adultos	-	-
P6	Grado en Ingeniería Industrial	X	X
P7	Grado en Ingeniería Química	-	-
P8	Grado en Ingeniería Mecánica	-	-
P9	Máster en Arquitectura	-	-
P10	Grado en Matemáticas	-	-
P11	Doble Grado en Matemáticas y Física	-	-
P12	Grado en Ingeniería Informática	-	-

Estas pruebas de funcionalidad se han realizado de forma iterativa, en dos iteraciones o fases:

6.1.1. Iteración 1

En la Iteración 1 se presentó la aplicación en un estado poco avanzado de la aplicación, pero con parte de la funcionalidad básica implementada (formulario inicial, propuestas de bloques de estudio, calendario, creación de bloques y eventos).

Utilizando el cuestionario TAM, se evaluó la intención de uso de los usuarios, dejándoles probar la aplicación sin explicaciones (paso 5 en el diagrama TAM). A mayores, y complementado ésto, se les pide rellenar un formulario (A.3) donde se les preguntaba sobre la utilidad y facilidad de uso percibidas de las diferentes interfaces, con una puntuación del 1 al 10, pudiendo así identificar aquéllas que obtuvieran una baja puntuación para su potencial mejora. Los resultados de este formulario se pueden ver en la Tabla 6.2, en la que se indica la pregunta, el código de participante junto con sus respuestas y el valor medio obtenido para cada pregunta.

Tabla 6.2: Resultados de los cuestionarios TAM

Pregunta	P1	P4	P6	Valor medio
Interfaz de inicio de sesión, puntuación a nivel de utilidad	8	10	9	9,00
Interfaz de inicio de sesión, puntuación a nivel de facilidad	10	10	9	9,67
Interfaz del formulario, puntuación a nivel de utilidad	9	10	5	8,00
Interfaz del formulario, puntuación a nivel de facilidad	10	9	9	9,33
Interfaz de configuración del calendario, puntuación a nivel de utilidad	10	10	7	9,00
Interfaz de configuración del calendario, puntuación a nivel de facilidad	10	10	9	9,67

Continúa en la siguiente página

Tabla 6.2: Resultados de los cuestionarios TAM (Continuación)

Interfaz para crear asignaturas, puntuación a nivel de utilidad	10	10	5	8,33
Interfaz para crear asignaturas, puntuación a nivel de facilidad	10	9	8	9,00
Interfaz para seleccionar propuestas, puntuación a nivel de utilidad	10	8	3	7,00
Interfaz para seleccionar propuestas, puntuación a nivel de facilidad	7	5	4	5,33
Interfaz principal, puntuación a nivel de utilidad	9	10	6	8,33
Interfaz principal, puntuación a nivel de facilidad	10	10	7	9,00
Popup para crear un evento, puntuación a nivel de utilidad	10	10	9	9,67
Popup para crear un evento, puntuación a nivel de facilidad	10	10	9	9,67
Interfaz de pomodoro, puntuación a nivel de utilidad	10	10	3	7,67
Interfaz de pomodoro, puntuación a nivel de facilidad	10	10	10	10,00

Aunque en general los valores promedio superaban el 7 (alcanzando puntuación media de 8,375 y 8,959 en términos de utilidad y facilidad, respectivamente), una de las interfaces que tuvo baja puntuación, tal y como se observa en la Tabla 6.2, fue la interfaz de propuestas (Figura 6.2), siendo ésta una interfaz que provocó confusión entre los usuarios, puesto que no entendían bien el propósito de esta interfaz ni su funcionamiento. Ésto generó las propuestas ITD1.3 e ITD1.6.



Figura 6.2: Interfaz de propuestas

Otro cambio de gran valor para la aplicación, fue la integración del correo electrónico en los usuarios (propuesta ITD1.1), permitiendo así la posibilidad de recuperar la contraseña a través de un código generado de forma aleatoria que se envía a la dirección de correo electrónico ligada a su cuenta.

Respecto a cambios generales, se realizaron algunas modificaciones explicadas en las propuestas ITD1.2, ITD1.6 ITD1.7 e ITD1.10, mejorando así diferentes interfaces de la aplicación.

Tabla 6.3: Listado de propuestas de la iteración de desarrollo nº1

Código de Propuesta	Propuesta	Descripción	Grado de alineación con el proyecto	Factibilidad
ITD1.1	Añadir correo al usuario	Añadir correo a la cuenta de usuario para recuperar la contraseña o recibir notificaciones	Alto	Sí
ITD1.2	Visualización de fecha y hora actuales en el calendario	Añadir una visualización de la fecha y hora actuales en la interfaz principal del calendario	Medio	Sí
ITD1.3	Mayor facilidad para ver propuestas	Mejorar la interfaz de propuestas para que se visualicen mejor las propuestas de bloques de estudio	Medio	Sí
ITD1.4	Añadir zoom al calendario	Opción para hacer zoom al calendario	Medio	No
ITD1.5	Editar la interfaz de estudio	Editar la interfaz de estudio para añadir fondo personalizado y poder quitar el pomodoro si no se quiere	Medio	No
ITD1.6	Añadir más popups informativos por la aplicación	Añadir más popups informativos de funcionalidades de la aplicación	Bajo	Sí
ITD1.7	Mostrar los números de los días de la semana del calendario	Mostrar el número de cada día de la semana en el calendario	Bajo	Sí
ITD1.8	Crear un grupo de colores preseleccionados para elegir	Crear una selección de colores predefinida para el usuario	Bajo	Sí
ITD1.9	Revisión diaria y semanal a elección del usuario	Permitir al usuario elegir si quiere o no realizar las revisiones	Bajo	Sí
ITD1.10	Gestionar el tamaño de las sesiones mediante un slider	Gestionar el tamaño de las sesiones de estudio en el formulario inicial mediante un slider	Bajo	Sí
ITD1.11	Añadir botón para pantalla completa	Añadir un botón que active la pantalla completa	Bajo	No

Esta primera iteración permitió visualizar cambios y mejoras que realizar en la aplicación, lo que permitió pulirla en detalles que mejoran la experiencia de usuario. A mayores, también se detectaron errores en la programación que, principalmente, estaban relacionados con el almacenamiento y gestión de fechas y horas. La detección de ésto

en una etapa temprana de desarrollo permitió solucionarlos y evitar problemas en las siguientes etapas.

6.1.2. Iteración 2

En la Iteración 2, se presentó a los usuarios una aplicación más avanzada, con las interfaces mejoradas y con unas funcionalidades más pulidas. Esta última iteración se utilizó para encontrar posibles errores y mejorar algunas interfaces para hacerlas más fáciles y usables para los usuarios. Se siguió el mismo procedimiento, utilizando el modelo TAM y un formulario (A.4). Los resultados del formulario TAM pueden verse en la Tabla 6.4. Sin compararnos los resultados obtenidos con la iteración anterior, cabe destacar que la mejora en relación a la utilidad y facilidad, subiendo a 9 y 9 de media. A pesar de esto, se sigue observando que la interfaz de propuestas es la que resulta menos útil y fácil. Marcando un objetivo de mejora claro para el trabajo futuro.

Tabla 6.4: Listado de propuestas de la iteración de desarrollo n^o2

Pregunta	P4	P6	Valor medio
Interfaz de inicio de sesión, puntuación a nivel de utilidad	9	10	9,50
Interfaz de inicio de sesión, puntuación a nivel de facilidad	10	10	10,00
Interfaz de recuperar contraseña, puntuación a nivel de utilidad	10	10	10,00
Interfaz de recuperar contraseña, puntuación a nivel de facilidad	10	10	10,00
Primera interfaz del formulario, puntuación nivel de utilidad	9	8	8,50
Primera interfaz del formulario, puntuación nivel de facilidad	9	9	9,00
Interfaz de configuración del calendario, puntuación a nivel de utilidad	10	8	9,00
Interfaz de configuración del calendario, puntuación a nivel de facilidad	10	9	9,50
Interfaz para crear asignaturas, puntuación a nivel de utilidad	9	10	9,50
Interfaz para crear asignaturas, puntuación a nivel de facilidad	8	10	9,00
Interfaz para seleccionar propuestas, puntuación a nivel de utilidad	7	5	6,00
Interfaz para seleccionar propuestas, puntuación a nivel de facilidad	9	6	7,50
Interfaz principal, puntuación a nivel de utilidad	10	10	10,00
Interfaz principal, puntuación a nivel de facilidad	9	9	9,00
Popup para crear un evento, puntuación a nivel de utilidad	9	10	9,50
Popup para crear un evento, puntuación a nivel de facilidad	9	9	9,00
Interfaz de pomodoro, puntuación a nivel de utilidad	10	8	9,00
Interfaz de pomodoro, puntuación nivel de facilidad	8	9	8,50
Interfaz de revisiones, puntuación nivel de utilidad	9	9	9,00
Interfaz de revisiones, puntuación nivel de facilidad	7	10	8,50

Como se puede observar en el número de propuestas de la Tabla 6.5, se implementaron todas las propuestas con alineamiento medio-alto, buscando una mayor aceptabilidad de la aplicación por parte de los usuarios.

Tabla 6.5: Listado de propuestas de la iteración de desarrollo n^o2

Código de Propuesta	Propuesta	Descripción	Grado de alineación con el proyecto	Factibilidad
ITD1.1	Poder borrar bloques y eventos	Poder borrar bloques y eventos	Alto	Sí
ITD1.2	Mayor facilidad para ver propuestas	Mejorar la interfaz de propuestas para que se visualicen mejor las propuestas de bloques de estudio	Medio	Sí
ITD1.3	Explicar mejor interfaces como la de propuestas	Añadir más y mejores explicaciones a las interfaces	Medio	Sí
ITD1.4	Cambiar el cierre de sesión a configuración	Cambiar el botón de cierre de sesión al apartado de configuración	Bajo	No
ITD1.5	Añadir el logo de la aplicación en más interfaces	Añadir el logo de la aplicación en más interfaces	Bajo	No
ITD1.6	Guardar el número de pomodoros realizados	Guardar la cantidad de pomodoros completados	Bajo	No

La interfaz de propuestas de bloques de estudio, a pesar de haber sido modificada tras la Iteración 1, siguió generando dudas entre los usuarios en esta segunda iteración, por lo que se procedió a modificarla de nuevo, buscando una comprensión más fácil de la misma, dando como resultado la interfaz que se presenta en la Figura 6.3

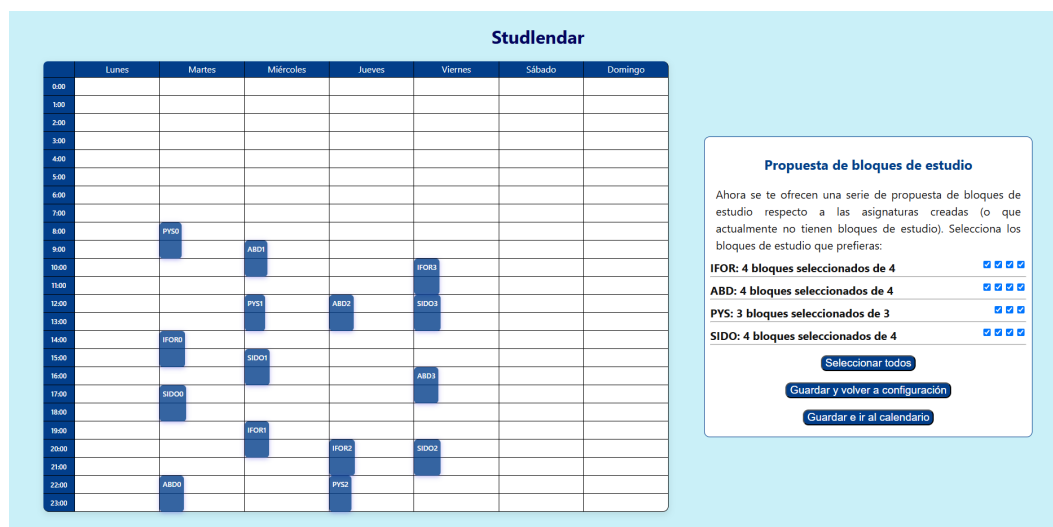


Figura 6.3: Interfaz de propuestas en su versión final

6.2. Pruebas de accesibilidad web de la aplicación

La accesibilidad, tal y como se mencionó en la Sección 5.3 es de vital importancia para cualquier aplicación web actual. Por lo tanto, para poder medir la accesibilidad de las diferentes rutas del proyecto y poder realizar potenciales mejoras en un futuro, se ejecutaron dos tipos de pruebas:

- Pruebas con la herramienta LightHouse de Google: Esta herramienta de código abierto realiza auditorías sobre la página web de rendimiento, accesibilidad, buenas prácticas y SEO. Cada uno de estos apartados se puntúa del 0 al 100, pudiendo tener una puntuación de ‘error’ si hay fallos graves. La interfaz de resultados tras realizar la auditoría es la siguiente:

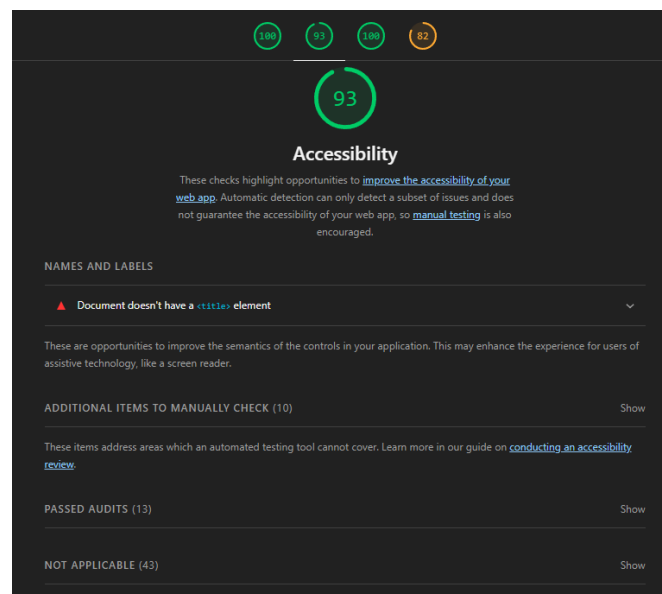


Figura 6.4: Captura de la interfaz de resultados de una auditoría realizada con Google LightHouse

- Pruebas manuales sobre la aplicación: aparte de utilizar la herramienta LightHouse de Google, es recomendable realizar pruebas manuales sobre la aplicación. La propia Google lo afirma en la interfaz de resultados ('...not guarantee the accessibility of your web app, so manual testing is also encouraged.'). Para hacer pruebas manuales, hay que realizar una serie de pasos detallados, tal y como se explica en 'Cómo revisar la accesibilidad web':
 - Navegación por teclado.
 - Pruebas con un lector de pantalla.
 - Revisión de los elementos interactivos, que deben mostrar su propósito y estado.
 - Aprovechar encabezados y puntos de referencia.

Por cuestiones de tiempo no se han podido realizar todas las pruebas deseadas sobre accesibilidad, priorizando la navegación por teclado sobre el resto.

Estas pruebas han mostrado que, en términos generales, la aplicación se puede usar mediante la navegación por teclado, pero en la interfaces donde está el calendario, esta operación se vuelve más complicada y compleja, alejándose del alcance inicial del proyecto.

6.2.1. Propuestas de mejora

La accesibilidad web es un apartado complejo de abordar en el desarrollo de páginas web, y siempre hay margen para mejorar la navegación, interacción y estilos. Puesto que alcanzar un nivel de accesibilidad alto es una labor que requiere una dedicación grande y la realización de esquemas sobre cómo debería realizarse la navegación, estilos a aplicar, incluso la creación de componentes web exclusivos para esta tarea, a continuación se presentan algunas propuestas de mejora sobre la aplicación:

- Mejora de la adaptación a diferentes tamaños de pantalla.
- Mejora de estilos a nivel de visibilidad.
- Modificación de elementos o rutas de la web para mejorar la accesibilidad y rendimiento de la misma.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Durante la realización de este trabajo de fin de grado se ha conseguido crear una aplicación web que ayude a estudiantes de educación superior, tanto de grados universitarios como de otros niveles académicos, a organizarse y poder tener un seguimiento de las diferentes tareas que supone estudiar. Retomando los objetivos iniciales, se puede afirmar que éstos han sido logrados:

- *Diseñar y desarrollar una aplicación web de calendario mediante la cual los estudiantes puedan organizar y hacer un seguimiento de sus estudios y las tareas asociadas a los mismos.*

Se ha desarrollado una aplicación web de calendario que cumple con los requisitos especificados en la Sección 3.2, obtenidos aplicando una planificación basada en Diseño Centrado en Usuario de forma iterativa.

Cabe destacar que el proceso de investigación, iteraciones, pruebas y análisis de los datos recogidos es costoso en términos de tiempo, y que no siempre se pueden alinear las expectativas de todos los usuarios con el objetivo del proyecto, puesto que tener en cuenta todas las peticiones o propuestas significaría un trabajo mucho mayor y una desviación respecto a la visión inicial del proyecto.

- *Alinear las funcionalidades de la herramienta con las necesidades y preferencias de los usuarios finales.*

Siguiendo las recomendaciones de [23], he tratado de hacer partícipe a los usuarios en las diferentes etapas del ciclo de vida de la herramienta. Para involucrar a los usuarios durante el proyecto, se han organizado tanto pruebas de concepto en tiempo de diseño (tres iteraciones), como pruebas de aceptación en la fase de desarrollo con prototipos funcionales (dos iteraciones). Para ello, me he valido de formularios para recabar información. Las pruebas de concepto, han permitido recoger información diversa sobre la aplicación, propuestas y mejoras. Por otro lado, las pruebas de aceptación han servido para validar la aplicación y encontrar posibles fallos, mejoras técnicas y visuales. Gracias a ésto, se ha podido asegurar que la idea de la aplicación es útil y fácil para los potenciales usuarios, teniendo en cuenta sus propuestas de cambio y mejora, siempre que éstas se mantuvieran en el alcance inicial del proyecto y dejando el resto de ideas como propuestas de trabajo futuro.

7.2. Limitaciones

Debido a que el Trabajo de Fin de Grado tiene un alcance limitado por tiempo, hay una serie de apartados que han quedado sin cubrir, algunos de forma parcial y otros de forma total:

- **Pruebas realizadas:** las pruebas realizadas no han sido, ni mucho menos, completas. Se han realizado pruebas de funcionalidad, pruebas de aceptación y algunas pruebas de accesibilidad web. Sin embargo, las

pruebas de accesibilidad no han cubierto todo, por ejemplo, no se han realizado pruebas con lectores de pantalla. Tampoco se han realizado pruebas sobre la base de datos, ni de rendimiento ni carga.

- **Accesibilidad:** La aplicación web es funcional en ordenadores y portátiles, pero no está correctamente adaptada para dispositivos móviles y tablets. Algunas de las interfaces, para alcanzar un grado alto de accesibilidad, requieren de una gran cantidad de horas, pruebas y herramientas de testeo (como son los lectores de pantalla), por lo que se decidió dejarlas fuera del alcance inicial del proyecto.
- **Funcionalidades:** para poder crear una aplicación funcional y tener la funcionalidad más importante implementada y funcionando correctamente, se tuvieron que dejar algunas propuestas de lado, unas por tiempo y otras por complejidad de desarrollo, como por ejemplo:
 - Integrar estrategias para tener en cuenta la Curva del Olvido ([15]) a la hora de formular las propuestas de bloques de estudio.
 - Configurar las propuestas de bloques de estudio teniendo en cuenta los bloques de clase y las preferencias manifestadas por el usuario (antes, en el mismo día o después de las clases).
 - Habilitar la repetición de bloques: actualmente los bloques se crean para una semana y no hay una opción que permita establecer una repetición diaria, semanal o mensual.

El resto de propuestas que han quedado fuera del alcance del proyecto se pueden ver en el apartado de factibilidad de las Tablas 3.2, 3.3, 3.4, 6.3 y 6.5.

7.3. Trabajo futuro

Si bien la aplicación desarrollada cumple con las funcionalidades mínimas y más importantes, hay una serie de mejoras que se irán implementando de forma progresiva:

- Mejora de la accesibilidad y aspecto general de la aplicación.
- Implementación de propuestas e ideas que no se han implementado por dar prioridad a las funcionalidades más básicas. Algunos ejemplos son las propuestas son IT2.01 (Curva del Olvido), IT2.06 (revisión de la organización de estudios), IT3.05 (plantilla de estudio u organización predefinida en función de los estudios). Para ver todas las propuestas de trabajo futuro, revisar el apartado ‘factibilidad’ de las Tablas 3.2, 3.3 y 3.4
- Fomentar el uso de la herramienta mediante la realización de una aplicación para PC instalable para trabajo en local y otra para dispositivos móviles.
- Exportación del calendario creado dentro de la aplicación en formato ICS.
- Integración con aplicaciones de gestión de proyectos y para tomar notas como Notion o similar.

7.4. Valoración personal

El proyecto, en todas sus fases y apartados, ha sido gran valor para mí. No solo me ha permitido conocer de primera mano las complejidades y dificultades de realizar una aplicación web, sino también descubrir la gran cantidad de funcionalidades que se pueden crear e implementar con diversas herramientas. Además, he aprendido sobre Diseño Centrado en el Usuario, las dificultades que conlleva aplicarlo y la importancia de utilizar este tipo de métodos para involucrar a los usuarios finales, produciendo así una aplicación alineada con sus necesidades.

Bibliografía

- [1] B.W. Boehm. «Software risk management: principles and practices». En: *IEEE Software* 8.1 (1991), págs. 32-41. DOI: 10.1109/52.62930. (último acceso: 02/06/2025).
- [2] Patricia Valdez y Albert Mehrabian. «Effects of color on emotions.» En: *Journal of experimental psychology. General* 123 4 (1994), págs. 394-409. URL: <https://api.semanticscholar.org/CorpusID:4541770>. (último acceso: 30/05/2025).
- [3] G. Booch, J. Rumbaugh e I Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999. ISBN: 978-0321267979.
- [4] William Sugar. «Applying human-centered design to technology integration: Three alternative technology perspectives». En: *Journal of Computing in Teacher Education* 19.1 (2002), págs. 12-17. (último acceso: 05/07/2025).
- [5] Bernard Desruisseaux. *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. RFC 5545. Sep. de 2009. DOI: 10.17487/RFC5545. URL: <https://www.rfc-editor.org/info/rfc5545>. (último acceso: 29/03/2025).
- [6] Rui Juliet Zhu. «Blue or Red? Exploring the Effect of Color on Cognitive Task Performances». En: *Science (New York, N.Y.)* 323 (mar. de 2009), págs. 1226-9. DOI: 10.1126/science.1169144. (último acceso: 30/05/2025).
- [7] T. Amabile y S. & Kramer. *The Power of Small Wins*. 2011. URL: <https://www.hbs.edu/faculty/Pages/item.aspx?num=40244>. (último acceso: 13/06/2025).
- [8] Bob Hughes, Mike Cotterell y Mall Rajib. *Software Project Management, 5th edition*. McGraw-Hill, 2011. ISBN: 978-0-07-107274-8.
- [9] Heather Albanesi. «Seeing Red: Quality of an essay, color of the grading pen and student reactions to the grading process». En: *The Social Science Journal* 50 (mar. de 2013), pág. 96. DOI: 10.1016/j.soscij.2012.07.005. (último acceso: 30/05/2025).
- [10] Interaction Design Foundation. *User Centered Design (UCD)*. 2016. URL: <https://www.interaction-design.org/literature/topics/user-centered-design>. (último acceso: 26/03/2025).
- [11] S. Cornér, E. Löfström y K. Pyhältö. *The relationship between doctoral students' perceptions of supervision and burnout*. 2017. URL: <https://www.informingscience.org/Publications/3754>. (último acceso: 13/06/2025).
- [12] Jana Fikrlová et al. «The power of red: The influence of colour on evaluation and failure - A replication». En: *Acta Psychologica* 198 (jun. de 2019). DOI: 10.1016/j.actpsy.2019.102873. (último acceso: 30/05/2025).
- [13] Yannis Dimitriadis, Roberto Martínez Maldonado y Korah Wiley. *Human-Centered Design Principles for Actionable Learning Analytics*. 2020. URL: https://link.springer.com/chapter/10.1007/978-3-030-64363-8_15. (último acceso: 23/03/2025).

- [14] Luis P. Prieto. *Cultivating the progress loop in your PhD*. 2020. URL: <https://ahappyphd.org/posts/progress-loop/>. (último acceso: 26/03/2025).
- [15] Charuan Aguilera. *Qué es la curva del olvido y cómo evitarla*. 2023. URL: <https://www.ispring.es/blog/curva-del-olvido>. (último acceso: 05/07/2025).
- [16] Cobaya Labs. *La importancia de un enfoque centrado en el usuario en el desarrollo de software*. 2023. URL: <https://es.linkedin.com/pulse/la-importancia-de-un-enfoque-centrado-en-el-usuario-desarrollo-nrbef>. (último acceso: 23/03/2025).
- [17] Editorial Team. *Psicología del color blanco en el Diseño*. 2023. URL: <https://thecolor.blog/es/psicologia-del-color-blanco/>. (último acceso: 30/05/2025).
- [18] Diana Walsh. *Mockup: qué es, para qué sirve y ejemplos*. 2023. URL: <https://blog.hubspot.es/website/mockup-que-es>. (último acceso: 23/03/2025).
- [19] Mark Dziak. *Technology acceptance model (TAM)*. 2024.
- [20] Universidad Europea. *¿Qué es un prototipo y para qué sirve?* 2024. URL: <https://creativecampus.universidadeuropea.com/blog/que-es-prototipo/>. (último acceso: 08/06/2025).
- [21] Geeks for Geeks. *MVC Framework Introduction*. 2024. URL: <https://www.geeksforgeeks.org/software-engineering/mvc-framework-introduction/>. (último acceso: 25/06/2025).
- [22] Ricardo Ernesto Pérez Ibarra, Mercedes Idania López Valenzuela, Daniela María Soto Saucedo y Stephania de Jesús Ramírez Torres. *Relación entre la procrastinación académica y la motivación en universitarios*. Universidad de Guadalajara, Centro Universitario de Ciencias de la Salud, 2024. Cap. 19. ISBN: 978-84-10470-25-5. URL: https://www.researchgate.net/profile/Cinthia-Cruz-Del-Castillo/publication/387075548_investigacion_actual_en_psicologia_EB00K1/links/675f3be2eea8d248be713dcd/investigacion-actual-en-psicologia-EB00K1.pdf#page=302. (último acceso: 23/03/2025).
- [23] Paraskevi Topali et al. «Designing human-centered learning analytics and artificial intelligence in education solutions: a systematic literature review». En: *Behaviour & Information Technology* 44.5 (2024), págs. 1071-1098. DOI: 10.1080/0144929X.2024.2345295. URL: <https://doi.org/10.1080/0144929X.2024.2345295>. (último acceso: 23/03/2025).
- [24] Julia Martins. *Cómo entender los procesos iterativos (con ejemplos)*. 2025. URL: <https://asana.com/es/resources/iterative-process>. (último acceso: 28/06/2025).

Apéndice A

Formularios utilizados con los participantes

A.1. Formulario perteneciente a la iteración de análisis nº2

El formulario para la iteración 2 se realizó con la herramienta Microsoft Forms. URL al formulario: <https://forms.office.com/e/pku2SfJXYE>.

A.2. Formulario perteneciente a la iteración de análisis nº3

El formulario para la iteración 3 se realizó con la herramienta Google Forms debido a que mostraba mejor los datos que la herramienta Microsoft Forms. URL al formulario: <https://forms.gle/JmAekmev4afFY1BcA>.

A.3. Formulario perteneciente a la iteración de desarrollo nº1

El formulario para la iteración de desarrollo fase 1 se realizó con la herramienta Google Forms. URL al formulario: <https://forms.gle/omY26FgnCtPtZ6EF9>.

A.4. Formulario perteneciente a la iteración de desarrollo nº2

El formulario para la iteración de desarrollo fase 2 se realizó con la herramienta Google Forms. URL al formulario: <https://forms.gle/ZeBHnMH8XmD1RA1N9>.