

GRADO EN INGENIERÍA EN TECNOLOGÍA DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

***DISEÑO E IMPLEMENTACIÓN DE UNA
APLICACIÓN MÓVIL MULTIPLATAFORMA
PARA UNA RED SOCIAL UNIVERSITARIA***

Alumno: González, Vega, Joshua

Director: Agujeta, Prieto, Gorka

Curso: 2018-2019

Fecha: Bilbao, 24 de junio de 2019

Índice

1	Introducción	11
1.1	Contexto	12
2	Objetivos	13
2.1	Objetivos del proyecto (alcance)	13
2.1.1	Objetivo Principal	13
2.1.2	Objetivo secundario	13
2.2	Objetivos personales	13
2.3	Especificaciones	14
2.3.1	Autenticación	15
2.3.2	Explora	15
2.3.3	Espacio personal	15
2.3.4	Perfil del usuario	16
2.3.5	Mensajería	17
2.3.6	Notificaciones	17
2.3.7	Frequently Asked Questions (FAQ)	17
3	Beneficios	18
3.1	Beneficios Técnicos	18
3.2	Beneficios Económicos	18
3.3	Beneficios sociales	19
4	Análisis de alternativas	20
4.1	Análisis de tipo de aplicación	20
4.1.1	Aplicación Nativa	20
4.1.2	Aplicación Web	21
4.1.3	Aplicación Híbrida	21
4.1.4	Valoración y elección	22
4.2	Análisis de los frameworks	23
4.2.1	Flutter	24
4.2.2	Xamarin	24
4.2.3	Cordova	24
4.2.4	Valoración y elección	25
4.3	Análisis de tecnología a emplear sobre Cordova	25

4.3.1	JQuery Mobile	25
4.3.2	IONIC	26
4.3.3	Valoración y elección	26
4.4	Análisis de versión de IONIC.....	27
4.4.1	V2, V3 frente a la V4	27
4.4.2	Valoración y elección	27
5	Análisis de riesgos.....	28
5.1	Requerimientos incompletos	28
5.2	Planificación incorrecta.....	29
5.3	Presupuesto limitado.....	30
5.4	Pérdida de producto desarrollado	30
6	Descripción de la solución.....	33
6.1	Modelo en V.....	33
6.2	Control de versiones.....	36
6.2.1	Ramas Feature	37
6.2.2	Ramas Release	37
6.2.3	Ramas Hotfix	37
6.3	Resumen del diseño.....	37
7	Planificación del trabajo.....	43
7.1	Fases.....	43
7.1.1	Especificación.....	43
7.1.2	Análisis y formación	44
7.1.3	Diseño	44
7.1.4	Desarrollo.....	45
7.1.5	Pruebas	45
7.1.6	Documentación.....	46
7.2	Tareas	46
7.3	Gantt.....	49
7.4	Seguimiento de las tareas	52
7.5	Plan de pruebas.....	53
7.5.1	Prueba de login	53

7.5.2	Prueba de comunicación.....	54
7.5.3	Prueba de modificación de datos	55
7.5.4	Prueba de navegación.....	56
7.5.5	Prueba de visualización.....	57
8	Presupuesto	58
9	Resultados	61
9.1	Aplicación final.....	61
9.1.1	Login.....	61
9.1.2	Mi Espacio	62
9.1.3	Explora	63
9.1.4	Perfil.....	64
9.1.5	Menú.....	65
9.2	Problemas y desviaciones.	65
9.2.1	Cross Origin Resource Sharing (CORS)	65
9.2.2	HOTLINK.....	66
9.2.3	Reestructuración de las pantallas	66
9.2.4	Retrasos de las entregas	67
10	Conclusiones.....	69
	Bibliografía	71

Resumen

Este documento corresponde a la memoria del trabajo de fin de grado “Diseño e implementación de una red social universitaria”, desarrollado para la titulación de Grado en Ingeniería en tecnología de telecomunicación en la especialidad de telemática, en la escuela de ingenieros de Bilbao de la Universidad del País Vasco, UPV/EHU.

El proyecto se ha desarrollado para la empresa Adoswe. Adoswe es una empresa emergente cuya misión es proveer de recursos a los estudiantes universitarios de forma que puedan adquirir experiencia y habilidades en el entorno laboral. Principalmente, Adoswe plantea una plataforma con la cual pretende crear y fomentar las relaciones laborales a través de la propuesta de retos y proyectos. El objetivo de este proyecto ha sido desarrollar una aplicación móvil de esta plataforma con el fin de facilitar a los estudiantes universitarios el contacto con el mundo laboral.

Para cumplir con este objetivo, se ha hecho uso de la API que se ha desarrollado en paralelo a este proyecto por Adoswe. Mediante esta API, se ha desarrollado una aplicación móvil multiplataforma que ofrece las funcionalidades básicas más importantes de la plataforma. Antes de realizar la implementación de la aplicación, ha sido necesario decidir qué tecnologías son óptimas para realizar el desarrollo, así como planificar cada una de las tareas, analizar los riesgos y preparar un presupuesto. Por último, se ha realizado el diseño de la aplicación junto con un plan de pruebas.

En este documento se puede encontrar todo el proceso seguido para desarrollar la aplicación móvil.

Laburpena

Dokumentu honek "Unibertsitateko sare sozial baten diseinua eta ezarpena" graduko amaiera lanaren memoriari dagokio, Telekomunikazio Teknologiaren Ingeniaritzako Graduan telematika espezialitatean garatu dena, UPV / EHUko Bilboko Ingeniaritza Eskolan.

Proiektu hau Adoswe enpresarentzat garatu da. Adoswe enpresa berri bat da, bere helburua unibertsitateko ikasleek lan esperientzia lortu ahal izateko baliabideak eskaintzea da. Helburu hau lortu ahal izateko, Adoswe-k plataforma bat proposatzen du. Plataforma honekin lan-harremanak sortzea eta sustatzea nahi du, erronka eta proiektuen proposamenaren bidez. Proiektuaren helburua, unibertsitateko ikasleei lan munduan sartzeko prozesua errazten duen plataforma baten aplikazioa garatzea da.

Helburu hau lortzeko, Adoswe-k proiektu horrekin batera garatutako APIa erabili da. API honen bidez, plataformaren funtzionalitate garrantzitsuenak eskaintzen dituen aplikazio multiplataforma bat garatu da. Aplikazioa gauzatu baino lehen, beharrezkoa izan da erabakitzea teknologia optimoenak, baita zeregin bakoitza planifikatzea, arriskuak aztertzea eta aurrekontua prestatzea. Amaitzeko, aplikazioaren diseinua proba plan batekin batera egin da.

Dokumentu honetan aplikazioa garatzeko jarraitu den prozesua aurki dezakezu.

Abstract

This document corresponds to the undergraduate thesis "Design and implementation of a university social network", developed for the Degree in Telecommunications Technology Engineering in the specialty of telematics, in the School of Engineering of Bilbao in the UPV / EHU University.

This project has been developed for the company Adoswe. Adoswe is a startup whose mission is to provide resources to university students so they can gain work experience. Adoswe proposes a platform with which it aims to create and promote labor relations through the proposal of challenges and projects. The objective of the project has been to develop a mobile application of a platform whose purpose is to provide university students an easier way of entering the labor force.

To fulfill this objective, an API has been developed alongside this project by Adoswe. Through this API, a multiplatform mobile application has been developed, offering the most important functionalities of the platform. Before implementing the application, it was necessary to decide which technologies are optimal to carry out the development, as well as planning each of the tasks, analyzing the risks and preparing a budget. Finally, the design of the application was done together with a test plan.

In this document you will find the process followed to develop the mobile application.

TABLA 1: EVALUACIÓN DE LAS ALTERNATIVAS PARA TIPO DE APLICACIÓN	22
TABLA 2: EVALUACIÓN DE LAS ALTERNATIVAS PARA FRAMEWORK.....	25
TABLA 3: EVALUACIÓN DE LAS ALTERNATIVAS PARA FRAMEWORK SOBRE CORDOVA.....	26
TABLA 4: DESCRIPCIÓN Y PIR DE LOS RIESGOS DEL PROYECTO.....	32
TABLA 5: TAREAS PARA EL DESARROLLO DEL PROYECTO	49
TABLA 6: PLANTILLA EJEMPLO DE SEGUIMIENTO DE TAREAS.....	52
TABLA 7: ESTADOS UTILIZADOS EN EL SEGUIMIENTO DE LAS TAREAS.....	53
TABLA 8: PARTIDA DE HORAS INTERNAS	58
TABLA 9: COSTE LICENCIAS DE SOFTWARE.....	59
TABLA 10: PARTIDA DE AMORTIZACIONES DEL PROYECTO	60
TABLA 11: TOTAL DEL PRESUPUESTO DEL PROYECTO.....	60

ILUSTRACIÓN 1: FASES DEL MODELO EN V [20]	34
ILUSTRACIÓN 2: EJEMPLO DE PROYECTO DESARROLLADO CON GIT-FLOW [22]	37
ILUSTRACIÓN 3: PATRÓN MVC [23]	38
ILUSTRACIÓN 4: MVC APLICADO AL CÓDIGO DE LA APLICACIÓN	39
ILUSTRACIÓN 5: DISEÑO MODULAR DE LA APLICACIÓN	40
ILUSTRACIÓN 6: SERVICIOS DE LA APLICACIÓN.....	41
ILUSTRACIÓN 7: DISEÑO DEL SISTEMA COMPLETO	42
ILUSTRACIÓN 8: PARTE I DEL GANTT.....	50
ILUSTRACIÓN 9: PARTE II DEL GANTT	51
ILUSTRACIÓN 10: PANTALLA DE LOGIN	61
ILUSTRACIÓN 11: PANTALLA PRINCIPAL DE MI ESPACIO	62
ILUSTRACIÓN 12: PANTALLA DE INFORMACIÓN Y FASES DE MI ESPACIO.....	62
ILUSTRACIÓN 13: PANTALLAS DE GRUPOS DE MI ESPACIO	63
ILUSTRACIÓN 14: PANTALLAS DE CREACIÓN E INTERACCIÓN DE FOROS	63
ILUSTRACIÓN 15: PANTALLAS DE EXPLORA.....	64
ILUSTRACIÓN 16: PANTALLA DE PERFIL.....	64
ILUSTRACIÓN 17: MENÚ DE LA APLICACIÓN	65

Abreviaturas

ABR: Aprendizaje Basado en Retos

API: Application Programming Interface

CORS: Cross Origin Resources Sharing

DCS: Dossier de Calidad Software

DDA: Documento de Diseño de Arquitectura

DDD: Documento de Diseño Detallado

DGCS: Documento de Gestión de Configuración SW

DPIS: Dossier de Pruebas de Integración Software

DPUS: Dossier de Pruebas Unitarias Software

DPVS: Dossier de Pruebas de Validación Software

GUI: Graphical User Interface ¹

MU: Manual de Usuario

MVC: Modelo Vista Controlador

PCS: Plan de Calidad Software

PDS: Plan de Desarrollo Software

PGCS: Plan de Gestión de Configuración Software

PIS: Plan de Integración Software

PVS: Plan de Validación Software

TFG: Trabajo de Fin de Grado

TS: TypeScript

VUCA: Volatilidad, Incertidumbre (uncertainty), Complejidad y Ambigüedad

¹ Interfaz gráfica de usuario

1 Introducción

Actualmente el mercado de las aplicaciones móviles está en auge. Cada vez existen más aplicaciones móviles que son imprescindibles para nuestro día a día. Este avance de la tecnología nos permite poder acceder a una infinidad de recursos haciendo uso de un dispositivo móvil y acceso a internet. De esta forma y en cierta manera, se consigue facilitar la vida de las personas dotándolas de mayor capacidad para poder desarrollar las actividades necesarias en su día a día.

Por otro lado, si observamos a los estudiantes que están en el proceso de búsqueda de un primer empleo, existe una gran deficiencia en plataformas que gestionen y faciliten este proceso. Es por eso que Adoswe, surge con el fin de solventar la necesidad existente en esta área. Mediante este Trabajo de Fin de Grado (TFG) se pretende extender la plataforma de gestión de Adoswe de la cual se hablará más adelante, a una aplicación móvil.

A lo largo del siguiente documento, se presenta el proceso de desarrollo seguido para llevar a cabo esta aplicación.

Primeramente, se detallan los resultados del análisis de los objetivos que se pretenden cumplir con este proyecto, así como las especificaciones que se han de cumplir para que la aplicación sea completa y funcional.

Seguidamente se presenta el estudio de las diferentes alternativas estudiadas con el fin de elegir las mejores tecnologías y herramientas para que el desarrollo del proyecto vaya acorde con las especificaciones marcadas.

Una vez decididas las tecnologías y recursos que más se adaptan a las especificaciones, se ha hecho un análisis de los riesgos que pueden existir a la hora de desarrollar este proyecto y los planes que pueden impedir o remediar que estos riesgos destruyan el proyecto. También se puede encontrar en este documento la metodología que se ha seguido a la hora de desarrollar el proyecto junto con un resumen del diseño de la solución que se expande con más detalle en los anexos correspondientes al diseño.

Por último, se encuentra la planificación del proyecto junto con el presupuesto del mismo y los resultados obtenidos a fecha de finalización del proyecto.

1.1 Contexto

Este proyecto tiene como fin desarrollar una aplicación para la empresa Adoswe. Adoswe es una empresa situada en el Impact Hub de Donostia. El Impact Hub pertenece a una red internacional de emprendedores, activistas, creativos y profesionales que comparten la visión de generar un impacto positivo en la sociedad a través de proyectos colaborativos. La visión de Adoswe es plantear una metodología que sistematiza la innovación abierta y el aprendizaje basado en retos (ABR) en la gestión académica. Todo ello a través de una plataforma virtual que gestiona la publicación de retos y proyectos emprendedores capacitando y visibilizando el talento diferencial de estudiantes con habilidades de interés para las empresas.

La realidad actual del mercado Volátil, Incierto, Complejo y Ambiguo (VUCA), demanda cada vez más a las empresas la gestión de la innovación a través de diferentes procesos caracterizados por ser multidisciplinarios, colaborativos y eficientes. Esta innovación abre las puertas a disponer de ciertas ventajas competitivas sobre la competencia. Uno de los problemas de esta innovación dinámica es que es un proceso que puede resultar caro y disminuye el ciclo de vida del producto.

Por otro lado, en el contexto empresarial en el que vivimos actualmente, las empresas se encuentran con ciertas dificultades para establecer con rapidez la adaptación de sus productos a las necesidades del cliente, y encontrar trabajadores con el perfil adecuado a las nuevas exigencias del mercado.

Estas condiciones en las que nos encontramos nos llevan a pensar en la necesidad existente de planes para reclutar y detectar candidatos con habilidades y competencias suficientes para desempeñar las tareas de estos nuevos retos. Existe un problema y es que los tiempos de contratación de las empresas hoy en día no coinciden con este nuevo planteamiento. Es por eso que este mismo mercado está estableciendo nuevas condiciones laborales, y fomentando tendencias de contratación por proyectos de forma independiente,

así como el deseo de muchos profesionales de trabajar de forma independiente y tener más control sobre sus carreras, por ejemplo, mediante la definición de sus propios términos de compromiso.

Frente a todo este planteamiento nos encontramos con Adoswe, la cual plantea un nuevo modelo de negocio de consultoría de innovación colaborativa, basada en relaciones cliente-consultor de "Management as a service". [1]

2 Objetivos

Siempre que empezamos un proyecto se debe pensar en algo central para obtener mejores resultados, normalmente es lo que se conoce como objetivos. Los objetivos, por lo tanto, son aquellas razones por las que llevamos a cabo el trabajo. La importancia de estos objetivos reside en el hecho de que nos han permitido ordenarnos mejor a la hora de trabajar, actuar, qué resultados buscar, etc.

2.1 Objetivos del proyecto (alcance)

2.1.1 Objetivo Principal

El objetivo principal del proyecto es desarrollar una aplicación móvil multiplataforma que ofrezca las funcionalidades básicas de la plataforma de gestión de la empresa Adoswe. La finalidad es crear una aplicación sencilla de usar a la vez que atractiva para los diferentes usuarios de la plataforma.

2.1.2 Objetivo secundario

Como objetivo secundario, se quiere facilitar y fomentar el uso de la plataforma entre los universitarios y empresas, ofreciéndoles la posibilidad de manejarla con sus teléfonos móviles.

2.2 Objetivos personales

Como se ha explicado, es muy importante marcar unos objetivos del proyecto, de la misma forma es importante definir qué metas se desea conseguir con un

proyecto a nivel personal. Para este proyecto se han definido los siguientes objetivos personales.

- Poner en práctica lo aprendido en el grado de Ingeniería en Tecnología de Telecomunicación, tanto en el área de desarrollo de proyectos como ingeniero, como en el desarrollo de software. Mediante la práctica se quiere obtener conocimiento acerca del coste y necesidades de un proyecto real.
- Desarrollar nuevas competencias en el área de desarrollo de proyectos, como la gestión del tiempo, planificación de recursos, establecimiento de planes de contingencia...
- Desarrollar una aplicación completamente funcional siendo posible sacarla al mercado.
- Ampliar los conocimientos en el área de desarrollo de software tanto en la parte de diseño como en la parte de codificación.
- Trabajar en un proyecto con aspectos reales, como por ejemplo la comunicación con proveedores o la relación con otras empresas.

2.3 Especificaciones

Como especificaciones tenemos las siguientes para el proyecto.

- Se debe desarrollar una aplicación para las plataformas móviles disponibles (Android e iOS).
- La aplicación móvil debe reflejar el contenido de la plataforma Adoswe. Los datos se obtendrán a partir de la API que se ha desarrollado en paralelo al proyecto por la propia empresa.

Para la aplicación se definen las siguientes especificaciones:

- La aplicación debe ser sencilla de manejar.
- La aplicación debe tener una GUI moderna y atractiva.

2.3.1 Autenticación

- La aplicación debe incorporar la opción de autenticarse, así como de registrarse en caso de que el usuario no se encuentre previamente ya registrado.
- La autenticación se realizará mediante el correo electrónico y una contraseña definida en el proceso de registro.

2.3.2 Explora

- La aplicación debe contener un espacio donde visualizar cada uno de los retos y proyectos registrados en la plataforma. Este espacio será común para todos los usuarios de la plataforma.
- El contenido de este espacio debe dividirse en 3 categorías diferentes:
 - Retos: Se mostrarán los retos cuya recompensa sea una compensación económica.
 - Retos + oferta: Se mostrarán los retos cuya recompensa sea una oferta de trabajo.
 - Proyectos: Se mostrarán los proyectos.
- El espacio denominado explora debe incorporar filtros determinados para poder establecer diferentes búsquedas de cada reto o proyecto.
- El espacio así mismo debe permitir la búsqueda a partir de una cadena introducida por el usuario de un reto o proyecto de la plataforma.
- Al seleccionar uno de los retos o proyectos deberá de mostrarse la información relativa al mismo. Esta información será:
 - Título del reto/proyecto.
 - Idiomas en los que está disponible.
 - Descripción del reto/proyecto.
 - Problemática que plantea resolver.
 - Entregables del reto/proyecto.

2.3.3 Espacio personal

- La aplicación debe incorporar un espacio donde cada usuario puede visualizar los retos y proyectos en los que está participando.

- Al seleccionar uno de los proyectos o retos en los cuales se está participando se deberá mostrar un espacio con los siguientes elementos:
 - Información del reto o proyecto en cuestión.
 - La información mostrada será la misma que la mostrada en la pantalla de explora.
 - Participantes del reto o proyecto. Los datos que se visualizarán serán:
 - Imagen estilo avatar de cada participante
 - Nombre completo de cada participante
 - Documentos que el grupo ha subido al reto o proyecto. Debe existir la posibilidad de visualizar dichos documentos. Se presentarán los siguientes datos:
 - Imagen representativa del tipo de documento.
 - Nombre del documento.
 - Autor del documento.
 - Foros y temas abiertos dentro del reto o proyecto. Se debe poder crear nuevos foros y responder a temas ya existentes dentro del proyecto o reto. Se visualizará un listado de los foros disponibles presentando:
 - El título del foro.
 - Última publicación del foro y la fecha.
 - Fases y tareas del reto o proyecto. Se deberá presentar:
 - Fecha de inicio y fin de la tarea.
 - Barra de progreso de la tarea.
 - Encargados de la tarea.
 - Avatar del usuario.
 - Título y descripción de la tarea.

2.3.4 Perfil del usuario

- La aplicación tendrá que disponer de un espacio en el que el usuario pueda visualizar y editar la información de su perfil. La información del perfil de usuario incluye:
 - Nombre del usuario

- Dirección de correo electrónico con la cual está registrado en la plataforma.
- Tipo de usuario.
- Imagen estilo avatar del usuario. En caso de que el usuario no disponga de imagen se autogenerará a partir de la primera letra del nombre.
- Teléfono del usuario.
- Dirección del usuario.

2.3.5 Mensajería

- Se debe poder enviar un mensaje a los participantes de un reto o proyecto de un mismo grupo.
- Se debe poder visualizar todos los mensajes de la bandeja de entrada.
- Existirá la opción de marcar como leído un mensaje.

2.3.6 Notificaciones

- La aplicación debe permitir visualizar las notificaciones de la plataforma.
- Las notificaciones se dividirán en distintas clases con distinta importancia. Se debe representar el grado de importancia de la notificación.

2.3.7 Frequently Asked Questions (FAQ)

- La aplicación debe incorporar un espacio donde se visualice información acerca de Adoswe.
- Esta información será un texto estático proporcionado por Adoswe.

3 Beneficios

Una vez se han definido los objetivos y especificaciones, para valorar el impacto del proyecto un atributo importante a estudiar son los beneficios que este aporta al ser realizado.

Para evaluar este impacto se han a estudiado los beneficios en varios ámbitos.

3.1 Beneficios Técnicos

Los beneficios técnicos que aporta este proyecto están adheridos a la definición del proyecto en sí, una aplicación que extiende el uso de la plataforma de gestión Adoswe.

El valor añadido que tiene esta aplicación, es que se puede utilizar en cualquier dispositivo, en cualquier lugar y de manera muy sencilla.

3.2 Beneficios Económicos

Como se ha explicado, este proyecto tiene como fin desarrollar una aplicación móvil que sirva como extensión de la plataforma de gestión de la empresa Adoswe. Con esta extensión se pretende conseguir que el uso de la plataforma sea más sencillo para los usuarios. Además, el tener la plataforma en un móvil, siendo el uso de los móviles tan alto, proporciona un uso más activo y extendido de la misma.

Teniendo en cuenta estas ventajas que aporta la aplicación frente a la plataforma web ya existente, al fomentarse el uso de la plataforma, aumentará el número de usuarios que participan en este sistema de gestión. De manera directamente proporcional al número de usuarios, aumentará el número de empresas que participan en la creación de retos, proyectos y ofertas de trabajo. Este aumento de actividad laboral por lo tanto influye directamente no solo en la situación económica de los individuos que participan en el sistema, también en la creación de empleo a nivel estatal aportando valor al mercado de la comunidad y/o país.

Con esta aplicación se consigue aumentar el porcentaje de éxito y disminuir el tiempo necesario en la obtención de empleo. Si nos centramos en participantes

como las universidades o centros de formación, podemos ver cómo el aumento de usuarios de esta plataforma influye directamente, ya que aumentará tanto la valoración externa de la entidad, como la cantidad de alumnos que desean realizar sus estudios en el centro.

3.3 Beneficios sociales

De la misma manera el desarrollo de una aplicación móvil que facilite y fomente el uso de esta plataforma aporta varios beneficios sociales.

Por un lado, fomenta las relaciones sociales entre los participantes de los diferentes retos o proyectos. Mejorando su habilidad de comprensión y escucha activa.

Por otro lado, se fomenta la relación personal, laboral y académica entre personas con los mismos intereses.

Mediante este proyecto se ofrece una gran oportunidad para que los futuros usuarios puedan desarrollar una red de contactos en el mundo laboral e interpersonal.

Por último, mediante el uso de la aplicación se fomenta el aprendizaje, la originalidad y creatividad de los individuos, así como el trabajo en equipo.

4 Análisis de alternativas

Para realizar el desarrollo de una aplicación existen diferentes posibilidades, cada una con sus ventajas e inconvenientes. A continuación, se presenta el análisis de cada una de las opciones existentes con el fin de obtener la alternativa que mejor se adapta a las necesidades de la aplicación. El primer paso de decisión se corresponde con la tecnología a emplear en el desarrollo de la aplicación. El análisis se ha realizado entre estos tres tipos de aplicaciones: aplicaciones nativas, aplicaciones web y aplicaciones híbridas.

Para realizar un análisis exitoso y poder seleccionar la opción que más se adapta a las necesidades, es necesario identificar los requerimientos o características con las que debe de cumplir la aplicación. Se han estudiado los siguientes aspectos: portabilidad, rendimiento, accesibilidad, tiempo de desarrollo, coste y aspecto.

4.1 Análisis de tipo de aplicación

4.1.1 Aplicación Nativa

Como primera opción para desarrollar una aplicación móvil tenemos la opción de desarrollarla de forma nativa. Una aplicación nativa es una app que se desarrolla para cada plataforma específicamente (Android, iOS, Windows) en el lenguaje utilizado por el sistema operativo (SO) específico.

En lo relativo a las ventajas de este tipo de aplicaciones destacan varias. Primeramente, las aplicaciones nativas resaltan entre todas por ser las que mejor rendimiento ofrecen ya que el código está optimizado para la arquitectura, haciendo un uso más adecuado de los recursos, además de poder obtener un look & feel (aspecto) adaptado al SO sobre el que se esté desarrollando la aplicación. En cuanto a la accesibilidad, las aplicaciones nativas permiten a los desarrolladores acceder al conjunto completo de funciones de la plataforma [2] [3].

Por otro lado, las aplicaciones nativas suponen un coste mayor frente a aplicaciones web o híbridas [2]. Uno de los requisitos de la aplicación a

desarrollar es que sea multiplataforma, es decir, que la aplicación sea funcional en cualquiera de los SSOO existentes en el mercado. En el caso de desarrollar la app de forma nativa no sería multiplataforma, sino que sería necesario realizar el desarrollo de la misma aplicación para los diferentes SSOO.

4.1.2 Aplicación Web

Una aplicación web, es aquella alojada en un servidor web y que es accesible desde cualquier navegador web. Estas aplicaciones se desarrollan mediante el uso de las tecnologías web.

Las aplicaciones web destacan por ser multiplataforma y multidispositivo debido a que no es necesario compilarlas ni instalarlas en el dispositivo móvil ya que se alojan en un servidor web y se acceden a ellas mediante un navegador web. Esto nos genera un inconveniente y es que para poder acceder a la aplicación es necesario disponer de una conexión a internet a diferencia de una aplicación nativa o híbrida. En lo relativo al rendimiento, ofrecen un rendimiento inferior que las aplicaciones nativas, las aplicaciones web son más lentas, además las características o capacidades del dispositivo móvil accesibles desde una aplicación web son mucho más limitadas, por lo tanto una aplicación web está más limitada que sus competidoras. Si comparamos el tiempo de desarrollo y el coste, este tipo de aplicaciones se llevan el primer puesto ya que es más sencillo construir una aplicación web que una aplicación nativa o híbrida y el coste es menor.

4.1.3 Aplicación Híbrida

El último caso a analizar es el caso de las aplicaciones híbridas. Las aplicaciones híbridas, a diferencia de las aplicaciones web y como lo hacen las aplicaciones nativas, se ejecutan en el dispositivo móvil. Estas aplicaciones híbridas se escriben con tecnologías web, es decir, las aplicaciones híbridas se ejecutan dentro de un contenedor nativo y utilizan el motor del navegador (no el navegador web en sí) para renderizar el HTML y procesar JavaScript de forma local, lo cual es una ventaja frente a aplicaciones web ya que no es necesario una conexión a internet. Este tipo de aplicaciones no ofrece una accesibilidad a las capacidades del SO tan amplias como una aplicación nativa, pero sí que ofrece

más que una aplicación web, como puede ser acceso a la cámara, acelerómetro o almacenamiento local [2]. El rendimiento de este tipo de tecnología se puede situar en un punto intermedio entre sus dos oponentes. Dado que una aplicación híbrida se basa en tecnologías web, el tiempo de desarrollo, el coste y la portabilidad son muy similares a las de una aplicación web, siendo posible desarrollar un mismo código base que correrá nativamente en los diferentes SSOO en un tiempo y con un coste mucho menor que una aplicación nativa [2]. Por último, dentro de la aplicación híbrida existen diferentes tecnologías que permiten adaptar o customizar la vista (aspecto) de las aplicaciones resultantes para las diferentes plataformas. No obtenemos un poder de customización tan amplio como con una aplicación nativa, pero sí que se nos ofrece mayores posibilidades que una aplicación web.

4.1.4 Valoración y elección

Tras analizar las diferentes alternativas, se ha procedido a evaluar numéricamente y ponderando los resultados en función de la importancia de las características valoradas (siguiendo los requerimientos y necesidades de la aplicación) para obtener la solución que mejor se adapta.

	Portabilidad (30%)	Rendimiento (10%)	Accesibilidad (10%)	Tiempo (20%)	Coste (15%)	Aspecto (15%)	Total
Aplicación Nativa	1	3	3	1	1	3	1,70
Aplicación WEB	3	1	1	3	3	1	2,30
Aplicación Híbrida	3	2	2	3	2	2	2,50

Tabla 1: Evaluación de las alternativas para tipo de aplicación

Como resultado de esta evaluación, podemos extraer que la solución híbrida es la tecnología más adecuada para desarrollar la aplicación. La solución híbrida nos proporciona una gran portabilidad siendo posible desarrollar una única aplicación para los diferentes sistemas operativos. Además, esto lo podemos lograr en un tiempo y con un coste menor que la solución nativa. Por último, nos

proporciona un resultado de gran calidad en lo relativo al aspecto visual de la aplicación.

4.2 Análisis de los frameworks

Una vez que hemos determinado el tipo de aplicación que vamos a desarrollar, es necesario decidir la tecnología a emplear dentro de esta categoría. Se han analizado tres opciones diferentes: Flutter, Xamarin, Cordova.

Estas tecnologías se denominan frameworks. Un framework en el desarrollo software, es una estructura que dispone de componentes que son personalizables e intercambiables con el fin de desarrollar una aplicación. Por lo tanto, se puede entender un framework como una aplicación genérica a la que añadir o quitar piezas para construir la aplicación final. [4]

Para valorar estas tecnologías y elegir la más adecuada a este proyecto, se proponen los siguientes parámetros:

- *Documentación*: Se ha valorado positivamente aquellos frameworks que dispongan de mayor documentación y comunidad. Gracias a esto se podrá conseguir resolver posibles problemas con mayor facilidad y rapidez.
- *Facilidad de uso*: Se ha priorizado la elección de un framework con mayor facilidad de uso frente a otro.
- *Uso actual*: Se define como el uso que la tecnología está recibiendo actualmente por los desarrolladores, relacionado también con la documentación y comunidad disponible.
- *Aspecto*: La posibilidad de obtener una calidad visual mayor en el resultado final.
- *Conocimiento Inicial*: Se ha valorado positivamente aquellas tecnologías de las que se poseía un conocimiento previo o, dicho de otra manera, aquellas que requerían menos tiempo aprender a utilizarlas.

4.2.1 Flutter

Flutter es un framework desarrollado por Google que permite crear aplicaciones para iOS y Android a partir de un mismo código base. Es una buena opción en lo relativo a la posibilidad que ofrece para crear una interfaz de usuario (UI) muy atractiva y flexible a la vez que ofrece un rendimiento nativo adecuado. Sin embargo, este framework está en su primera versión lanzada en diciembre de 2018, lo que nos deja con una tecnología relativamente nueva con la que podríamos tener problemas en encontrar documentación y comunidad, enfrentándonos a la posibilidad de que se aumente el tiempo requerido para realizar el desarrollo [5]. Además, este framework utiliza un nuevo lenguaje de programación de Google denominado Dart lo cual implica la necesidad de tener que aprender dicho lenguaje. [6]

4.2.2 Xamarin

El framework Xamarin de Microsoft, utiliza el lenguaje C# [7]. Una ventaja de Xamarin frente a otros frameworks es que ofrece un rendimiento nativo muy alto y a diferencia de Flutter, es un framework muy usado con mayores recursos en cuanto a documentación y comunidad [8]. Como desventaja, destaca que el desarrollo de la interfaz gráfica consume mucho tiempo habiendo componentes que han de ser desarrollados para cada plataforma individual.

4.2.3 Cordova

Cordova, permite desarrollar aplicaciones móviles multiplataforma utilizando CSS, HTML y JavaScript [9]. Si bien es cierto que Cordova no ofrece un rendimiento nativo tan bueno como Xamarin o Flutter, permite la creación de aplicaciones multiplataforma de forma más sencilla [10]. Cordova utiliza tecnologías web que son más sencillas y rápidas de aprender que otros lenguajes de programación utilizados por otros frameworks. Además, mediante el uso de otros frameworks por encima de Cordova, es posible crear aplicaciones atractivas y adaptadas al SO [11]. Otra de las ventajas de Cordova es que tiene una gran comunidad con un nivel mayor de interacción y apoyo que el que puede tener Xamarin.

4.2.4 Valoración y elección

	Documentación (25%)	Facilidad de uso (30%)	Uso actual (10%)	Aspecto (10%)	Conocimiento Inicial (25%)	Total
Flutter	1	2	1	3	1	1,50
Xamarin	2	1	2	3	1	1,55
Cordova	3	3	3	2	2	2,65

Tabla 2: Evaluación de las alternativas para framework

En lo relativo al framework a utilizar, podemos observar en la tabla que el que más se adapta a este proyecto es el framework Cordova. Si bien es cierto que este framework no nos ofrece un aspecto de la aplicación del calibre del que nos ofrecen los otros frameworks, Cordova se encuentra entre los frameworks más utilizados y con mayor comunidad y recursos. Esta característica nos facilitará su uso y nos ahorrará tiempo en el desarrollo de la aplicación.

4.3 Análisis de tecnología a emplear sobre Cordova

Siguiendo con el análisis, vamos a evaluar que tecnología es más conveniente utilizar sobre el framework Cordova para llevar a cabo nuestra aplicación. Se han analizado 2 alternativas: jQuery Mobile e IONIC. Los aspectos a valorar en este análisis para las diferentes tecnologías han sido: aspecto, documentación disponible, madurez y actividad

4.3.1 JQuery Mobile

jQuery Mobile se puede considerar como un sistema de UI basado en HTML, diseñado para crear sitios web responsive y aplicaciones accesibles para todos los teléfonos inteligentes, tabletas y dispositivos de escritorio [12]. jQuery Mobile presenta ciertas desventajas, la primera es que a pesar de que existe una gran comunidad, la actividad actual no es tal como la que presenta la comunidad de IONIC. Por otro lado, la última versión de jQuery Mobile es la 1.5 lanzada en

2017 [13], es decir, jQuery Mobile lleva 2 años sin ser actualizado lo cual puede generar un problema y a su vez, representa la actividad de los desarrolladores. Como ventaja podemos destacar que es un framework ligero y fácil de utilizar [14].

4.3.2 IONIC

Ionic es una herramienta gratuita y open source para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS [15]. IONIC ofrece la posibilidad de desarrollar aplicaciones integradas con las funcionalidades nativas de los dispositivos a través de los diferentes plugins de Cordova existentes [16]. IONIC cuenta con una amplia y activa comunidad de usuarios que colaboran en dar solución a las cuestiones y problemas que surgen, esto corresponde una ventaja frente a jQuery Mobile ya que posibilita la opción de encontrar solución de una manera más rápida a los posibles problemas que surjan en el proceso de desarrollo. Ofrece muy buenos resultados a nivel visual y una gran facilidad para obtenerlos gracias a que IONIC define una serie de componentes para la UI de alta calidad y modernos.

4.3.3 Valoración y elección

	Documentación (30%)	Madurez (20%)	Actividad (30%)	Aspecto (30%)	Total
jQuery Mobile	2	2	1	1	1,10
IONIC	2	2	2	2	2,00

Tabla 3: Evaluación de las alternativas para framework sobre Cordova

Teniendo en cuenta las diferentes opciones analizadas, se ha decidido utilizar IONIC sobre Cordova para desarrollar la aplicación móvil. Se ha elegido IONIC ya que se puede obtener un resultado más atractivo hacia el usuario con mayor facilidad. Además, toma ventaja frente a la alternativa valorada en el uso que está recibiendo en la actualidad, la actividad de la comunidad y los recursos que se pueden encontrar como documentación o foros.

4.4 Análisis de versión de IONIC

Existen diferentes versiones de IONIC, actualmente está en su cuarta versión. A continuación, se han analizado las diferentes versiones para determinar cual es la más adecuada para desarrollar la aplicación.

4.4.1 V2, V3 frente a la V4

Para evaluar que versión tenemos que usar, se han analizado que mejoras introduce la nueva versión frente a las anteriores.

La versión 4 de IONIC es la más actual. En esta versión, IONIC utiliza las Web APIs y cada componente está empaquetado como un componente web. Los componentes web son un conjunto de APIs que permiten crear nuevas etiquetas HTML personalizadas, reutilizables y encapsuladas para usar en páginas web y aplicaciones web [17]. De esta manera, se consiguen importantes mejoras en el rendimiento y en tiempos de carga [18] de las aplicaciones modernas. Estos componentes no solo se distinguen de la versión anterior por la mejora del rendimiento si no por el aspecto más moderno y estilizado. La versión 4 se diferencia de versiones anteriores por ofrecer compatibilidad con otros frameworks. A su vez, IONIC 4 incorpora ciertas mejoras de la interfaz de línea de comandos [19].

Por otro lado, la versión 3 lleva más tiempo usándose por lo que se puede encontrar más documentación, sin embargo, debido a la alta actividad tanto de la comunidad como de los propios desarrolladores los recursos de la versión 4 se extienden de forma muy rápida.

4.4.2 Valoración y elección

Visto que la versión 4 de IONIC ofrece ciertas mejoras frente a las versiones anteriores y, además, es posible encontrar recursos y documentación de manera fácil que contribuyen en la solución de posibles problemas, se ha decidido utilizar IONIC V4.

5 Análisis de riesgos

A la hora de planificar un proyecto es necesario identificar los riesgos que pueden surgir a lo largo del proceso ya que de esta forma se puede crear un plan para solventar la problemática derivada de los posibles riesgos.

Para identificar la severidad de las medidas a tomar se ha hecho una evaluación de cada riesgo. Tanto para evitar que sucedan como para mitigar su impacto en el caso de que sí sucedan.

A cada riesgo identificado se le asignó:

- Probabilidad: 1 a 10, siendo 1 la más baja y 10 la más alta.
- Impacto: 1 a 10, siendo 1 la más baja y 10 la más alta.

La prioridad e importancia del riesgo (PIR) se calcula mediante la fórmula:

$$PIR = Probabilidad \times Impacto$$

En función del resultado obtenido, se le ha asignado a cada riesgo un color identificativo:

- Si $PIR > 40$ identificador **ROJO**
- Si $40 \geq PIR > 16$ identificador **NARANJA**
- Si $16 \geq PIR \geq 1$ identificador **VERDE**

Tras este análisis se identificaron los siguientes riesgos más relevantes para el proyecto.

5.1 Requerimientos incompletos

Uno de los riesgos más altos, es tener los requerimientos incompletos del proyecto o tener requerimientos ambiguos.

Causa

Las posibles causas de este riesgo son:

- No existe una idea clara de lo que se quiere conseguir.

- No se definen los requerimientos de manera clara y completa.
- Comprensión errónea de los requerimientos.

Medida de prevención

Para evitar este riesgo se han llevado a cabo las siguientes acciones:

- Realizar varias reuniones con el fin de definir bien los requerimientos.
- Crear un listado de preguntas sobre temas que no hayan quedado claros en reuniones anteriores.

5.2 Planificación incorrecta

Uno de los riesgos más evidentes del proyecto es la planificación incorrecta del mismo. La planificación incorrecta se refiere a realizar un reparto de recursos y de tiempo erróneo para cada fase y tarea del proyecto.

Causa

Las posibles causas de este riesgo son las siguientes:

- Falta de experiencia en la planificación de proyectos.
- Desconocimiento de requerimientos totales.

Medida de prevención

Para prevenir una incorrecta planificación se han llevado a cabo las siguientes acciones. Primeramente, se han estudiado otros proyectos de requerimientos y alcance similares. De esta forma, se han podido asignar unos tiempos y recursos a las tareas de forma coherente y adaptada a la realidad. Además, se propuso establecer, concretar y cerrar los requerimientos del proyecto antes de realizar cualquier proceso de planificación del mismo. Por último, se ha revisado y validado la planificación por el director para poder cubrir cualquier imprevisto que pueda surgir.

Se ha asignado a cada fase del proyecto tiempos de respaldo por si ocurriera que la planificación prevista no es correcta. Es decir, entre fases del proyecto existe holgura, de esta forma se evitarán daños mayores en caso de una mala planificación.

5.3 Presupuesto limitado

El presupuesto es un posible riesgo a tener en cuenta en el proyecto, ya que la mala realización de un presupuesto nos puede llevar a perder dinero, tiempo y recursos, además de no poder finalizar el proyecto.

Causa

Una de las causas que puede ocasionar este riesgo es la falta de fuente de ingresos. Además de otras causas como:

- No contemplar imprevistos en el proyecto.
- Dimensionamiento erróneo del presupuesto.

Medida de prevención

Para evitar este riesgo, una vez finalizado el presupuesto, se ha revisado y se ha añadido un 10% del total para posibles imprevistos. Además, se ha tratado en la medida de lo posible de utilizar recursos libres, de esta forma se ha podido reducir el coste total del proyecto.

5.4 Pérdida de producto desarrollado

En este proyecto existe el riesgo de pérdida de producto desarrollado. El producto desarrollado puede ser: bases de datos, módulos de la aplicación...

Causa

Las posibles causas de este riesgo son:

- Fallo de la máquina.
- Fallo del IDE.
- Accidente: Borrado manual de ficheros, archivos, proyectos...
- No realizar el guardado después de añadir código.

Medida de prevención

Para prevenir este riesgo, se han llevado a cabo varias medidas. Por un lado, se ha marcado como obligatorio el uso de un sistema de control de versiones. Por

otro lado, se ha establecido un protocolo de guardado y de backup². Para evitar la pérdida de datos, el protocolo define que es necesario tener todos los componentes al menos en:

- Máquina donde se está desarrollando el producto.
- Disco duro externo.
- Repositorio remoto del proyecto.

En la siguiente tabla se recogen todos los riesgos identificados para el proyecto.

Código	Descripción	Probabilidad	Impacto	PIR
R-001	Planificación incorrecta	7	9	63
R-002	No disponibilidad de software	6	8	48
R-003	Requerimientos incompletos o ambiguos	5	9	45
R-004	Falta de conocimiento y experiencia	6	5	30
R-005	Subestimación del tamaño de la aplicación	4	7	28
R-006	Tardanza en la reparación de defectos	4	7	28
R-007	Falta de especificación de la arquitectura lógica	3	9	27

² Termina que se refiere a una **copia de seguridad**.

R-008	Diseño de interfaces incompleto	3	8	24
R-009	Desconocimiento de la lógica de negocio	3	8	24
R-010	Pruebas parciales o no son completas	3	8	24
R-011	Presupuesto limitado	3	7	21
R-012	Perdida de producto desarrollado	2	10	20
R-013	Software a implementar complejo	2	8	16
R-014	Recursos limitados	2	8	16
R-015	Integración del software compleja	2	7	14
R-016	Falta de documentación de código fuente	3	3	9

Tabla 4: Descripción y PIR de los riesgos del proyecto

6 Descripción de la solución

A la hora de desarrollar software, es muy importante decidir qué metodología de desarrollo de software se va a emplear. Existen diferentes metodologías a seguir a la hora de desarrollar software, cada una con sus ventajas y desventajas. Para este proyecto, se ha decidido utilizar la metodología de desarrollo en “V”.

6.1 Modelo en V

El modelo en V se caracteriza por ser una metodología de software en la que cada una de las fases del ciclo de vida del proyecto sirve para realimentar otras fases y de esta forma poder crear un producto software robusto y completo.

En el modelo en V existe una simetría entre procesos de producción y de verificación. Se fundamenta en el principio de triangulación que se aplica entre las fases de diseño y prueba de producto. Este principio nos permite retroceder a una fase de diseño después de haber realizado determinadas pruebas y observar las carencias y problemáticas que existen en nuestro producto. De la misma forma la documentación se genera antes y durante cada fase, no se deja para el final [20].

A continuación, se muestra una imagen con el esquema de fases en las que se divide el modelo en V para el desarrollo de software. En este esquema se observa claramente los principios anteriormente explicados. Además, para cada fase se observa la documentación que ha de ser generada durante la misma.

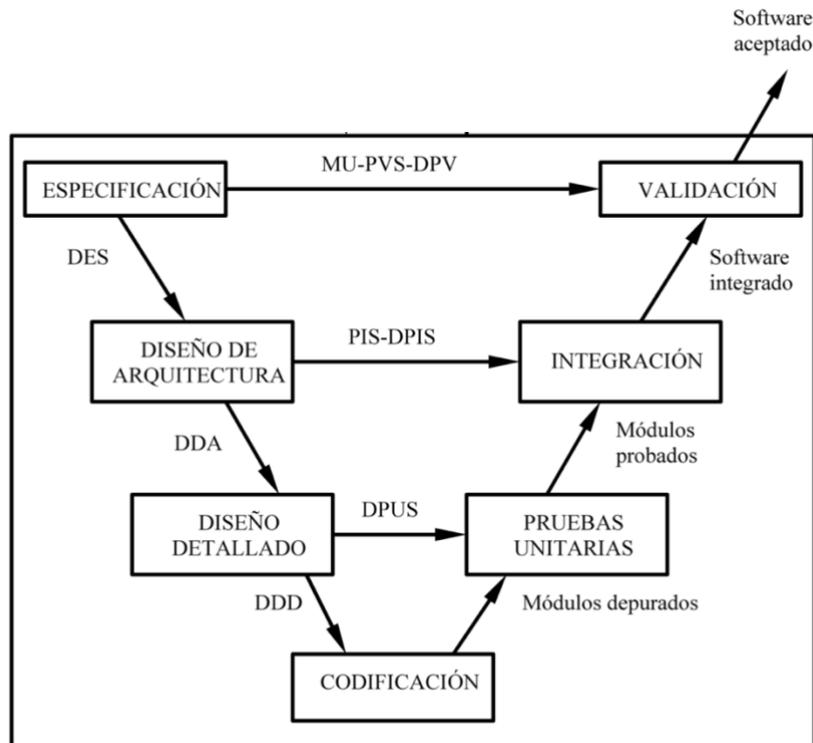


Ilustración 1: Fases del modelo en V [20]

Este esquema se puede dividir en diferentes secciones. Primeramente, tenemos los rectángulos que representan las fases. Las fases indican en que estado del proceso de desarrollo estamos. Por otro lado, tenemos las flechas que indican el flujo de las fases, es decir, como se relacionan las fases de un proyecto software entre sí. Por último, tenemos los documentos que se generan en cada fase y que se consumen en la fase siguiente.

- **Fase de especificación:** Esta fase comprende cada uno de los procesos y tareas relacionados con determinar que funcionalidades y características tiene que cumplir el producto software. A partir de esta fase se generan dos documentos:
 - Documento de Especificaciones Software (DES): Documento que recoge el resultado de la fase de especificación y que será consumido por la fase de diseño.
 - Manual de Usuario (MU): Documento que recoge las instrucciones de uso del producto.

- Dossier de Pruebas de Validación + Plan de Validación del software (DPV+PVS): Documento que recoge las pruebas que tiene que cumplir el producto para marcarlo como validado.
- **Fase de diseño:** Esta fase por lo general se divide en fases más detalladas. Por un lado, tenemos el diseño de la arquitectura o de alto nivel. Por otro lado, tenemos el diseño detallado o de bajo nivel. En estas fases se detalla cómo va a ser el diseño del producto software para que cumpla con las especificaciones acordadas en la fase anterior. Como resultado de esta fase se obtienen varios documentos:
 - Dossier de Diseño de la Arquitectura (DDA): Documento que se compone de los planos con el diseño de alto nivel del producto software. Los módulos que se van a utilizar, la relación entre ellos, etc.
 - Dossier del Diseño Detallado (DDD): Documento que corresponde al diseño de bajo nivel del producto.

De la misma forma se obtiene los planes de pruebas tanto unitarios como de integración que se consumirán en la parte de pruebas.

- Dossier de Pruebas Unitarias (DPU): Compuesto por las pruebas unitarias que tienen que cumplir cada módulo del producto.
- Dossier de Pruebas de Integración (DPI): Compuesto por las pruebas necesarias para comprobar que los módulos se relacionan según el diseño de alto nivel y por lo tanto cumplen con la especificación.
- **Fase de codificación:** En esta fase se hace uso de los documentos del diseño. Mediante las tecnologías necesarias, se materializa este diseño en un producto software inestable a falta de pruebas y de la integración.
- **Fase de pruebas + integración:** En esta fase a partir de los módulos software desarrollados en la fase de codificación se obtiene una versión finalizada del producto software completo. Valiéndose de los documentos de pruebas e integración se testea cada módulo y una vez que cumple con cada una de las pruebas definidas, se integran todos los módulos en un producto final.

- **Fase de validación:** Se comprueba que el producto final cumple con las especificaciones iniciales y que su uso está definido completamente en el MU creado junto con las especificaciones. Al final de esta fase tenemos un producto software completamente funcional y listo para entregárselo al cliente.

6.2 Control de versiones

Podemos definir dos etapas en la vida del software. Por un lado, tenemos la fase de desarrollo y por otro la fase de explotación. En ambas fases, el software sufre modificaciones. En la fase de desarrollo, el software se ve sometido a una evolución, por eso nos interesa que esta evolución sea controlada. En la fase de explotación, es necesario realizar mantenimiento al producto para mantener su estabilidad e incorporar mejoras. Para mantener esta evolución controlada y poder recuperar diferentes versiones del software es necesario el uso de un software de control de versiones.

Para realizar este control se ha hecho uso de Git. Git es un software de control de versiones distribuido de código abierto y gratuito, diseñado para manejar desde proyectos pequeños hasta proyectos muy grandes con rapidez y eficiencia [21]. Para hacer uso de esta herramienta se ha desarrollado el proyecto en un repositorio de GitHub. GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos usando el sistema de control de versiones Git.

Para hacer uso de esta herramienta se ha seguido el modelo Git-Flow. La metodología git-flow consiste en lo siguiente.

El trabajo se organiza en dos ramas principales:

- Rama **master**: cualquier commit que pongamos en esta rama debe estar preparado para subir a producción.
- Rama **develop**: rama en la que está el código que conformará la siguiente versión planificada del proyecto.

Cada vez que se incorpora código a la rama master, tenemos una nueva versión.

Además de estas dos ramas, Se proponen las siguientes ramas auxiliares:

6.2.1 Ramas Feature

Estas ramas se utilizan para desarrollar nuevas características de la aplicación que, una vez terminadas, se incorporan a la rama develop.

6.2.2 Ramas Release

Estas ramas se utilizan para preparar el siguiente código en producción. En estas ramas se hacen los últimos ajustes y se corrigen los últimos bugs antes de pasar el código a producción incorporándolo a la rama master.

6.2.3 Ramas Hotfix

Esas ramas se utilizan para corregir errores y bugs en el código en producción. Funcionan de forma parecida a las ramas release, siendo la principal diferencia que los hotfixes no se planifican.

A continuación, se muestra un ejemplo del modelo Git-Flow.

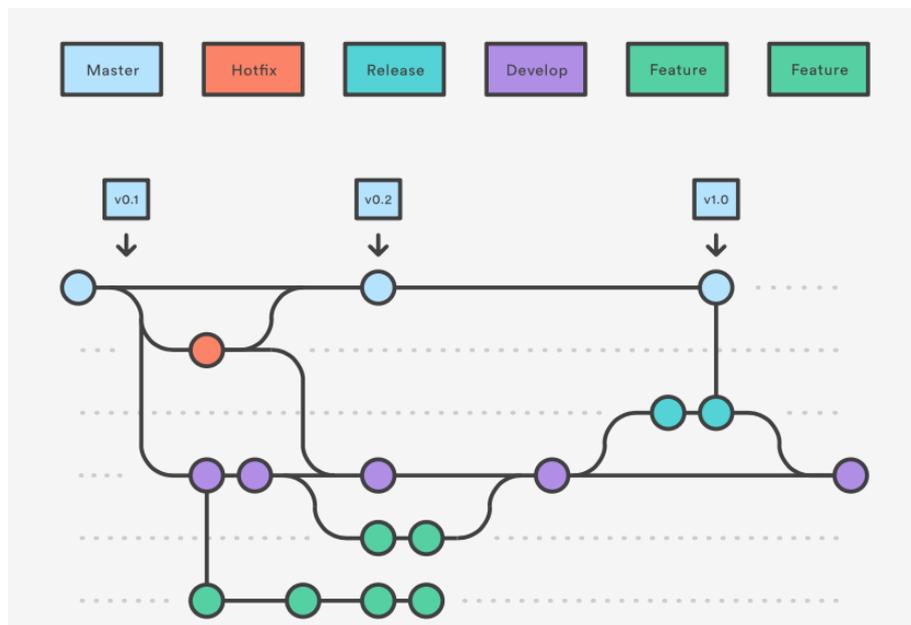


Ilustración 2: Ejemplo de proyecto desarrollado con Git-Flow [22]

6.3 Resumen del diseño

Para implementar la aplicación se ha seguido el patrón de diseño Modelo Vista Controlador (MVC). El framework que se ha utilizado para la aplicación permite implementar este diseño de forma sencilla debido a la distribución que utiliza

para componer las páginas de la aplicación. Dentro de este patrón de diseño podemos distinguir 3 partes:

- Modelo. Es el encargado de acceder a los datos.
- Vista. Es la parte encargada de representar los datos.
- Controlador. Se podría definir como el intermediario entre la capa de datos (el modelo) y la vista a través del usuario.

Esta imagen ilustra de manera muy clara las funcionalidades de cada parte y como interactúan entre ellas.

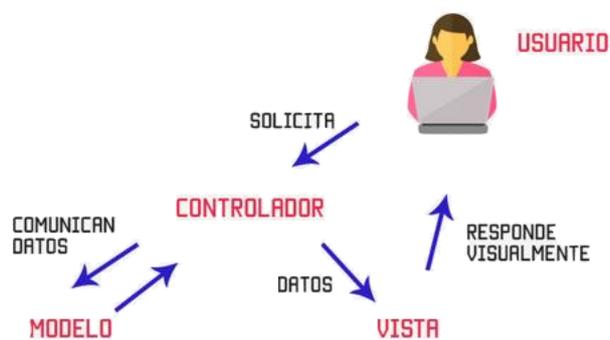


Ilustración 3: Patrón MVC [23]

En el esquema anterior las flechas indican como es el proceso de comunicación y la relación entre las partes del patrón MVC. El proceso completo se compone de un usuario interactuando con el controlador para solicitar las operaciones disponibles, que en este proyecto se corresponden con acciones como mandar un mensaje a un usuario, obtener los retos de un usuario determinado, etc. Posteriormente el controlador se comunica con el modelo con el fin de obtener o de entregar los datos necesarios para que la operación se complete. Una vez realizada la operación, el controlador le comunica los datos correspondientes a la vista para mostrárselos al usuario y finalizar la transacción.

Siguiendo este patrón de diseño nuestro código ha quedado distribuido de la siguiente manera.

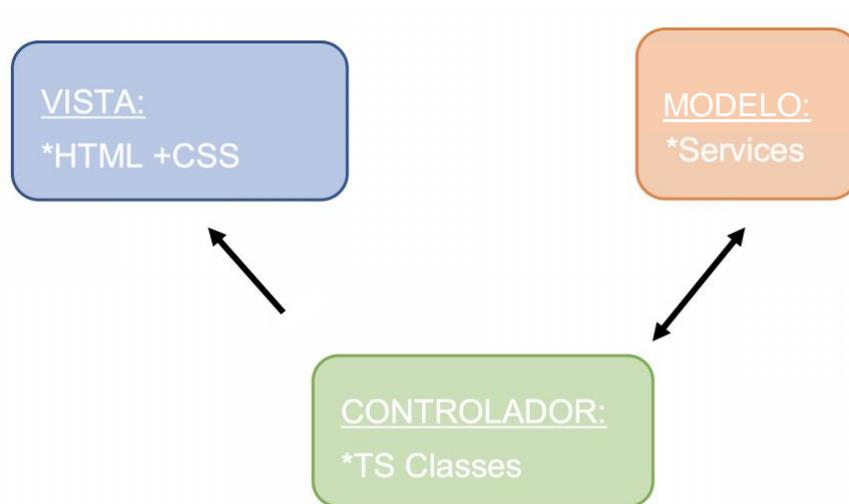


Ilustración 4: MVC aplicado al código de la aplicación

En este esquema las flechas indican la comunicación entre las diferentes partes, es decir, el flujo de datos aplicado a la distribución del código que se ha usado para desarrollar la aplicación. En este caso en particular y siguiendo el esquema anterior (Ilustración 3: Patrón MVC [23]), el usuario solicita una operación haciendo uso de las funciones definidas en las clases *typescript* (TS). Estas clases TS se comunican bidireccionalmente con el modelo para obtener los datos necesarios para que el controlador pueda mostrarle el resultado de la operación al usuario a través de la vista.

Las tecnologías que se han utilizado para desarrollar este tipo de diseño han sido las siguientes.

Por un lado, se ha implementado la vista haciendo uso de los *HTML* y *CSS* de cada página de la aplicación en particular. En este caso se ha hecho uso de las etiquetas y componentes que proporciona el framework *IONIC* para darle a esta vista un estilo más moderno y de mayor atractivo visual.

Por otro lado, tenemos el controlador. Para implementar la funcionalidad del controlador se ha hecho uso de las clases TS que albergarán tanto la comunicación con el modelo para obtener los datos necesarios que el usuario solicita, como la lógica necesaria para tratar estos datos y entregárselos a la vista.

Por último, para implementar el modelo se ha hecho uso de los servicios. Dado que los datos no se encuentran en una base de datos local, sino que se consiguen a través de la API de Adoswe, los diferentes servicios están compuestos de las peticiones HTTP a dicha API que obtienen los datos.

Para que la aplicación cumpla con las funcionalidades y especificaciones que se han detallado, se ha pensado el siguiente diseño.

En lo relativo a las vistas y el controlador, se ha diseñado la siguiente estructura. Cada recuadro representa una vista (HTML+CSS) y su correspondiente controlador (clase TS).

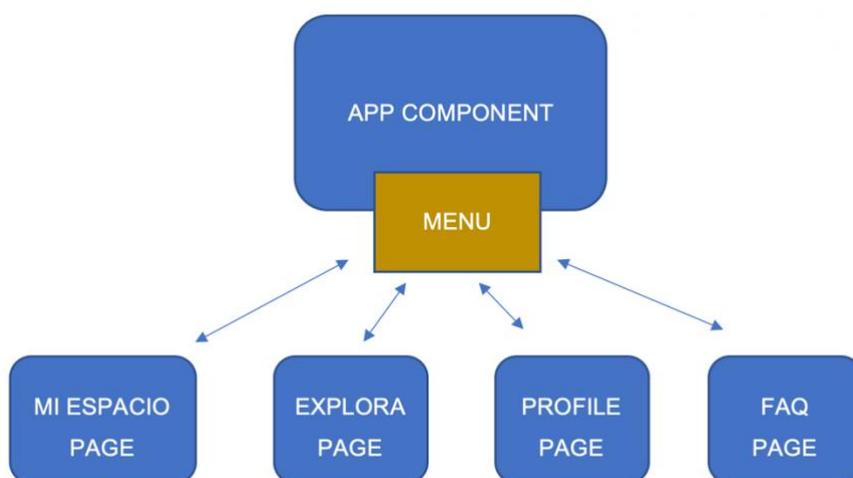


Ilustración 5: Diseño modular de la aplicación

Como se observa, la aplicación consta de un componente principal en el que está integrado un menú. Este menú sirve como relación entre el componente principal (que es donde se carga la vista y controlador activo) y las vistas con sus controladores disponibles.

Es necesario definir los servicios de los que va a hacer uso el controlador de cada vista. Se han definido tres servicios, el servicio de Explora, el servicio de Firebase y el servicio de Mi Espacio. Cada uno de estos servicios cumple las funcionalidades específicas de las páginas a las que da soporte. Es decir, se encargan de obtener y guardar los datos necesarios para ofrecer estas funcionalidades.

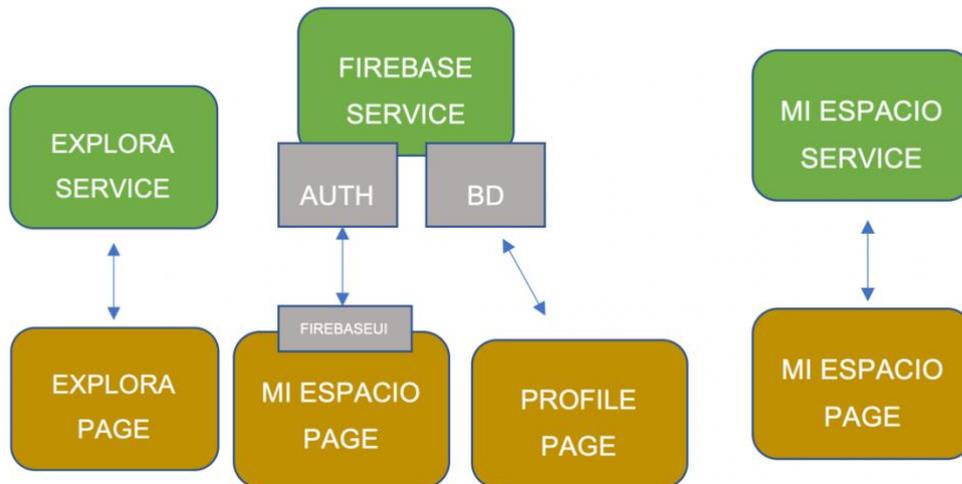


Ilustración 6: Servicios de la aplicación

El servicio de Explora, es utilizado por la pantalla de Explora y sus derivados (disponibles en el Anexo II: Diseño de alto nivel). Este servicio, se encarga de obtener los datos necesarios para las funcionalidades de Explora haciendo uso de la API que proporciona Adoswe. El servicio de Explora incorpora todas las peticiones HTTP, que se hacen mediante Angular. Como ya se ha explicado, la forma de relacionar la vista de Explora con el servicio de Explora es mediante el controlador, es decir, la clase TS asociada a la pantalla de Explora.

Como siguiente servicio tenemos el servicio de Firebase. Este servicio surge por la necesidad de realizar la autenticación y la identificación de los usuarios haciendo uso de bases de datos diferentes. Aunque para el usuario es transparente, en la aplicación existe una distinción entre usuarios Firebase (usado para autenticar y ofrecer acceso al usuario) y el usuario Adoswe (usado para obtener la información del usuario de la plataforma). Para relacionar estos dos usuarios, se ha hecho uso de la base de datos que proporciona Firebase. De este modo, podemos asociar el identificador de usuario de Firebase con el identificador de usuario de Adoswe y tratarlo como un único usuario global con dos identificadores. Es por eso que este servicio abastece de datos a dos módulos, el módulo de Mi Espacio para validar los datos de autenticación y el módulo de perfil para obtener los datos de Adoswe. Debido a que existe un módulo ya desarrollado para realizar el proceso de autenticación, el módulo de Mi Espacio hace uso de *“FirebaseUI”* para realizar este proceso.

Por último, la estructura del servicio de Mi Espacio es similar al servicio de Explora, pero con las funcionalidades específicas del módulo de Mi Espacio.

Tanto el diseño de alto nivel como el de bajo nivel, se encuentra detallado en los Anexos II y III respectivamente. El sistema completo tendrá la siguiente estructura.

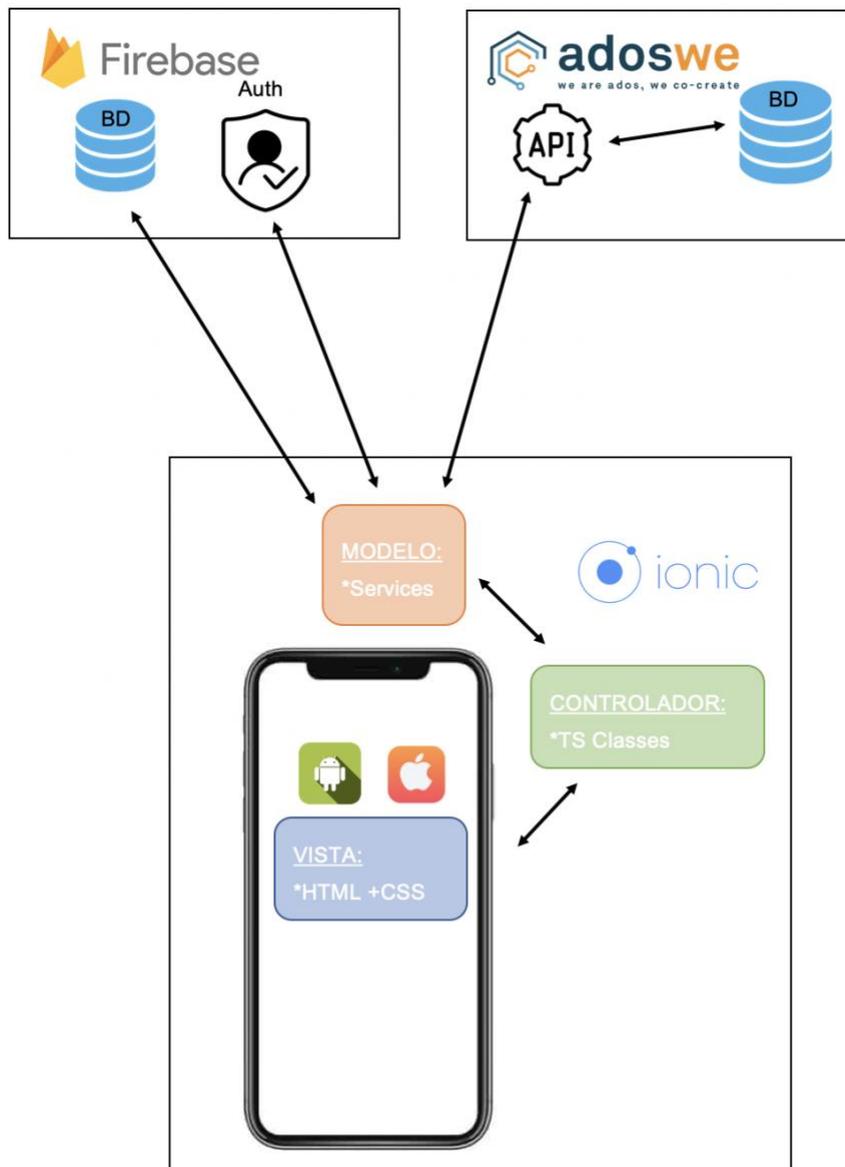


Ilustración 7: Diseño del sistema completo

En este esquema global, se puede ver como es el flujo de información entre las diferentes partes que componen el sistema.

7 Planificación del trabajo

Para desarrollar este proyecto, se han a definido una serie de tareas. Estas tareas están agrupadas en diferentes fases. Para dar por finalizada una fase se requiere haber completado cada una de las tareas que la componen. Además, tras la finalización de cada fase, se ha planificado la correspondiente reunión de validación de cada una. Con esta organización se pretende obtener un desarrollo eficiente y ordenado, además de poder tener organizados los recursos necesarios para el desarrollo de cada tarea.

7.1 Fases

A continuación, se muestra cada fase incluyendo una pequeña descripción de lo que consta cada una, los recursos necesarios y los entregables que se han generado.

7.1.1 Especificación

Descripción

En esta primera fase, se ha realizado una especificación completa tanto de las funcionalidades de la aplicación como del aspecto visual que tiene que tener la misma.

Recursos

Para llevar a cabo esta fase se ha hecho uso de:

- Ordenador con acceso a internet.
- Software de editor de documentos.

Entregables

Al finalizar esta fase, se ha generado un documento con las especificaciones determinadas y a partir de las cuales se ha llevado a cabo el proyecto.

7.1.2 Análisis y formación

Descripción

En esta fase se realizó un estudio de las herramientas a utilizar en el desarrollo del proyecto. Una vez determinadas las herramientas a utilizar, se estudió la documentación de las herramientas seleccionadas.

Recursos

Para llevar a cabo esta fase se ha usado un ordenador con acceso a internet.

Entregables

Los documentos generados en esta fase son:

- Análisis de alternativas.
- Selección de la solución (valoración y selección de las alternativas analizadas).

7.1.3 Diseño

Descripción

Esta fase denominada como fase de diseño, comprende las tareas necesarias para definir las características del diseño de los diferentes componentes que integra el sistema de forma que estos cumplan con las especificaciones establecidas. De la misma forma, se han definido las diferentes pruebas tanto unitarias como de integración que debe de cumplir el sistema para poder validar que verdaderamente cumple con las especificaciones definidas.

Recursos

Para llevar a cabo esta fase se ha hecho uso de:

- Ordenador con acceso a internet.
- Software de edición de documentos.

Entregables

Dado que en esta fase se define el cómo del desarrollo del proyecto, se han generado varios documentos:

- Diseño de alto nivel de la aplicación.
- Diseño de bajo nivel de la aplicación.
- Plan de pruebas.

7.1.4 Desarrollo

Descripción

En la fase de desarrollo, como su nombre indica se ha realizado el desarrollo de la aplicación siguiendo el diseño planificado.

Recursos

Para llevar a cabo esta fase se ha requerido:

- Ordenador con acceso a internet.
- IDE de programación (Visual Studio Code).

Entregables

Al finalizar esta fase se obtiene la **aplicación final** con cada una de las funcionalidades implementadas.

7.1.5 Pruebas

Descripción

A partir del plan de pruebas definido en la fase de diseño, se han realizado cada una de las pruebas necesarias para comprobar tanto el correcto funcionamiento del sistema como el correcto cumplimiento de las especificaciones.

Recursos

Para llevar a cabo esta fase se ha usado:

- Ordenador con acceso a internet
- IDE de programación (Visual Studio Code)
- Navegador web
- Dispositivo Android e IOS

Entregables

Al finalizar esta fase se generaron varios documentos:

- Resultado de las pruebas (Mediante este documento se evaluará la **necesidad de realizar los cambios** pertinentes en la aplicación)

7.1.6 Documentación

Descripción

Esta fase se ha desarrollado paralelamente a cada una de las fases descritas anteriormente. En esta fase se ha documentado cada uno de los pasos seguidos para llevar a cabo el proyecto.

Recursos

Para llevar a cabo esta fase se ha hecho uso de:

- Ordenador con acceso a internet
- Software de edición de documentos

Entregables

Con la finalización de esta fase se ha obtenido el **informe final** que además integra cada uno de los entregables de las demás fases.

7.2 Tareas

Una vez presentadas las fases del proyecto, se presentan las tareas y reuniones descritas anteriormente. Para cada tarea, se indica la fecha de comienzo y final. Se han planificado las siguientes tareas:

Nombre	Inicio	Final
Estudio y familiarización de la herramienta Ionic	03/13/19	03/21/19
Definir Especificaciones	03/13/19	03/13/19
Diseño e implementación de la estructura de la aplicación	03/14/19	03/21/19

Diseño y codificación menú lateral	03/14/19	03/14/19
Diseño y codificación de la pantalla "Mi Espacio"	03/15/19	03/19/19
Diseño y codificación de la estructura de la pantalla "Mi Espacio"	03/15/19	03/15/19
Diseño y codificación del modal de Notificaciones	03/18/19	03/18/19
Diseño y codificación del modal de Mensajes	03/19/19	03/19/19
Diseño y codificación de la pantalla "Explora"	03/15/19	03/15/19
Diseño y codificación de la pantalla "Perfil"	03/18/19	03/18/19
Diseño y codificación de la pantalla "FAQ"	03/19/19	03/19/19
Codificación de la navegación entre las pantallas	03/20/19	03/20/19
Pruebas de navegación entre las pantallas	03/21/19	03/21/19
Reunión de validación del esqueleto de la aplicación	03/22/19	03/22/19
Implementación login de la aplicación	03/22/19	04/03/19
Estudio de la herramienta Firebase	03/22/19	03/25/19
Diseño y codificación pantalla de Login con Firebase	03/26/19	03/28/19
Pruebas de autenticación y creación de nuevos usuarios	03/29/19	03/29/19
Codificación de la pantalla de perfil con datos de autenticación firebase	04/01/19	04/02/19
Pruebas con datos de perfil de diferentes usuarios	04/03/19	04/03/19
Reunión de validación de la autenticación y datos básicos del perfil	04/05/19	04/05/19
Implementación funcionalidades de la pantalla "Explora"	04/05/2019	04/19/19
Implementación separación de diferentes retos	04/05/19	04/08/19
Análisis de datos de la API	04/05/19	04/05/19
Codificación de los separadores de pantallas	04/05/19	04/08/19

Representación de datos de la API en tabs	04/09/19	04/10/19
Traducción de datos de API a datos de aplicación	04/09/19	04/09/19
Implementación de llamadas en función de la pestaña seleccionada	04/09/19	04/10/19
Implementación de búsqueda de texto y filtros	04/11/19	04/15/19
Pruebas de obtención de diferentes datos	04/16/19	04/16/19
Implementación de visualización de los detalles de cada reto	04/11/19	04/18/19
Implementación de visualización de detalles de retos	04/11/19	04/11/19
Implementación de visualización de detalles de retos + ofertas	04/15/19	04/16/19
Implementación de visualización de detalles de proyectos	04/17/19	04/18/19
Pruebas de validación del sistema Explora completo	04/19/19	04/19/19
Reunión de validación de "explora"	04/19/19	04/19/19
Implementación funcionalidades pantalla "Mi Espacio"	04/19/19	05/09/19
Implementación pantalla principal	04/19/19	04/19/19
Implementación del apartado de documentación	04/22/19	04/23/19
Implementación del apartado de fases	04/24/19	04/24/19
Implementación del apartado de participantes	04/25/19	04/25/19
Implementación de envío de mensajes a todos los participantes	04/26/19	04/29/19
Implementación de foros	04/30/19	05/08/19
Visualización de foros existentes	04/30/19	05/01/19
Implementación de la participación en el foro	05/02/19	05/06/19
Creación de nuevo foro	05/07/19	05/08/19
Pruebas de validación de "Mi Espacio"	05/09/19	05/09/19

Reunión de validación de "Mi Espacio"	05/10/19	05/10/19
Implementación de mensajería	05/10/19	05/22/19
Recepción de mensajes actuales	05/10/19	05/13/19
Envío de nuevo mensaje individual	05/14/19	05/15/19
Envío de nuevo mensaje grupal	05/16/19	05/17/19
Marcar mensaje como leído	05/20/19	05/21/19
Pruebas de validación de mensajes	05/22/19	05/22/19
Reunión de validación de mensajería	05/23/19	05/23/19
Implementación notificaciones	05/23/19	05/31/19
Obtención de notificaciones	05/23/19	05/27/19
Clasificación de notificaciones	05/28/19	05/30/19
Pruebas de validación de notificaciones	05/31/19	05/31/19
Reunión de validación de notificaciones	06/03/19	06/03/19
Implementación de FAQ	06/03/19	06/05/19
Prueba de sistema completo	06/06/19	06/06/19
Reunión de validación del sistema completo	06/07/19	06/07/19

Tabla 5: Tareas para el desarrollo del proyecto

7.3 Gantt

A continuación, se muestra el Gantt del proyecto, con las tareas expuestas anteriormente. Para desarrollar el Gantt se ha planificado que cada jornada es de 5 horas y que se trabajará en el proyecto 5 días a la semana. De esta forma, desde el inicio del proyecto al final del mismo, **se habrán realizado 300 horas**, completando así los requerimientos en cuanto a tiempo del trabajo en cuestión.

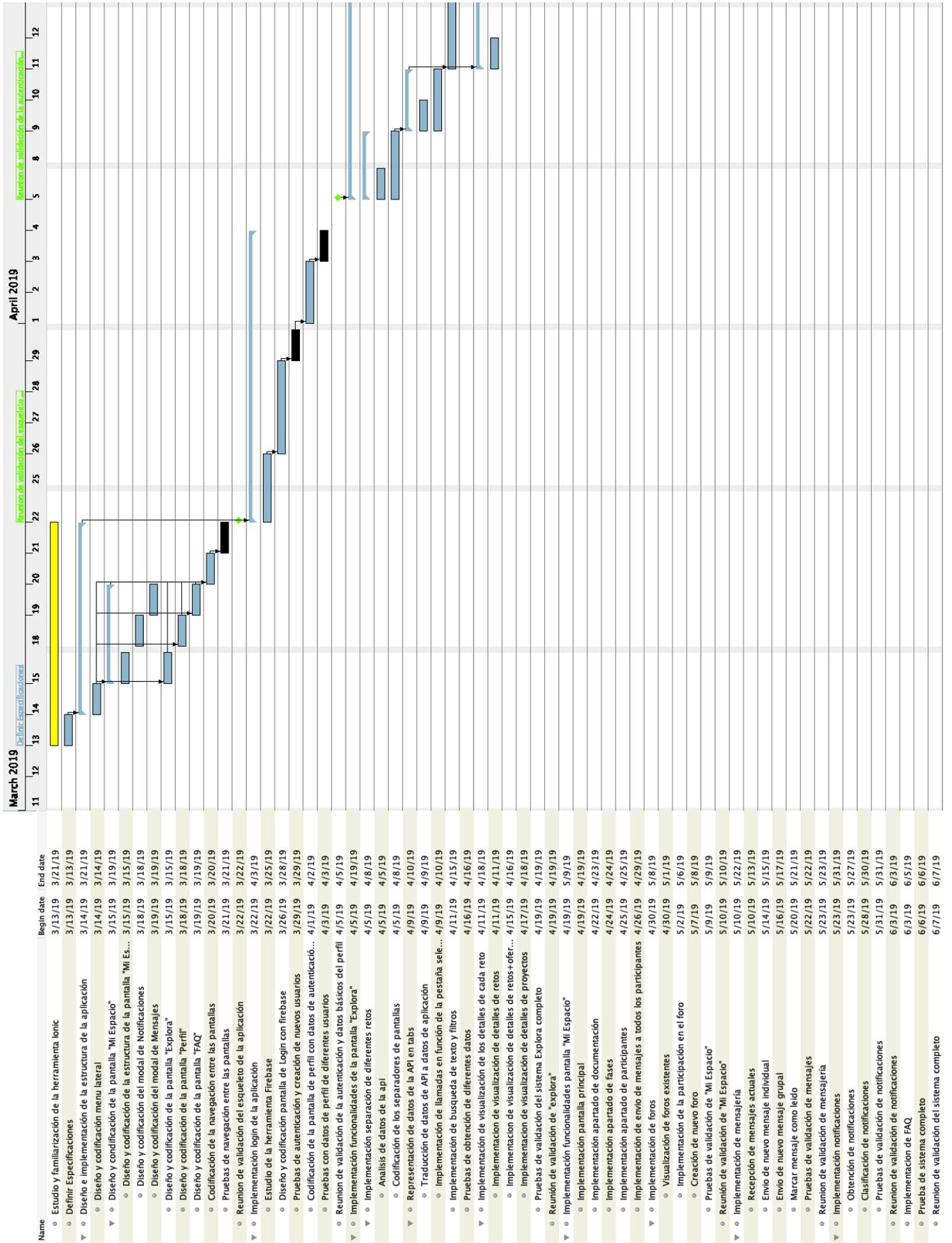


Ilustración 8: Parte I del Gantt

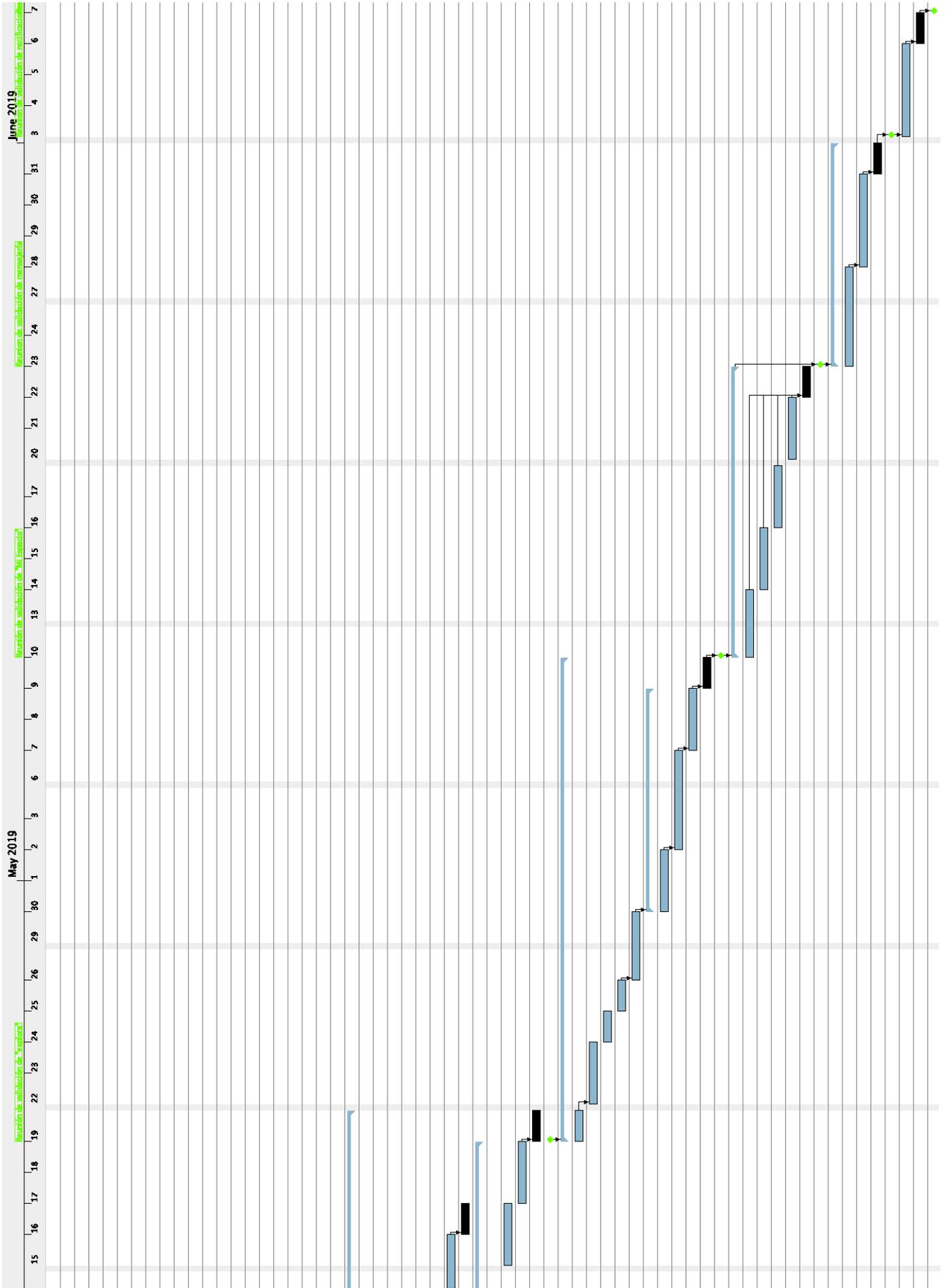


Ilustración 9: Parte II del Gantt

7.4 Seguimiento de las tareas

Una vez que se han definido tanto las fases como las tareas es necesario realizar un seguimiento de las mismas. Mediante este seguimiento, se ha podido evaluar varios aspectos interesantes relacionados con la planificación del trabajo. Por un lado, se ha evaluado el estado de las tareas, es decir, si la tarea está completada, en desarrollo o sin comenzar. Por otro lado, se ha evaluado el retraso o anticipo que lleva cada tarea. De esta forma es muy sencillo obtener el estado del proyecto completo.

Para llevar a cabo este seguimiento se ha seguido una evaluación del estado de las tareas de tipo semáforo. Esta evaluación se basa en asignar diferentes colores a cada posible estado de una tarea. Para determinar la desviación del tiempo del proyecto, se ha optado por recoger los valores de inicio, fin y duración real de cada tarea. Por último, se recoge un comentario por tarea a modo de breve explicación de posibles contratiempos que hayan retrasado la tarea o que hayan hecho que sea difícil de completar.

La siguiente tabla muestra la plantilla seguida para evaluar el avance del proyecto y comprobar si el estado del mismo se corresponde con la planificación definida.

Nombre de la tarea	Fecha de inicio real	Fecha de fin real	Duración real (días)	Estado	Comentario
Tarea 1	03/13/19	03/21/19	8		
Tarea 2	03/17/19	-	-		
Tarea 3	-	-	-		Retraso por complicaciones en tarea 2

Tabla 6: Plantilla ejemplo de seguimiento de tareas

COLOR	DESCRIPCIÓN
	Tarea completada
	Tarea completada con retraso
	Tarea en desarrollo
	Tarea sin comenzar con retraso
	Tarea sin comenzar
	Tarea cancelada

Tabla 7: Estados utilizados en el seguimiento de las tareas

7.5 Plan de pruebas

Para comprobar que el proyecto desarrollado cumple con las especificaciones definidas, es necesario definir un plan de pruebas. Este plan de pruebas además, permite comprobar que no existe ninguna problemática o malfuncionamiento de la aplicación a la hora de realizar las tareas requeridas.

7.5.1 Prueba de login

Con esta prueba se pretende probar la autenticación de la aplicación. Para obtener unos resultados óptimos se deben realizar diferentes pruebas.

7.5.1.1 Autenticación de un usuario registrado

Credenciales correctas

Se probará el login de un usuario **registrado** en la plataforma. Para este caso, la aplicación deberá comprobar las credenciales y **permitir** el acceso del usuario a la aplicación.

Credenciales incorrectas

Se probará el login de un usuario **registrado** en la plataforma con credenciales incorrectas. Para este caso la aplicación deberá comprobar las credenciales y **no permitir** el acceso del usuario a la aplicación.

7.5.1.2 Autenticación de un usuario no registrado

Se probará el login de un usuario **no registrado** en la plataforma. Para este caso la aplicación deberá comprobar las credenciales y **no permitir** el acceso del usuario a la aplicación.

7.5.1.3 Registro de un usuario y autenticación

Credenciales correctas

Para esta prueba se deberá de registrar un nuevo usuario en la plataforma. Después de esto se probará a acceder a la aplicación con este nuevo usuario. La aplicación deberá comprobar las credenciales y **permitir** el acceso del usuario a la aplicación

Credenciales incorrectas

Para esta prueba se deberá de registrar un nuevo usuario en la plataforma. Después de esto se probará a acceder a la aplicación con este nuevo usuario introduciendo credenciales incorrectas. La aplicación deberá comprobar las credenciales y **no permitir** el acceso del usuario a la aplicación

7.5.2 Prueba de comunicación

Con esta prueba se pretende comprobar que la comunicación entre el servidor donde se alojan los datos y la aplicación es correcta. Para ello se realizarán cada una de las peticiones de los datos al servidor y se comprobará que los datos recibidos en la aplicación y los datos alojados en el servidor son idénticos.

7.5.2.1 Datos de retos

Se realizará la petición para descargar un determinado reto. Se comprobará que los datos obtenidos son los mismos que los de la plataforma.

7.5.2.2 Datos de proyectos

Se realizará la petición para descargar un determinado proyecto. Se comprobará que los datos obtenidos son los mismos que los de la plataforma.

7.5.2.3 Datos de usuario

Se deberán descargar los datos de un usuario. Se comprobará que los datos obtenidos son los mismos que los de la plataforma además de comprobar que los datos de usuario corresponden con el usuario que se encuentra logueado³ en la aplicación.

7.5.2.4 Mensajes

Se descargarán los mensajes de un usuario. Se comprobará que los datos obtenidos son los mismos que los de la plataforma además de comprobar que los mensajes corresponden con el usuario que se encuentra logueado en la aplicación.

7.5.3 Prueba de modificación de datos

Esta prueba está relacionada con la prueba de comunicación. Con esta prueba se pretende comprobar que las modificaciones realizadas en los datos tanto del lado del servidor como del cliente (aplicación) se ven reflejados en los dos extremos. Primeramente, se deberá de realizar modificaciones en los datos alojados en el servidor y realizar las peticiones pertinentes desde la aplicación. Recibidos los datos, se comprobará que los datos recibidos son los mismos que los que hay en el servidor. Se realizará la misma labor, pero en este caso en el lado de la aplicación.

7.5.3.1 Añadir un reto

Para esta prueba se añadirá un nuevo reto a la plataforma. Una vez añadido, se realizará la petición de los retos. Se espera obtener los datos del reto añadido. Se debe comprobar que los datos recibidos del nuevo reto se corresponden con los datos de la plataforma.

³ El usuario ha realizado el proceso de login y esta autenticado en la aplicación.

7.5.3.2 Añadir un proyecto

Para esta prueba se añadirá un nuevo proyecto a la plataforma. Una vez añadido, se realizará la petición de los proyectos. Se espera obtener los datos del proyecto añadido. Se debe comprobar que los datos recibidos del nuevo proyecto se corresponden con los datos de la plataforma.

7.5.3.3 Eliminar un reto

Para esta prueba se eliminará un reto existente de la plataforma. Se deberá realizar la petición de retos y comprobar que no se reciben datos del reto eliminado.

7.5.3.4 Eliminar un proyecto

Para esta prueba se eliminará un proyecto existente de la plataforma. Se deberá realizar la petición de retos y comprobar que no se reciben datos del reto eliminado.

7.5.3.5 Editar un reto

Para esta prueba se modificará un reto existente de la plataforma. Se deberá realizar la petición de retos y comprobar que los datos del reto se reciben conforme a las modificaciones de la plataforma.

7.5.3.6 Editar un proyecto

Para esta prueba se modificará un proyecto existente de la plataforma. Se deberá realizar la petición de proyectos y comprobar que los datos del proyecto se reciben conforme a las modificaciones de la plataforma.

7.5.3.7 Editar información del perfil

En esta prueba se deberá de modificar información del perfil desde la aplicación. Se deberá comprobar desde la plataforma que se ven reflejados los cambios.

7.5.4 Prueba de navegación

Con esta prueba se pretende comprobar que la navegación a través de la app es fluida y sencilla a la vez que intuitiva. Además, se comprobará que se puede acceder a todas las funcionalidades de la aplicación. Para realizar esta prueba,

se deberá navegar por cada una de las pantallas y deberá medirse el tiempo de carga de cada pantalla. Por otro lado, esta prueba deberá realizarse por un usuario externo al desarrollo del proyecto y valorar los tres aspectos mencionados anteriormente: fluidez, sencillez y como de intuitiva es la navegación.

7.5.5 Prueba de visualización

Deberá comprobarse que cada uno de los componentes de las diferentes pantallas se visualiza de acorde al diseño de la interfaz desarrollada. Deberá comprobarse tanto para un dispositivo Android como para uno iOS.

Especialmente se deberá de realizar las siguientes pruebas para cada componente.

Para componentes de texto

7.5.5.1 Visualización de cadenas largas (>150 caracteres)

Se probará a visualizar en los diferentes cuadros de texto cadenas de caracteres con **más de 150 caracteres**. Habrá que comprobar que la visualización es correcta, es decir, los textos no se solapan, se entienden bien y se adaptan a las pantallas de los dispositivos correctamente.

7.5.5.2 Visualización de cadenas cortas (<150 caracteres)

Se probará a visualizar en los diferentes cuadros de texto cadenas de caracteres con **menos de 150 caracteres**. Habrá que comprobar que la visualización es correcta, es decir, los textos no se solapan, se entienden bien y se adaptan a las pantallas de los dispositivos correctamente.

7.5.5.3 Visualización de cadenas nulas

Se probará a visualizar en los diferentes cuadros de texto cadenas de caracteres **nulas**. Para este caso se deberá comprobar que el elemento de texto con valor nulo no se genera en la pantalla y que no ocurre ninguna anomalía en la interfaz gráfica.

Para componentes de imagen

7.5.5.4 Visualización de imágenes con distinta resolución

Por último, se probará a visualizar fotos con **diferentes resoluciones** en los componentes de imágenes. Para esta prueba se deberá comprobar que para cada una de las imágenes no se pierden las proporciones y se visualizan rellenando completamente el espacio reservado para la imagen.

8 Presupuesto

En el siguiente apartado se muestran las diferentes partidas del presupuesto elaborado para hacer frente a este proyecto.

La primera partida a calcular es la partida de horas internas. Para este proyecto se han necesitado un programador junior, y un programador senior. El sueldo de un programador junior se estima que es de una media de 19.247€ anuales [23]. Para el caso de un programador senior tenemos que la media salarial anual es de 28.851 € [24].

Con estos datos y sabiendo que la jornada anual es de 1944 horas [25] se calculan los costes horarios para el programador junior y el programador senior mediante la división de: **salario anual / horas anuales**. De esta manera, podemos calcular el coste de las horas internas dedicadas al proyecto.

Personal	Cantidad	Sueldo (€/h)	Horas	Total (€)
Programador senior	1	14,84	30	445,2
Programador junior	1	9,9	300	2970
			Total	3415,2

Tabla 8: Partida de horas internas

Por otro lado, es necesario tener en cuenta en el presupuesto los recursos necesarios para el proyecto. En este caso se han necesitado varios equipos para realizar el desarrollo del proyecto, así como diversas licencias de software.

En relación a las licencias de software utilizado, tenemos la siguiente tabla con el desglose del presupuesto.

Licencias	Coste (€)
Paquete Office	149
Desarrollador Play Store	15
Desarrollador Apple	90
Total	254

Tabla 9: Coste licencias de software

Por último, se recogen los recursos hardware utilizados para desarrollar el proyecto. Cada uno de los equipos cuenta con una vida útil determinada. Como la duración del proyecto es menor que la vida útil de los equipos, ha sido necesario calcular la parte proporcional del coste del equipo atribuible al proyecto. Para realizar este cálculo necesitamos conocer varias cosas:

- **Vida útil:** Es la vida media estimada que un objeto puede tener, cumpliendo correctamente con la función para la que se ha creado.
- **Coste unitario:** El coste total del objeto que se está contemplando
- **Uso real en el proyecto:** La cantidad de horas del proyecto en las que se ha hecho uso del activo.

Conocidos estos datos, se ha calculado el coste del equipo atribuible al proyecto con la siguiente función.

$$\text{Amortización} = \frac{\text{Coste Unitario(€)} \times \text{Uso Real}(h)}{\text{Vida útil}(h)}$$

Ecuación 1: Cálculo de la amortización de activos del proyecto

La siguiente tabla representa la partida de amortizaciones para los equipos utilizados en el desarrollo del proyecto.

Equipo	Coste unidad	Unidades	Vida útil (años)	Uso (h)	Total (€)
Apple Macbook pro	1500	1	5	300	46,29
Iphone X	900	1	2	100	23,14
Samsung Galaxy Grand	250	1	2	100	6,43
				Total	75,86

Tabla 10: Partida de amortizaciones del proyecto

Teniendo todo esto en cuenta el coste total del proyecto ha sido el siguiente.

Presupuesto Proyecto	
Horas Internas	3415,2 €
Amortizaciones	75,86 €
Licencias	254 €
SUBTOTAL	3745,06 €
Imprevistos (10%)	374,50 €
TOTAL	4119,56 €

Tabla 11: Total del presupuesto del proyecto

Se estima que el coste del proyecto ha sido de **4119,56€**. Como se habló en el apartado de **Análisis de riesgos** se ha estimado un 10% del coste total del proyecto como imprevistos para poder hacer frente a cualquier planificación y distribución errónea del presupuesto.

9 Resultados

Como resultado principal de este proyecto, se ha conseguido desarrollar una aplicación, que cumple con las especificaciones, a partir del diseño planteado para la misma. Si bien es cierto, debido a diferentes problemas que han surgido a lo largo del ciclo de vida del proyecto ha sido necesario rediseñar y cambiar partes del proyecto para adaptarlo. En el siguiente apartado, se exponen los resultados finales a nivel visual de la aplicación, así como los cambios realizados en el diseño y la justificación de los mismos.

9.1 Aplicación final

Los resultados de la aplicación, se basan en el diseño tanto de alto como de bajo nivel desarrollado en la fase de diseño. Los resultados de esta fase están disponibles en el *Anexo II* y el *Anexo III*.

9.1.1 Login

Como se especificó al inicio del proyecto, el login se ha desarrollado haciendo uso de la autenticación de firebase. El aspecto de la pantalla de login de la aplicación es el siguiente.

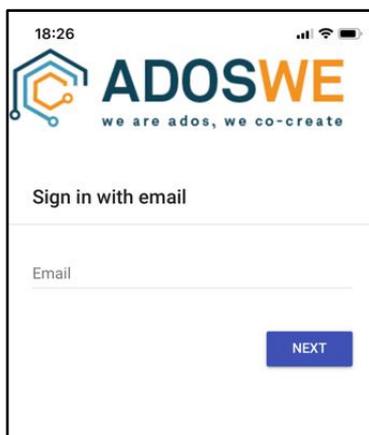


Ilustración 10: Pantalla de login

9.1.2 Mi Espacio

A continuación, se muestran las capturas reales de la aplicación para la pantalla y funcionalidades de Mi Espacio.



Ilustración 11: Pantalla principal de Mi Espacio

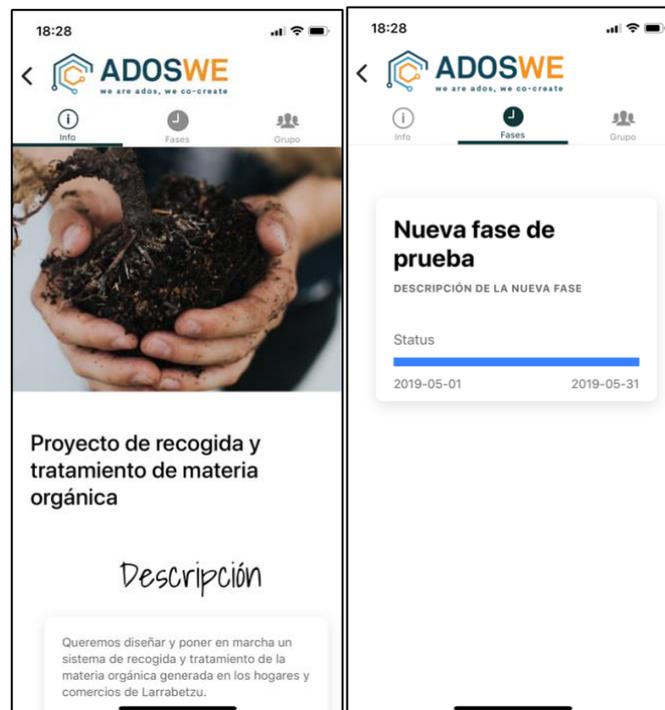


Ilustración 12: Pantalla de información y fases de Mi Espacio

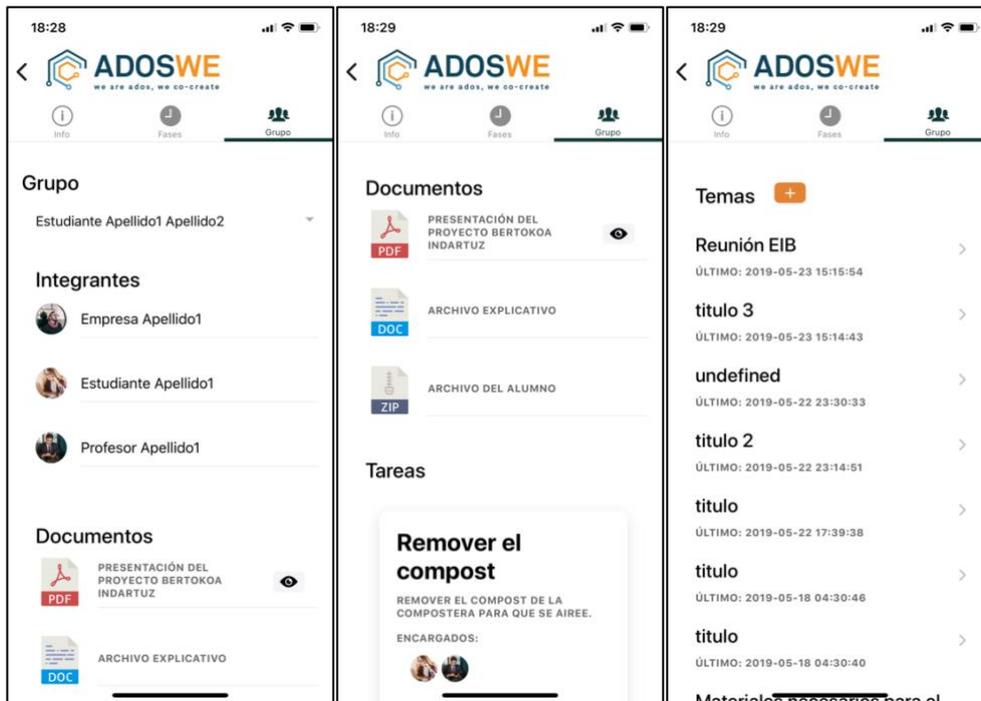


Ilustración 13: Pantallas de grupos de Mi Espacio

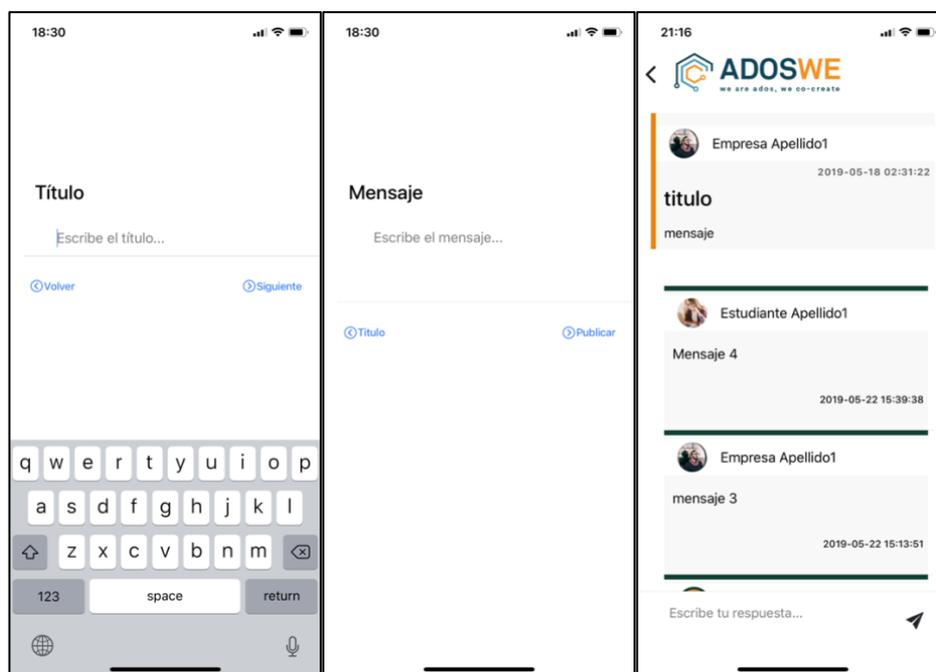


Ilustración 14: Pantallas de creación e interacción de foros

9.1.3 Explora

A continuación, se muestran las capturas reales de la aplicación para la pantalla y funcionalidades de Explora.

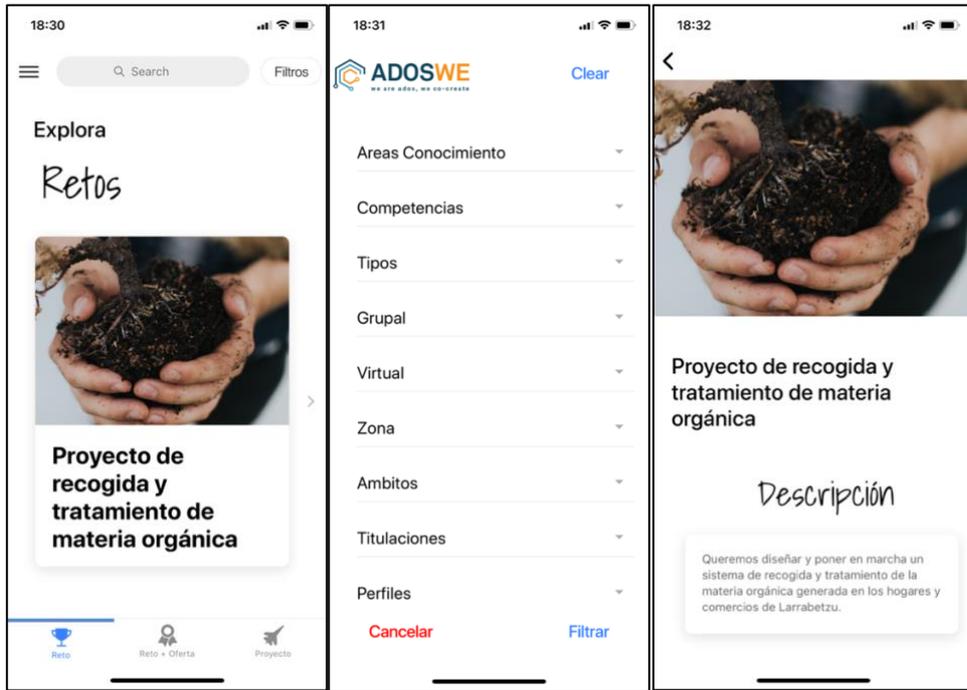


Ilustración 15: Pantallas de Explora

9.1.4 Perfil

A continuación, se muestran las capturas reales de la aplicación para la pantalla del perfil.



Ilustración 16: Pantalla de perfil

9.1.5 Menú

A continuación, se muestran las capturas reales de la aplicación para el menú de la aplicación.

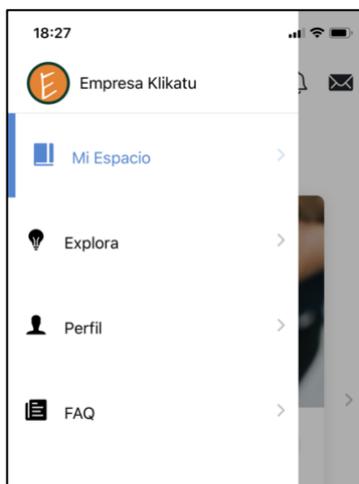


Ilustración 17: Menú de la aplicación

9.2 Problemas y desviaciones.

A lo largo de la fase de implementación han surgido diferentes inconvenientes que han ocasionado cambios en el diseño de la aplicación.

9.2.1 Cross Origin Resource Sharing (CORS)

La tecnología CORS, define la forma en la que un navegador y un servidor deben interactuar con el fin de decidir si es seguro aceptar una petición de origen cruzado. El problema surge cuando el servidor destino no tiene configurado ciertos parámetros que permiten a determinados clientes acceder a sus recursos, ya que en estos casos las políticas de CORS bloquean las peticiones y no permiten obtener el contenido que se requiere. Este ha sido uno de los principales problemas con lo que se ha tenido que lidiar a la hora de desarrollar la aplicación. Principalmente se propuso implementar la aplicación utilizando angular para realizar las peticiones HTTP correspondientes a la API de Adoswe. Debido a este problema, se replanteó el diseño de forma que las peticiones se hicieran a través de los recursos nativos del teléfono móvil. Tras implementar la aplicación a partir de este segundo diseño, se observó que existía una latencia extremadamente alta a la hora de cargar los recursos de la aplicación, cosa que

no se consiguió resolver. Por lo tanto, se volvió a pensar el diseño de forma que las peticiones se hicieran a través de angular haciendo uso de un proxy. Este último diseño, a pesar de que es completamente funcional, presenta varias desventajas:

- Dependencia de un proxy público. Si el proxy dejase de estar disponible la aplicación dejaría de funcionar
- Privacidad. La información que se intercambia entre el servidor y la aplicación pasa a través de un equipo que no es de nuestro dominio.

Por lo tanto, para solucionar esto, es necesario replantear y reconfigurar la política de CORS para que de esta forma acepte las peticiones y no sea necesario utilizar un proxy.

9.2.2 HOTLINK

Otro de los problemas que surgió fue el tema de los hotlink para las url de las imágenes. Los recursos de Adoswe se encuentran almacenados a través de un hosting de terceros. Debido a esto, las imágenes estaban protegidas de forma que no se podía acceder a ellas desde dominios externos a Adoswe. Esto ocasionaba que no se pudieran visualizar las imágenes en la aplicación. Para solucionar este inconveniente, la empresa de hosting desactivó la protección de hotlink que tenía sobre las imágenes.

9.2.3 Reestructuración de las pantallas

En el diseño principal no existía relación entre la sección de mensajes y los retos de los usuarios. Debido a un cambio en las especificaciones, se fijó que los mensajes que los usuarios enviaban debían estar relacionados directamente con los retos/proyectos de la aplicación. Se reestructuró el acceso y envío de los mensajes, trasladando la funcionalidad, hasta el momento, independiente de mensajes, a una subfuncionalidad del apartado de Mi Espacio. De esta forma, en el diseño actual los mensajes solo son accesibles y solo se pueden enviar desde la pantalla de Mi Espacio.

9.2.4 Retrasos de las entregas

Debido a factores externos, no se han tenido a tiempo los recursos necesarios para desarrollar la parte de las notificaciones, mensajería y FAQ de la aplicación. Para solventar este problema, se han desarrollado estas partes de forma externa a la plataforma Adoswe. Estos apartados se han desarrollado haciendo uso de Firebase y la base de datos que ofrece.

Para desarrollar el chat y las notificaciones, la base de datos se ha diseñado de la siguiente forma. Se ha planteado una base de datos estilo árbol. En esta base de datos se almacenan los datos en forma de **clave:valor**.

Para la funcionalidad de mensajería se han desarrollado tres ramas. Por un lado, la rama de usuarios que guarda la siguiente información:

- Nombre de usuario.
- Id de usuario.
- Foto de usuario.

Por otro lado, la rama de chats, en la que se guarda cada chat creado con la siguiente información:

- Id del chat.
- Id del participante a.
- Id del participante b.

Por ultimo, la rama de mensajes. Estos mensajes estarán formados de la siguiente forma:

- Id del mensaje.
- Id del chat al que pertenecen.
- Id del remitente.
- Texto del mensaje.

Para desarrollar la parte de notificaciones se ha creado otra rama en la base de datos de Firebase. Esta rama se ha denominado como rama de notificaciones y dentro de esta rama existe una entrada para cada uno de los usuarios

registrados en la plataforma. Cada una de las entradas se identifica a partir del ID de Firebase de cada usuario. Dentro de la entrada de cada usuario existe una entrada para cada notificación que se quiera mandar a dicho usuario con la siguiente información:

- Título: Título de la notificación.
- Descripción: Descripción donde se aclara el contenido de la notificación.
- Tipo: Grado de importancia de la notificación. Se han definido tres grados:
 - Urgente: Notificación con mayor grado de importancia.
 - Importante: Notificación con grado de importancia media.
 - Normal: Notificación con menor grado de importancia.

Para el apartado de FAQ, se ha utilizado contenido estático. Esta solución es temporal con la meta de tener un prototipo “funcional” y se ha diseñado de forma que la estructura sea reutilizable con vistas a integrarlo con la plataforma Adoswe cuando se dispongan de los recursos necesarios.

En lo relativo a cambios en la aplicación, se han añadido dos servicios adicionales para cumplir con esta funcionalidad.

El servicio de Mensajes, que se encarga de obtener:

- Los contactos disponibles a los cuales mandar un mensaje.
- Los chats abiertos con diferentes usuarios y su contenido.

A su vez, se encarga de mandar los mensajes que el usuario genera.

El servicio de Notificaciones, se encarga de recuperar las notificaciones de un usuario.

Tanto el servicio de mensajes como de notificaciones se encargan de obtener y escribir los datos en la base de datos de Firebase según la operación indicada por el controlador correspondiente.

10 Conclusiones

Es importante que una vez se haya finalizado el proyecto, se valore si este a conseguido cumplir con los objetivos y especificaciones que se han determinado. También es necesario evaluar la planificación y el desarrollo seguido con el fin de obtener puntos débiles en la planificación y poder mejorarlos en futuros proyectos y así tener mayor tasa de éxito.

Como se presentó al comienzo del trabajo, el objetivo principal del proyecto, era realizar una aplicación funcional que cumpliera con las especificaciones dadas por la empresa Adoswe. Dentro de estas especificaciones, debido a los problemas que han surgido a lo largo de la vida del proyecto, no se han llegado a cumplir todas, sin embargo, se han conseguido encontrar soluciones a estos problemas y desarrollar una aplicación que, aun no cumpliendo con todas las especificaciones, ofrece cada una de las funcionalidades que se propusieron. Por lo que podemos decir que, aunque el objetivo principal no se ha completado en su totalidad debido a los contratiempos surgidos, se ha conseguido cumplir en gran medida el objetivo principal de realizar una aplicación móvil multiplataforma para una red social universitaria. La aplicación desarrollada, está preparada para incorporar el resto de funcionalidades de la plataforma cuando éstas estén disponibles. El resto de objetivos se han cumplido satisfactoriamente.

Uno de los puntos débiles de la planificación del trabajo, ha sido la asignación de tiempo a cada tarea. Debido a la falta de experiencia y al tiempo limitado se asignaron tiempos muy justos a cada tarea. Si bien es cierto, que se han conseguido cumplir los plazos establecidos, no existía mucho tiempo de maniobra ante posibles retrasos o imprevistos, lo que suponía un riesgo muy grande para el proyecto. Por lo tanto, una conclusión que obtenemos de la planificación de este proyecto, es que es necesario asignar unos márgenes de maniobra más amplios en el reparto de tiempos de cada tarea de forma que podamos trabajar con un poco de holgura ante posibles imprevistos.

Por otro lado, este TFG se ha desarrollado con el equipo de Adoswe, esto ha sido ventajoso en lo relativo al aprendizaje y adquisición de experiencia. Debido

a esto, en paralelo a los conocimientos técnicos que ha proporcionado el propio desarrollo de la app, se ha podido adquirir experiencia en la gestión de plazos de entrega, comunicación con clientes y comunicación con proveedores. Además, se han podido vivir en primera persona, riesgos que surgen cuando se trabaja en proyectos con otras personas, y lo más importante, se ha podido trabajar en solucionar estos riesgos de forma que el proyecto se vea afectado lo mínimo posible lo cual en ocasiones supone un gran reto. Dicho esto, se han conseguido alcanzar parte de los objetivos técnicos del proyecto como los objetivos personales marcados al inicio del proyecto.

Para finalizar, se exponen las futuras mejoras y funcionalidades que se pueden agregar al proyecto.

Futuras mejoras

- Integración de autenticación de la plataforma Adoswe con la autenticación de Firebase.
- Añadir autenticación a través de redes sociales.
- Utilizar peticiones HTTP nativas o modificar el servidor para evitar el uso de un proxy para solucionar del problema de CORS.
- Ofrecer una GUI personalizable para cada usuario.
- Integrar la parte de mensajes con la plataforma de Adoswe.
- Integrar la parte de notificaciones con la plataforma de Adoswe
- Añadir las FAQ específicas de la plataforma de Adoswe.

El proyecto que se ha desarrollado se encuentra disponible en el siguiente repositorio privado: <http://github.com/teratonix/adoswe.git>.

A su vez, el proyecto de Firebase creado para esta aplicación se encuentra disponible en el siguiente enlace:

<https://console.firebase.google.com/project/logintest-ec9b6/overview?hl=es-419>

Bibliografía

- [1] A. F. Patterson, «Contexto de Adoswe,» 2019.

- [2] A. Batanero y M. Sesma, «VERSUS: Apps híbridas VS Apps Nativas,» 10 Julio 2017. [En línea]. Available: <https://www.paradigmadigital.com/dev/versus-apps-hibridas-vs-apps-nativas/>. [Último acceso: 9 Abril 2019].

- [3] Ionic, «Hybrid vs Native,» 2018.

- [4] J. J. Gutiérrez, «¿Qué es un framework?,» [En línea]. Available: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf. [Último acceso: 27 Abril 2019].

- [5] P. Chalin, D. R. Gonzalez y S. Zakhour , «Flutter.dev,» Abril 3 2019. [En línea]. Available: <https://flutter.dev/docs/resources/technical-overview>. [Último acceso: 29 Abril 2019].

- [6] «Hacker Noon,» 18 Septiembre 2018. [En línea]. Available: <https://hackernoon.com/flutter-vs-react-native-vs-xamarin-for-cross-platform-development-5f92cfb178ff>. [Último acceso: 12 Abril 2019].

- [7] Microsoft, «Visual Studio,» Microsoft, [En línea]. Available: <https://visualstudio.microsoft.com/xamarin/>. [Último acceso: 29 Abril 2019].

- [8] Codemagic, «Flutter vs xamarin a developer's perspective,» [En línea]. Available: <https://blog.codemagic.io/flutter-vs-xamarin-a-developer-s-perspective/>. [Último acceso: 29 Abril 2019].

- [9] N. Khandelwal, D. Blotsky, M. Myers y R. Knoll, «Cordova overview,» 18 Octubre 2018. [En línea]. Available: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>. [Último acceso: 2019 Abril 27].
- [10] S. Malik, «Xamarin versus Cordova,» *Code Magazine*, nº 2017 - March/April, Abril 2017.
- [11] Skelia, «XAMARIN VS APACHE CORDOVA: WHICH ONE IS BETTER?,» 2017 Enero 27. [En línea]. Available: <https://skelia.com/articles/xamarin-vs-apache-cordova-one-better/>. [Último acceso: 2019 Abril 29].
- [12] JS js.foundation, «jQuery Mobile,» [En línea]. Available: <https://jquerymobile.com>. [Último acceso: 28 Abril 2019].
- [13] A. Schmitz, «Updates | jQuery Mobile,» 11 Mayo 2017. [En línea]. Available: <https://blog.jquerymobile.com/category/updates/>. [Último acceso: 2019 Abril 29].
- [14] H. Atha, «Performance and Speed - 6 Hybrid Frameworks For Better App Development,» 12 Octubre 2017. [En línea]. Available: <https://www.moveoapps.com/blog/performance-and-speed-hybrid-frameworks-for-better-app-development/>. [Último acceso: 2019 Abril 27].
- [15] Ionic, «Ionic - Cross-Platform Mobile App Development,» [En línea]. Available: <http://ionicframework.com>. [Último acceso: 24 Abril 2019].
- [16] Ionic, «Ionic Documentation,» [En línea]. Available: <https://ionicframework.com/docs/native>. [Último acceso: 24 Abril 2019].

- [17] Web Components, «Discuss & share web components,» [En línea]. Available: <https://www.webcomponents.org/introduction>. [Último acceso: 30 Abril 2019].
- [18] M. Lynch, «Introducing Ionic 4: Ionic for Everyone,» 23 Enero 2019. [En línea]. Available: <https://blog.ionicframework.com/introducing-ionic-4-ionic-for-everyone/>. [Último acceso: 2019 Abril 30].
- [19] D. Jabif, «Ionic 4 vs Ionic 3—What you need to know about Ionic 4,» 27 Febrero 2019. [En línea]. Available: <https://ionicthemes.com/tutorials/about/ionic-4-vs-ionic-3>. [Último acceso: 2019 Abril 30].
- [20] A. F. Vazquez, «Ciclo de vida de un proyecto SW,» 15 Noviembre 2019. [En línea]. [Último acceso: 18 Mayo 2019].
- [21] Git, «Git,» [En línea]. Available: <https://git-scm.com>. [Último acceso: 18 Mayo 2019].
- [22] Atlassian, «Gitflow Workflow,» [En línea]. Available: <https://www.atlassian.com/cs/git/tutorials/comparing-workflows/gitflow-workflow>. [Último acceso: 18 Mayo 2019].
- [23] Indeed, «Salarios para empleos de Programador/a junior en España | Indeed.es,» 11 Mayo 2019. [En línea]. Available: <https://www.indeed.es/salaries/Programador/a-junior-Salaries>. [Último acceso: 14 Mayo 2019].
- [24] Indeed, «Salarios para empleos de Programador/a senior en España | Indeed.es,» 10 Mayo 2019. [En línea]. Available: <https://www.indeed.es/salaries/Programador/a-senior-Salaries>. [Último acceso: 14 Mayo 2019].

[25] Iberley, «Cómputo anual de la jornada de trabajo cuando no se especifique nada en convenio colectivo,» 13 Abril 2016. [En línea]. Available: <https://www.iberley.es/practicos/caso-practico-computo-anual-jornada-trabajo-no-especifique-convenio-colectivo-3231>. [Último acceso: 14 Mayo 2019].