



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería de Software

**AquaCyL: Desarrollo de una aplicación móvil
para la consulta del estado de las Zonas de Baño
de Castilla y León**

Alumno: Pablo Varela Vázquez

Tutores: Yania Crespo González-Carvajal

Raúl Sánchez Plaza



A mis abuelas

Agradecimientos

Quiero agradecer a mi familia y amigos su apoyo durante todo este tiempo para poder llegar hasta hoy con este proyecto.

Además, quiero dar las gracias a mi tutora académica, Yania, por su asesoramiento y apoyo a lo largo del desarrollo de este proyecto.

De igual manera, agradezco a mi tutor de empresa, Raúl, su orientación y experiencia que ha aportado al desarrollo de este proyecto.

Finalmente, quiero agradecer a las 29 personas que probaron la aplicación por todas sus sugerencias y reportes de problemas.

Muchas gracias a todos.

Resumen

El objetivo de este proyecto es desarrollar una aplicación para dispositivos móviles que permita consultar información sobre el estado de las zonas de baño autorizadas en la comunidad autónoma de Castilla y León. Se hará uso de datos abiertos de la Junta de Castilla y León. Además, se proporcionará otros datos relevantes sobre las localidades donde se encuentren dichas zonas de baño, a partir de APIs públicas proporcionadas por servicios externos. La aplicación incluye algunos aspectos de red social como puntuaciones de los usuarios, comentarios temporales, pensados para una situación puntual, y permanentes, para una valoración general de la zona.

Esta aplicación móvil se ha desarrollado utilizando el framework Flutter, que emplea el lenguaje de programación Dart, y se apoya en Firebase como backend y en APIs públicas, como se ha mencionado antes. El proyecto se ha gestionado utilizando una adaptación del marco de trabajo ágil *Scrum*.

Palabras Clave: Flutter, Firebase, Scrum, Zonas de Baño, Castilla y León, Datos Abiertos.

Abstract

The aim of this project is to develop a mobile application that enables users to access information about the status of authorized bathing areas in the autonomous community of ‘Castilla y León’. The application will use Open Data provided by the regional government of ‘Castilla y León’. Additionally, it will offer other pertinent information about the localities where these bathing areas are located, sourced from public APIs provided by external services. The application will incorporate social networking features, including user ratings, temporary comments for specific situations, and permanent comments for a general assessment of the area.

This mobile application has been developed using Flutter, which uses the Dart programming language. The app relies on Firebase for its backend and integrates with public APIs as previously mentioned. The project has been managed using an adaptation of the agile Scrum framework.

Key words: Flutter, Firebase, Scrum, bathing areas, Castilla y León, Open Data.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XV
Lista de tablas	XIX
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	2
1.3.1. Objetivos de desarrollo	2
1.3.2. Objetivos personales	2
1.4. Estructura de la memoria	3
2. Requisitos y Planificación	5
2.1. Scrum	5
2.1.1. El equipo Scrum	6
2.1.2. Eventos de Scrum	7

2.1.3. Artefactos de Scrum	8
2.1.4. Adaptación de Scrum al proyecto	9
2.2. Análisis de riesgos	10
2.2.1. Riesgos generales	10
2.2.2. Riesgos particulares	13
2.3. Planificación y calendarización	15
2.4. Product Backlog	17
2.5. Presupuesto	17
2.5.1. Presupuesto simulado	18
2.5.2. Presupuesto real	19
3. Análisis	21
3.1. Stakeholders	21
3.2. Requisitos funcionales como historias de usuario	21
3.3. Requisitos de información como historias de usuario	24
3.4. Requisitos no funcionales	24
3.5. Reglas de negocio	25
3.6. Modelo de dominio	26
3.7. Modelo de casos de uso	27
3.8. Modelado de las zonas de baño como máquinas de estado	29
4. Tecnologías utilizadas	31
4.1. Herramientas de desarrollo	31
4.1.1. Flutter	31
4.1.2. Firebase	36
4.1.3. Postman	40
4.1.4. Justinmind	40

- 4.1.5. Android Studio 41
- 4.1.6. Visual Paradigm 42
- 4.1.7. Bootstrap 43
- 4.1.8. Visual Studio Code 43
- 4.2. Herramientas de gestión del proyecto 43
 - 4.2.1. GitHub 43
 - 4.2.2. Jira 45
 - 4.2.3. Sourcetree 47
- 4.3. Herramientas de análisis 47
 - 4.3.1. Google Analytics 47
- 4.4. Herramientas de comunicación 48
 - 4.4.1. Microsoft Teams 48
- 4.5. Herramientas para la documentación 48
 - 4.5.1. Overleaf 48
- 5. Diseño 49**
 - 5.1. Resumen de las decisiones de diseño 49
 - 5.2. Arquitectura MVVM 50
 - 5.3. Diseño inicial de la interfaz de usuario 51
 - 5.4. Imagen de marca 55
 - 5.5. Diseño arquitectónico 56
 - 5.6. Diseño de la base de datos 64
 - 5.7. Diseño de comunicación 64
 - 5.8. Diagrama de despliegue 68
- 6. Implementación y pruebas 69**
 - 6.1. Implementación 69

6.1.1.	Principales problemas y retos	69
6.1.2.	Cambios en requisitos	70
6.1.3.	Guías de diseño seguidas	70
6.1.4.	Licencia	71
6.1.5.	Organización del repositorio	71
6.2.	Pruebas	74
6.2.1.	Pruebas de aceptación de usuario (UAT)	75
6.2.2.	Prueba en canal cerrado de Google Play	88
7.	Seguimiento del proyecto	89
7.1.	Seguimiento de los Sprints	89
7.1.1.	Sprint 0 (20/02/2024 - 29/02/2024)	89
7.1.2.	Sprint 1 (29/02/2024 - 14/03/2024)	91
7.1.3.	Sprint 2 (14/03/2024 - 04/04/2024)	92
7.1.4.	Sprint 3 (04/04/2024 - 18/04/2024)	94
7.1.5.	Sprint 4 (18/04/2024 - 02/05/2024)	95
7.1.6.	Sprint 5 (02/05/2024 - 16/05/2024)	96
7.1.7.	Sprint 6 (16/05/2024 - 30/05/2024)	97
7.1.8.	Sprint 7 (30/05/2024 - 13/06/2024)	99
7.2.	Resumen del desarrollo del proyecto	100
7.2.1.	Riesgos	100
7.2.2.	Calendarización y tiempo empleado	100
7.3.	Costes	100
7.3.1.	Coste simulado final	100
7.3.2.	Coste real final	101
7.3.3.	Evaluación de Costes Futuros	102

8. Conclusiones	103
8.1. Líneas de trabajo futuras	104
Bibliografía	105
A. Manuales	113
A.1. Manual de instalación	113
A.2. Manual de mantenimiento y de despliegue en Google Play	113
A.3. Manual de usuario	114
A.3.1. Acceso a la aplicación	114
A.3.2. Pantalla Home	115
A.3.3. Pantalla de detalles	117
A.3.4. Pantalla de favoritos	119
A.3.5. Ajustes	119
B. Resumen de enlaces adicionales	121

Lista de Figuras

2.1. Ciclo de vida de Scrum. Tomada de [91].	5
3.1. Modelo de dominio	26
3.2. Diagrama de casos de uso de la aplicación	28
3.3. Diagrama principal estados zona de baño	29
3.4. Diagrama temporada baño estados zona de baño	30
4.1. Flujo de actualización flexible. Tomada [34]	33
4.2. Flujo de actualización inmediato. Tomada [34]	33
4.3. Interfaz Android Studio. Tomada [32]	42
4.4. Interfaz de GitHub con el proyecto	44
4.5. Interfaz de Jira	46
4.6. Interfaz de la aplicación en una tarea	47
5.1. Patrón MVVM. Tomada [12]	50
5.2. Pantalla principal	51
5.3. Pantalla inicio de sesión	51
5.4. Pantalla de registro	52
5.5. Listado de zonas de baño	52
5.6. Hamburger Menu	52
5.7. Filtros	52

5.8. Listado de zonas de baño con filtros aplicados	53
5.9. Listado de zonas de baño favoritas	53
5.10. Pantalla de detalles	53
5.11. Pantalla de alojamientos	53
5.12. Listado de las incidencias de una zona de baño	54
5.13. Pantalla de detalles de una incidencia	54
5.14. Pantalla de nueva incidencia	54
5.15. Pantalla de ajustes	54
5.16. Pantalla de favoritos	55
5.17. Logotipo de la aplicación	56
5.18. Arquitectura general de la aplicación	57
5.19. Decomposition Entities	58
5.20. Decomposition Utils	58
5.21. Decomposition Widgets	59
5.22. Decomposition & Uses Styles Net	59
5.23. Decomposition Views	60
5.24. Decomposition & Uses Styles Accomodation	60
5.25. Decomposition & Uses Styles Account	61
5.26. Decomposition & Uses Styles Comments	61
5.27. Decomposition & Uses Styles Details	61
5.28. Decomposition & Uses Styles Feedback	61
5.29. Decomposition & Uses Styles Home	62
5.30. Decomposition & Uses Styles Legal	62
5.31. Decomposition & Uses Styles Login	62
5.32. Decomposition & Uses Styles Photo	62
5.33. Decomposition & Uses Styles Register	63

5.34. Decomposition & Uses Styles Reset Pass	63
5.35. Decomposition & Uses Styles Upload	63
5.36. Decomposition & Uses Styles Weather	63
5.37. Decomposition & Uses Styles Welcome	63
5.38. Diseño de base de datos	64
5.39. Diagrama de secuencia principal para RF01	66
5.40. Diagrama de secuencia obtener zonas de baño	67
5.41. Diagrama de despliegue	68
6.1. Estructura del proyecto	72
6.2. Estructura de la carpeta de aplicación	73
6.3. Estructura de la carpeta lib dentro de la aplicación	74
A.1. Pantalla principal	114
A.2. Pantalla Iniciar Sesión	115
A.3. Pantalla Registrarse	115
A.4. He olvidado la contraseña	115
A.5. Pantalla Home	116
A.6. Cerrar sesión	116
A.7. Ordenar zonas de baño	117
A.8. Filtrar zonas de baño	117
A.9. Pantalla de detalles	118
A.10.Pantalla de alojamientos	118
A.11.Pantalla de comentarios	118
A.12.Pantalla de favoritos	119
A.13.Pantalla de ajustes	120
A.14.Pantalla de cuenta	120

LISTA DE FIGURAS

A.15.Pantalla de feedback 120

A.16.Pantalla de aspectos legales 120

Lista de Tablas

2.1. RK01: Enfermedad del estudiante	10
2.2. RK02: Desconocimiento de la tecnología a emplear	11
2.3. RK03: Mala planificación del sprint	11
2.4. RK04: Mala planificación del sprint	12
2.5. RK05: Fallos técnicos del equipo de trabajo	12
2.6. RK06: Ausencia temporal de alguno de los tutores	13
2.7. RK07: Superación de los límites gratuitos de Firebase	13
2.8. RK08: Dependencia de terceros y actualizaciones de paquetes	14
2.9. RK09: Uso de datos abiertos	14
2.10. RK10: Cambio de formato de los datos abiertos	15
2.11. RK11: Coincidencia con una asignatura	15
2.12. Planificación inicial	16
2.13. Planificación Evaluación y Rendimiento de Sistemas Software	17
2.14. Product Backlog	17
2.15. Presupuesto simulado	19
2.16. Precio en kWh en España el día 14 de marzo de 2023	20
2.17. Presupuesto real	20
3.1. Historias de usuario de la EP01	22
3.2. Historias de usuario de la EP02	22

3.3. Historias de usuario de la EP03	22
3.4. Historias de usuario de la EP04	23
3.5. Historias de usuario de la EP06	23
3.6. Historias de usuario de la EP05	23
3.7. Requisitos de Información	24
3.8. Requisitos no funcionales	25
3.9. Reglas de negocio	25
6.1. Características Samsung Galaxy A54	75
6.2. Características Samsung Galaxy A03	75
6.3. Características Nokia 6.1	75
6.4. Prueba UAT 01	76
6.5. Prueba UAT 02	76
6.6. Prueba UAT 03	76
6.7. Prueba UAT 04	77
6.8. Prueba UAT 05	77
6.9. Prueba UAT 06	77
6.10. Prueba UAT 07	78
6.11. Prueba UAT 08	78
6.12. Prueba UAT 09	78
6.13. Prueba UAT 10	79
6.14. Prueba UAT 11	79
6.15. Prueba UAT 12	79
6.16. Prueba UAT 13	80
6.17. Prueba UAT 14	80
6.18. Prueba UAT 15	80
6.19. Prueba UAT 16	81

6.20. Prueba UAT 17	81
6.21. Prueba UAT 18	81
6.22. Prueba UAT 19	82
6.23. Prueba UAT 20	82
6.24. Prueba UAT 21	82
6.25. Prueba UAT 22	83
6.26. Prueba UAT 23	83
6.27. Prueba UAT 24	83
6.28. Prueba UAT 25	84
6.29. Prueba UAT 26	84
6.30. Prueba UAT 27	84
6.31. Prueba UAT 28	85
6.32. Prueba UAT 29	85
6.33. Prueba UAT 30	85
6.34. Prueba UAT 31	86
6.35. Prueba UAT 32	86
6.36. Prueba UAT 33	86
6.37. Prueba UAT 34	87
6.38. Prueba UAT 35	87
6.39. Prueba UAT 36	87
6.40. Fallos detectados en la aplicación	88
6.41. Propuestas presentadas por testers	88
7.1. Tareas realizadas en el Sprint 0	90
7.2. Tareas realizadas en el Sprint 1	91
7.3. Tareas realizadas en el Sprint 2	93
7.4. Tareas realizadas en el Sprint 3	94

7.5. Tareas realizadas en el Sprint 4	95
7.6. Tareas realizadas en el Sprint 5	97
7.7. Tareas realizadas en el Sprint 6	98
7.8. Tareas realizadas en el Sprint 7	99
7.9. Coste simulado final	101
7.10. Coste real final	101

Capítulo 1

Introducción

1.1. Contexto

En la comunidad autónoma de Castilla y León, el verano es una época de intenso calor. La Agencia Estatal de Meteorología (AEMET) informó en junio de 2023 que los veranos serían cada vez más largos y más calurosos [1]. Es común que los residentes de esta comunidad y los visitantes que provienen de otras comunidades autónomas busquen refrescarse en las diferentes zonas de baño disponibles en la comunidad, como los diferentes lagos, ríos, embalses o piscinas naturales. Sin embargo, acceder a información actualizada sobre el estado de estas áreas puede resultar más complicado. En la actualidad, se informa al ciudadano sobre el estado de las zonas de baño a través del portal de Salud Pública de la Junta de Castilla y León [14], de los medios de comunicación o de las cuentas oficiales de redes sociales de la Consejería de Sanidad de la Junta de Castilla y León.

Ante esta necesidad, surge la idea de desarrollar una aplicación que pueda proporcionar a los usuarios información acerca del estado de las aguas. Con esta aplicación, los usuarios podrán planificar con antelación sus actividades de ocio, garantizando una experiencia de baño mucho más segura y satisfactoria.

Este proyecto se desarrollará en convenio con Getronics, empresa del sector tecnológico. Esto quiere decir que dicho proyecto estará supervisado por parte de la Universidad de Valladolid y Getronics. De acuerdo con la normativa de Trabajo de Fin de Grado (TFG) y TFG en convenio con empresas, se ha designado un tutor por parte de la Universidad de Valladolid y otro tutor por parte de Getronics. El tutor designado por Getronics tutorizará la parte técnica del proyecto y el tutor de la Universidad de Valladolid tutorizará la parte académica.

1.2. Motivación

La motivación detrás de desarrollar esta aplicación móvil radica en la creación de una herramienta que aborde una necesidad real y relevante para los residentes y visitantes de Castilla y León durante los meses de verano. Con la región siendo hogar de una gran cantidad de zonas de baño naturales y paisajes impresionantes, surge la oportunidad de facilitar a las personas el acceso a información crucial para disfrutar de estas áreas de manera segura y enriquecedora.

La aplicación tiene como objetivo principal proporcionar a los usuarios datos actualizados sobre el estado de las aguas, la disponibilidad de servicios y cualquier riesgo potencial asociado con las zonas de baño habilitadas en la región. Al brindar esta información de manera clara y accesible a través de una plataforma móvil, se busca fomentar una mayor conciencia sobre la seguridad en el agua y promover que los bañistas tomen decisiones informadas sobre dónde y cuándo disfrutar de actividades acuáticas.

1.3. Objetivos

En este proyecto, nos encontraremos dos tipos de objetivos: los objetivos de desarrollo y los objetivos personales.

1.3.1. Objetivos de desarrollo

El principal objetivo de desarrollo es el desarrollo de una aplicación móvil disponible para Android, que disponga de las siguientes funcionalidades:

1. Consulta del estado de las zonas de baño de Castilla y León.
2. Consulta de detalles de una zona de baño.
3. Guardar zonas de baño favoritas y permitir consultarlas.
4. Realizar comentarios en las zonas de baño.

1.3.2. Objetivos personales

Mis objetivos personales durante el desarrollo del proyecto son los siguientes:

1. Mejorar el conocimiento de Flutter con un proyecto iniciado desde cero.
2. Utilizar algunos de los principales servicios de Firebase, que no he utilizado previamente.

3. Aprender a aplicar de manera real un marco de trabajo ágil basado en Scrum.
4. Desarrollar una mayor habilidad en la experiencia de usuario y en el diseño de aplicaciones.
5. Publicar una aplicación en Google Play.

1.4. Estructura de la memoria

Este documento se estructura de la siguiente forma:

Capítulo 2 Requisitos y planificación: En este capítulo se detallará las tareas realizadas antes del inicio del proyecto. Esto incluye el marco teórico de la metodología de trabajo ágil SCRUM, el análisis de riesgos, la planificación, el Product Backlog y, finalmente, el presupuesto.

Capítulo 3 Análisis: En este capítulo, se detalla las tareas llevadas a cabo durante la fase de análisis. Esto incluye los requisitos funcionales, los requisitos no funcionales, los requisitos de información, las reglas de negocio. Además, se incluyen tres diagramas: el modelo de casos de uso, el modelo de dominio y el modelo de las zonas de baño como maquina de estados.

Capítulo 4 Tecnologías utilizadas: En este capítulo se especifica todas las herramientas y tecnologías que se han empleado durante el desarrollo de este proyecto.

Capítulo 5 Diseño: En este capítulo se especificará algunos detalles sobre la fase de diseño del proyecto. Entre ellos se encuentran: las decisiones de diseño, la arquitectura que usa la aplicación, un diseño inicial de la interfaz de usuario de la aplicación y la imagen de marca. Además, se han incluido diagramas de la fase de diseño como son el diagramas sobre el diseño arquitectónico de la aplicación, el diseño de la base de datos, el diagrama de despliegue y el diagrama de comunicación (secuencia).

Capítulo 6 Implementación y pruebas: En este capítulo se detallará algunos aspectos relacionados con la fase de implementación de la aplicación. Además, se incluyen las pruebas llevadas a cabo.

Capítulo 7 Seguimiento del proyecto: En este capítulo se detalla cómo se ha desarrollado el proyecto Sprint a Sprint y las diferentes incidencias en el desarrollo del proyecto que han tenido lugar.

Capítulo 8 Conclusiones: En este capítulo se expondrá las conclusiones finales del desarrollo del proyecto. Además, se expondrá algunas propuestas que podrían llevarse a cabo a corto y medio plazo.

Anexo A Manuales: Incluye manuales de mantenimiento, instalación, despliegue, y uso.

Anexo B Resumen de enlaces adicionales: Incluye enlaces de interés sobre el proyecto, entre los que se incluye el repositorio de código, el enlace a Google Play para descargar la aplicación y el enlace para acceder a página web de la aplicación.

Capítulo 2

Requisitos y Planificación

2.1. Scrum

Scrum [106] es un marco de trabajo ágil que proporciona una forma de realizar un trabajo o proyecto dividiéndolo en partes más pequeñas en las que se tiene unos bucles continuos de experimentación y feedback a la vez que se aprende y se mejora. Scrum ayuda a las personas y a los equipos a proporcionar un valor incremental de forma colaborativa.

El ciclo de Scrum se puede observar en la Figura 2.1.

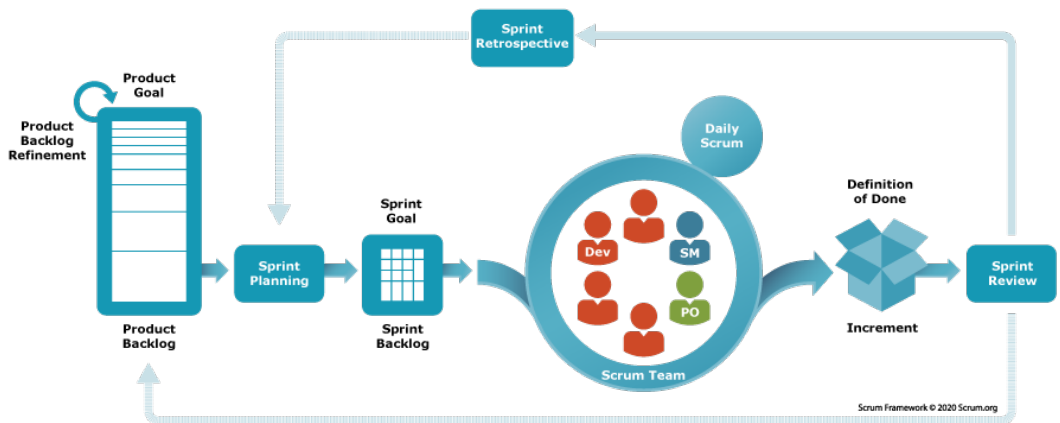


Figura 2.1: Ciclo de vida de Scrum. Tomada de [91].

Scrum [88] se basa en el empirismo, que afirma que el conocimiento proviene de la expe-

riencia, y el pensamiento Lean, que se centra en lo esencial.

Los pilares de Scrum [88] son los siguientes:

1. **Transparencia:** El proceso y el trabajo emergentes debe ser visible para el que realiza el trabajo como para el que recibe el trabajo.
2. **Inspección:** El progreso hacia los objetivos acordados debe ser inspeccionados con frecuencia para detectar posibles problemas.
3. **Adaptación:** El proceso o los entregables que se producen deben poder ajustarse si se detecta un desvío fuera de los límites aceptables o si no se considera aceptable el producto resultante.

Scrum [88, 106, 93] presenta los siguientes valores:

1. **Coraje:** Los miembros del equipo de Scrum tiene coraje para hacer lo correcto y resolver los problemas difíciles.
2. **Enfoque:** Todo el mundo debe enfocarse en el Sprint y en los objetivos del equipo Scrum.
3. **Compromiso:** Las personas tienen el compromiso personal de lograr los objetivos del equipo Scrum.
4. **Respeto:** Los miembros del equipo Scrum se respetan entre ellos para ser personas capaces e independientes.
5. **Apertura:** El Equipo Scrum y los stakeholders se comprometen a ser abiertos acerca de todo el trabajo y los desafíos durante la realización del trabajo.

En las siguientes subsecciones se explicará en detalle los elementos de Scrum.

2.1.1. El equipo Scrum

La unidad fundamental de Scrum [92] es un pequeño equipo de personas, el equipo Scrum. El equipo Scrum está formado por los desarrolladores, el propietario del producto (Product Owner) y el Scrum Master. A continuación, se explicará un poco más en profundidad cada uno de los miembros.

Desarrolladores

Los desarrolladores [95] son las personas que se encargan de crear cualquier aspecto de un incremento en un sprint. Los desarrolladores son responsables de las siguientes actividades:

- Crear un plan para el sprint, el Sprint Backlog
- Inculcar la calidad
- Adaptar su plan cada día hacia el Objetivo Sprint
- Responsabilizarse como profesionales.

Propietario del producto (Product Owner)

El propietario del producto [98] es el responsable de maximizar el valor del producto resultante. Algunas de las tareas de las que es responsable el propietario del producto son:

- El desarrollo y la comunicación del objetivo del producto.
- La creación y la comunicación clara de elementos pendientes del producto.
- Asegurarse de que el trabajo pendiente cumpla con los valores de Scrum.

Scrum Master

El Scrum Master [99] es el responsable de establecer Scrum tal y como se define en la guía Scrum [88]. Es responsable de la efectividad del equipo Scrum y son los verdaderos líderes de la organización.

2.1.2. Eventos de Scrum

Los eventos de Scrum [90] son la oportunidad para inspeccionar y adaptar los artefactos de Scrum. Están diseñados para conseguir la transparencia requerida.

Dentro de los eventos Scrum nos encontramos con el **Sprint**, el **Sprint Planning**, el **Daily Scrum**, el **Sprint Review** y el **Sprint Retrospective**.

Sprint

El Sprint [100] es una de las partes clave de Scrum. Son eventos de una longitud fija en el que se desarrollan los elementos planificados para ese periodo de tiempo. La duración de un Sprint suele ser de una semana a un mes según sea el equipo de desarrollo y sus posibilidades de entregar valor al final de cada sprint.

Sprint Planning

El Sprint Planning [107] es la que se encarga de inicializar el sprint estableciendo el trabajo a realizar durante ese sprint. La planificación de sprint debe abordar los siguientes temas:

1. ¿Por qué el sprint es valioso?
2. ¿Qué se puede hacer en este sprint?
3. ¿Cómo se realizará el trabajo elegido?

Además, se debe establecer el objetivo del sprint, los elementos de trabajo pendientes del producto y el plan para entregarlos. Esto recibe el nombre de **Sprint Backlog**

Daily Scrum

El Daily Scrum [105] se utiliza para inspeccionar el progreso hacia el objetivo del Sprint y adaptar el Sprint Backlog. La duración máxima de este evento es de 15 minutos y los participantes son los desarrolladores del proyecto. Se realiza durante todos los días laborables en los que se este desarrollando el sprint.

Sprint Review

El Sprint Review [108] se utiliza para inspeccionar el resultado del sprint y determinar futuras adaptaciones de cara a los sucesivos sprints.

Sprint Retrospective

El Sprint Retrospective [103] se utiliza para planificar formas de aumentar la calidad y la eficiencia. En este evento, se inspecciona como fue el último sprint. Se analiza qué fue bien, qué problemas se encontraron y cómo fueron resueltos esos problemas, en caso de que fueran resueltos. Este evento concluye el sprint.

2.1.3. Artefactos de Scrum

Los artefactos de Scrum [89] son los que representan el trabajo y el valor y están diseñados para maximizar la transparencia de la información clave.

Los artefactos que veremos en esta sección son el **Product Backlog**, el **Product Goal**, el **Sprint Backlog**, el **Sprint Goal**, el **Increment** y **Definition of Done**.

Product Backlog

El Product Backlog [96] es una lista emergente y ordenada de lo que se necesita para mejorar el trabajo. Los elementos dentro de esta lista son candidatos a ser seleccionados durante el Sprint Planning.

Product Goal

El Product Goal [97] es el estado futuro del producto que puede servir como objetivo para el equipo Scrum.

Sprint Backlog

El Sprint Backlog [101] se compone del objetivo del sprint, el conjunto de elementos de trabajo pendiente del producto seleccionado para el Sprint y un plan accionable para entregar el incremento.

Sprint Goal

El Sprint Goal [102] es el único objetivo para el Sprint. Este objetivo se debe agregar al trabajo pendiente del Sprint (Sprint Backlog).

Incremento

Un Incremento [104] es un paso hacia el Product Goal. Cada incremento es aditivo a todos los incrementos anteriores.

Se pueden crear varios incrementos dentro de un Sprint. Esos incrementos se presentan en Sprint Review.

Definition of Done

Definition of Done [94] es una descripción formal del incremento cuando cumple con las medidas de calidad requeridas para el producto.

2.1.4. Adaptación de Scrum al proyecto

Para la realización de este proyecto, se va a adaptar Scrum para poder trabajar de acuerdo a las condiciones existentes tanto en el ámbito profesional como en el ámbito académico.

El equipo Scrum estará formado por tres personas: el alumno, que asumirá los roles de desarrollador y Product Owner y los dos tutores, tanto de la Universidad como de la empresa, que asumirán el rol de Scrum Master.

En cuanto a los eventos Scrum, los sprints tendrán una duración fija de dos semanas. En total habrá siete sprints con una carga aproximada de 40 horas de trabajo por sprint más 20 horas adicionales de trabajo en el sprint 0. En cuanto al desarrollo de los eventos de *Sprint Planning*, *Sprint Review* y *Sprint Retrospective* se desarrollarán cada dos semanas. Dadas las características de este proyecto, se hace imposible desarrollar una *Daily Scrum* (reunión de Scrum diaria), por lo que se va a adaptar a una *Weekly Scrum*, evento de características similares a la *Daily Scrum* pero que se desarrollará una vez por semana, los jueves. Cada dos jueves coincidirá con los eventos de *Sprint Review*, *Sprint Retrospective* y *Sprint Planning*.

2.2. Análisis de riesgos

Un riesgo [46] es la probabilidad de que un peligro pueda darse y perjudicar a uno o más individuos y al desarrollo del proyecto.

Para este proyecto, se identifican dos clases de riesgos: los riesgos generales que se pueden dar en cualquier proyecto de características similares y los riesgos particulares que se dan en este proyecto.

2.2.1. Riesgos generales

Los riesgos generales de este proyecto se encuentran entre las Tablas 2.1 y 2.6.

Identificador	RK01
Título	Enfermedad del estudiante
Descripción	El estudiante enferma y le es imposible continuar el desarrollo del proyecto
Probabilidad	Baja
Impacto	Alto
Nivel de riesgo	Medio
Mitigación	1. Intentar llevar una vida saludable 2. Evitar situaciones de riesgo en la que exista una mayor probabilidad de caer enfermos, como estar cerca de una persona enferma
Contingencia	1. Hacer uso de los sprints extra en caso de que fuera necesario 2. Hacer replanificación de las tareas de los sprints siguientes

Tabla 2.1: RK01: Enfermedad del estudiante

Identificador	RK02
Título	Desconocimiento de la tecnología a emplear
Descripción	La tecnología que se va a emplear en el desarrollo de este proyecto es desconocida por los miembros del equipo de desarrollo
Probabilidad	Baja
Impacto	Bajo
Nivel de riesgo	Bajo
Mitigación	1. Realizar formación con la tecnología que se va a utilizar antes de empezar el proyecto
Contingencia	1. Consultar la documentación oficial de Flutter, Firebase u otras tecnologías que se utilizarán en el proyecto 2. Utilizar foros de terceros para resolver los diferentes problemas que vayan surgiendo.

Tabla 2.2: RK02: Desconocimiento de la tecnología a emplear

Identificador	RK03
Título	Mala planificación del sprint
Descripción	Se ha realizado una mala planificación del sprint y no se ha podido alcanzar los objetivos planificados para dicho Sprint
Probabilidad	Baja
Impacto	Bajo
Nivel de riesgo	Bajo
Mitigación	1. Realizar un proceso de planificación claro donde se incluya la priorización adecuada de las tareas
Contingencia	1. Mantener un margen de tiempo adicional al final del proyecto para abordar tareas pendientes.

Tabla 2.3: RK03: Mala planificación del sprint

2.2. ANÁLISIS DE RIESGOS

Identificador	RK04
Título	Cambios en los requisitos
Descripción	El product owner o el cliente realizan cambios en los requisitos del proyecto.
Probabilidad	Alta
Impacto	Alto
Nivel de riesgo	Alto
Mitigación	<ol style="list-style-type: none">1. Establecer un proceso formal de gestión de cambios que incluya la documentación detallada de los requisitos iniciales, la revisión regular con el cliente y la aprobación formal de cualquier cambio.2. Educar al cliente sobre el impacto de los cambios en el cronograma y en el presupuesto del proyecto.
Contingencia	<ol style="list-style-type: none">1. Mantener un equipo de desarrollo flexible y preparado para adaptarse a los requisitos cambiantes del producto.

Tabla 2.4: RK04: Mala planificación del sprint

Identificador	RK05
Título	Fallos técnicos del equipo de trabajo
Descripción	Debido a un fallo en el material técnico nos es imposible continuar con el desarrollo del proyecto
Probabilidad	Baja
Impacto	Alto
Nivel de riesgo	Bajo
Mitigación	<ol style="list-style-type: none">1. Depositar el trabajo realizado en un repositorio online2. Trabajar en la memoria en una herramienta online3. Tener copias de seguridad de los archivos
Contingencia	<ol style="list-style-type: none">1. Obtener otro equipo donde se pueda continuar realizando el trabajo

Tabla 2.5: RK05: Fallos técnicos del equipo de trabajo

Identificador	RK06
Título	Ausencia temporal de alguno de los tutores
Descripción	No se puede continuar con el seguimiento del proyecto debido a la ausencia de algunos de los tutores
Probabilidad	Media
Impacto	Bajo
Nivel de riesgo	Bajo
Mitigación	1. Mantener el contacto a través de medios electrónicos para saber cuando se puede dar el caso.
Contingencia	1. Continuar con la realización del trabajo hasta que se pueda mantener una reunión con el tutor afectado. 2. Realizar comunicación a través de una herramienta que permita conversación asíncrona.

Tabla 2.6: RK06: Ausencia temporal de alguno de los tutores

2.2.2. Riesgos particulares

Los riesgos particulares de este proyecto se desarrollan entre las Tablas 2.7 y 2.11

Identificador	RK07
Título	Superación de los límites gratuitos de Firebase
Descripción	Se superan los límites gratuitos que tiene Firebase y se corta el acceso al servicio hasta que se pague por el plan superior.
Probabilidad	Bajo
Impacto	Alto
Nivel de riesgo	Bajo
Mitigación	1. Realizar un análisis de los límites gratuitos que ofrece Firebase. 2. Establecer una monitorización del estado de Firebase 3. Considerar opciones de cambio de plan si se prevé un crecimiento significativo
Contingencia	1. Monitorizar el uso y los límites gratuitos para que cuando se detecte que se va a sobrepasar dichos límites se pueda actualizar a otro plan 2. Reservar un parte del presupuesto para este fin

Tabla 2.7: RK07: Superación de los límites gratuitos de Firebase

2.2. ANÁLISIS DE RIESGOS

Identificador	RK08
Título	Dependencia de terceros y actualizaciones de paquetes
Descripción	La aplicación va a utilizar paquetes de terceros que han sido desarrollados por otros usuarios de la comunidad Flutter. Esto puede tener problemas de compatibilidad o falta de mantenimiento de dichos servicios o paquetes.
Probabilidad	Alta
Impacto	Medio
Nivel de riesgo	Medio
Mitigación	<ol style="list-style-type: none">1. Realizar una evaluación cuidadosa de los paquetes de terceros a utilizar2. Mantener un registro de las dependencias y realizar actualizaciones regulares de los paquetes
Contingencia	<ol style="list-style-type: none">1. Tener un plan de respaldo en caso de que un paquete crítico deje de ser mantenido.

Tabla 2.8: RK08: Dependencia de terceros y actualizaciones de paquetes

Identificador	RK09
Título	Uso de datos abiertos
Descripción	La aplicación va a utilizar datos abiertos provenientes de la Junta de Castilla y León. Estos datos pueden quedar desactualizados.
Probabilidad	Media
Impacto	Alto
Nivel de riesgo	Medio
Mitigación	<ol style="list-style-type: none">1. Realizar una evaluación exhaustiva de los conjuntos de datos abiertos2. Solicitar a la Junta de Castilla y León su actualización.
Contingencia	<ol style="list-style-type: none">1. Implementar mecanismos de recuperación de esos datos de fuentes alternativas.

Tabla 2.9: RK09: Uso de datos abiertos

Identificador	RK10
Título	Cambio de formato de los datos abiertos
Descripción	El formato de los datos proporcionados puede cambiar en un futuro
Probabilidad	Alta
Impacto	Alto
Nivel de riesgo	Alto
Mitigación	1. Mantener la comunicación con el proveedor de datos
Contingencia	1. Establecer un proceso de monitorización continuo de las fuentes. 2. Tener preparado un proceso para realizar ajustes en el código en caso de que se produzca un cambio de formato de datos.

Tabla 2.10: RK10: Cambio de formato de los datos abiertos

Identificador	RK11
Título	Coincidencia con una asignatura del grado
Descripción	El desarrollo del proyecto se va a compatibilizar con una asignatura pendiente del grado. Algunas de las actividades planificadas pueden afectar al correcto desarrollo del proyecto.
Probabilidad	Alta
Impacto	Alta
Nivel de riesgo	Alta
Mitigación	1. Realizar una planificación detallada de acuerdo con los horarios y trabajos requeridos por dicha asignatura.
Contingencia	1. Reservar tiempo a mayores en los sprints siguientes al desarrollo de una tarea de la asignatura.

Tabla 2.11: RK11: Coincidencia con una asignatura

2.3. Planificación y calendarización

La guía docente del Trabajo Fin de Grado [113] establece que al desarrollo de este proyecto le corresponde 12 ECTS, lo que equivale a 300 horas.

Siguiendo el marco de trabajo ágil basado en Scrum, se ha dividido el desarrollo de este proyecto en 7 sprints más 2 sprints extras que solo se utilizarán en caso de que fuera necesario. La duración de los sprints será de 2 semanas por sprint. Se prevé un descanso en el proyecto durante el periodo de vacaciones de Semana Santa, cuyo periodo abarca desde el 23 de marzo de 2024 al 2 de abril de 2024, según el calendario académico de la Universidad de Valladolid [18].

Se utilizará, inicialmente un Sprint 0 [109] en el que se realizarán una serie de tareas preparatorias de cara al Sprint 1. Entre ellas, se encuentra la preparación del entorno, la

2.3. PLANIFICACIÓN Y CALENDARIZACIÓN

definición de la arquitectura, la recopilación de los requisitos iniciales, la identificación y preparación de las historias de usuario y la planificación del proyecto.

La estimación de trabajo por semana es de media **20 horas por semana** por lo que sumadas las horas de trabajo establecidas en el sprint 0 serían un total de **300 horas**.

En la Tabla 2.12 podemos observar la planificación inicial del desarrollo del proyecto.

Evento	Fecha Inicio	Fecha fin
Sprint 0	20/02/2024	29/02/2024
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	29/02/2024	
Sprint 1	29/02/2024	14/03/2024
Scrum Weekly	07/03/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	14/03/2024	
Sprint 2	14/03/2024	04/03/2024
Scrum Weekly	21/03/2024	
Vacaciones de Semana Santa	23/03/2024	02/04/2024
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	04/04/2024	
Sprint 3	04/04/2024	18/04/2024
Scrum Weekly	11/04/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	18/04/2024	
Sprint 4	18/04/2024	02/05/2024
Scrum Weekly	25/04/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	02/05/2024	
Sprint 5	02/05/2024	16/05/2024
Scrum Weekly	09/05/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	16/05/2024	
Sprint 6	16/05/2024	30/05/2024
Inicio solicitud de defensa 1ª convocatoria	20/05/2024	
Scrum Weekly	23/05/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective y Sprint Planning	30/05/2024	
Sprint 7	30/05/2024	13/06/2024
Scrum Weekly	06/05/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective	13/06/2024	
Sprint Extra 1	13/06/2024	27/06/2024
Fecha límite de solicitud de defensa 1ª convocatoria	20/06/2024	
Scrum Weekly	20/06/2024	
Inicio solicitud defensa 2ª convocatoria	24/06/2024	
Scrum Weekly, Sprint Review, Sprint Retrospective	27/06/2024	
Sprint Extra 2	27/06/2024	04/07/2024
Scrum Weekly, Sprint Review, Sprint Retrospective	04/07/2024	
Fecha límite de solicitud de defensa 2ª convocatoria	07/07/2024	

Tabla 2.12: Planificación inicial

Un aspecto importante a tener en cuenta es que el desarrollo de este proyecto se compatibiliza con un trabajo a media jornada (20 horas por semana) y la asignatura Evaluación y Rendimiento de Sistemas Software del grado. Esta asignatura, de acuerdo con el calendario de actividades [16] y el calendario de exámenes [17], tiene la programación que se puede observar en la Tabla 2.13, que puede tener alguna incidencia en el desarrollo normal del

proyecto:

Actividad	Fecha entrega o realización
Practica Individual I	15/03/2024
Practica Grupal	26/04/2024
Practica Individual II	29/05/2024
Examen Ordinario	05/06/2024
Examen Extraordinario	20/06/2024

Tabla 2.13: Planificación Evaluación y Rendimiento de Sistemas Software

2.4. Product Backlog

En la Tabla 2.14 se puede observar las Épicas del proyecto que se va a desarrollar. Se identifican con **EPXX** donde XX es el número que identifica a la épica.

ID Épica	Descripción
EP01	Como usuario quiero poder ver las zonas de baño disponibles para poder identificar las distintas opciones disponibles
EP02	Como usuario quiero poder ver los detalles de las zonas de baño disponibles para tener más información acerca de las zonas de baño y otros detalles
EP03	Como usuario quiero poder publicar un comentario en una zona de baño para ayudar a otros usuarios en su elección
EP04	Como usuario quiero poder aplicar diferentes ajustes para poder personalizar la aplicación.
EP05	Como usuario quiero poder usar una cuenta opcionalmente para tener acceso a funciones extra de la aplicación
EP06	Como desarrollador de la aplicación quiero poder avisar a los usuarios cuando haya una actualización de la aplicación disponible para que los usuarios tengan la última versión

Tabla 2.14: Product Backlog

2.5. Presupuesto

Como última fase de la planificación, se han elaborado dos presupuestos diferentes: un primer presupuesto que sería el presupuesto simulado en caso de que este proyecto se desarrollara fuera del ámbito académico (ver Sección 2.5.1), y un segundo presupuesto que es el presupuesto real, ya que este proyecto se desarrolla dentro de un ámbito académico. (ver Sección 2.5.2).

Se debe aclarar que no se contempla la elaboración de un presupuesto de cara a presentarlo un cliente sino que lo presupuestado sería una estimación de los costes de desarrollo del proyecto.

2.5.1. Presupuesto simulado

En primer lugar, nos centraríamos en los empleados. Necesitaríamos un desarrollador Full Stack [10], que se encargaría tanto del desarrollo del front-end como del back-end. La horquilla de sueldo base de un desarrollador Full Stack en España [29] se encuentra entre los 25.000€ y los 39.000€. Nosotros adoptaremos como sueldo bruto 25.000€ anuales. Teniendo en cuenta que el sueldo se reparte en 14 pagas, el salario bruto sería de **1.785,71€** por mes. A mayores, necesitaríamos un Product Owner y un Scrum Master. La horquilla de sueldo base de un Product Owner [30] está entre los 40.000€ y los 53.000€. Nosotros adoptaremos los 40.000€ brutos anuales. El sueldo bruto mensual sería **2.857,14€**. En cuanto al Scrum Master, la horquilla de sueldo base [31] estaría entre 36.000€ y 50.000€. Nosotros adoptaremos los 36.000€ brutos anuales. El sueldo bruto mensual sería de **2.571,42€**.

Respeto al material técnico, se va a utilizar un ordenador portátil *HP LAPTOP 15-DA0071NS* [114] cuyo precio es de **653,94 €** con IVA incluido. Además, debido a las diferencias que existen en Android entre versiones, se va a utilizar 3 teléfonos para realizar las pruebas de la aplicación. Los modelos elegidos han sido elegido el *Nokia 6.1* [44], con Android 10, cuyo precio es de **99€**; **Samsung Galaxy A03** [86], con Android 13, cuyo precio es de **150€** y **Samsung Galaxy A54** [87], con Android 14, cuyo precio es de **499€**. Se estima la vida útil de los 3 teléfonos y del portátil en 2 años. Por tanto, dispondríamos de los precios unitarios que se pueden ver en la Tabla 2.15

Respecto a las herramientas que se van a utilizar, se utilizarán de forma gratuita Android Studio [32], Firebase [33], Sourcetree [5] y Jira [3]. Justinmind [40] tiene un coste de 9\$ por mes, que en euros a cambio actual son **8,53€** por mes. Visual Paradigm Professional [54] tiene un coste de 35\$ por mes, que en euros a cambio actual supondría **31,99€** por mes. Además para comunicación, necesitaremos Microsoft Teams [49] cuyo precio por mes es de 4\$ por mes, que en euros a cambio actual supondría **3,66€** por mes.

Al querer publicar la aplicación en Google Play y tratarse de la primera aplicación que voy a publicar en la tienda de Google, se exige el pago de una tasa al crear una cuenta de desarrollador. El coste de esta tasa es de 25\$. El cambio en euros es de **23,81€**.

Se va a emplear una modalidad de teletrabajo 100 %. En este caso se aplica la *Ley 10/2021, de 9 de julio, de trabajo a distancia* [19], que establece en el Artículo 11 que el desarrollo del teletrabajo deberá ser compensado o sufragado por la empresa. Siguiendo la *Resolución de 13 de julio de 2023, de la Dirección General de Trabajo, por la que se registra y publica el XVIII Convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública*. [20], el complemento por teletrabajo es de **17€** mensuales.

Por tanto, teniendo en cuenta que la previsión de este proyecto es de **4 meses**, obtenemos un coste total de **29.495,31€**. A este presupuesto le sumaremos un 25 % adicional por si hay

que hacer frente a algún imprevisto. Los detalles desglosados se muestran en la Tabla 2.15.

Concepto	Precio Unitario	Cantidad	Total
Sueldo Full Stack	1.785,71€/mes	4 meses	7.142,84€
Sueldo Product Owner	2.857,14€/mes	4 meses	11.428,56€
Sueldo Scrum Master	2.571,42€/mes	4 meses	10.285,68€
Equipo PC	27,25€/mes	4 meses	109€
Nokia 6.1	4,13€/mes	4 meses	16,5€
Samsung Galaxy A03	6,25€/mes	4 meses	25€
Samsung Galaxy A54	20,8€/mes	4 meses	83,2€
Complemento teletrabajo	51€/mes	4 meses	204€
Subscripción Justinmind	8,53€/mes	4 meses	34,12€
Subscripción Visual Paradigm	31,99€/mes	4 meses	127,96€
Subscripción Microsoft Teams	3,66€/mes	4 meses	14,64€
Tasa Google Play			23,81€
Total			29.495,31€
+25 % previsión de riesgos			36.869,14€

Tabla 2.15: Presupuesto simulado

2.5.2. Presupuesto real

Al tratarse de un Trabajo Fin de Grado, en el presupuesto real se prescinde de algunos de los costes mencionados anteriormente.

El rol de Full Stack Developer y Product Owner lo asume el estudiante y no recibirá remuneración. Al igual que el rol de Scrum Master. Tampoco hay complemento por teletrabajo, por lo que se puede prescindir de ese coste.

En cuanto a las herramientas de trabajo, el ordenador y los dispositivos móviles ya fueron adquiridos previamente antes del inicio de este trabajo. La Universidad de Valladolid proporciona licencia para Visual Paradigm y para Microsoft Teams. Se utilizará la versión gratuita de Justinmind para el desarrollo de prototipos.

Se mantiene el coste de la tasa de Google Play, debido a que es la primera aplicación que publico en Google Play.

Se debe tener en cuenta el precio actual de la luz. Se toma como referencia el precio de la luz del 13 de marzo de 2024. El precio de la luz [110] entre las 15:00 y las 21:00, que es el horario en el que se estima que es donde se va a trabajar en este proyecto, se puede observar en la Tabla 2.16.

2.5. PRESUPUESTO

Precio kWh en España (14/03/2024)	
15:00	0.06472 €/kWh
16:00	0.06706 €/kWh
17:00	0.08569 €/kWh
18:00	0.19597 €/kWh
19:00	0.22182 €/kWh
20:00	0.26379 €/kWh
Media	0,149842 €/kWh

Tabla 2.16: Precio en kWh en España el día 14 de marzo de 2023

Teniendo en cuenta el precio de la luz podemos ver el desglose de lo que sería el presupuesto simulado en la Tabla 2.17.

Concepto	Precio Unitario	Cantidad	Total
Electricidad	0,149842 €/kWh	300 kWh	44,95€
Tasa Google Play			23,81€
Equipo PC	27,25€/mes	4 meses	109€
Nokia 6.1	4,13€/mes	4 meses	16,5€
Samsung Galaxy A03	6,25€/mes	4 meses	25€
Samsung Galaxy A54	20,8€/mes	4 meses	83,2€
Total			302,46€

Tabla 2.17: Presupuesto real

Capítulo 3

Análisis

En este capítulo, se abordará lo relativo a la fase de análisis del sistema. Esto incluye los requisitos funcionales, que son las acciones que debe realizar el sistema; los requisitos no funcionales, que establecen las restricciones; requisitos de información, que describen los datos necesarios y las reglas de negocio. Adicionalmente, se incluye diagramas como el modelo de dominio, el modelo de casos de uso y un modelo de máquina de estados.

3.1. Stakeholders

Podemos identificar los siguientes stakeholders dentro de nuestra aplicación:

- **Usuario:** Es cualquier persona que pueda utilizar la aplicación tenga o no tenga una cuenta creada.
- **Usuario registrado:** Es cualquier persona que utilice la aplicación y que tenga una cuenta creada.
- **Equipo de desarrollo:** Será el stakeholder encargado del proceso de desarrollo y del mantenimiento de la aplicación.

Las historias de usuario se describirán de la siguiente forma: *Como <stakeholder/tipo de usuario> quiero <objetivo> para <valor que aporta la historia de usuario>*

3.2. Requisitos funcionales como historias de usuario

A partir del Product Backlog detallado en la Sección 2.4 de este documento, se pueden extraer los **requisitos funcionales como historias de usuario**. Se enumeran los requisitos

3.2. REQUISITOS FUNCIONALES COMO HISTORIAS DE USUARIO

funcionales agrupados según las épicas. Como identificador se utiliza la nomenclatura **RFXX** donde XX es el número que identifica al requisito funcional.

ID Historia Usuario	Descripción
RF01	Como usuario quiero poder ver un listado de las zonas de baño disponibles para poder identificar las distintas opciones
RF02	Como usuario quiero poder filtrar el listado de zonas de baño disponible por provincia para poder identificar las zonas de baño disponibles en cada provincia
RF03	Como usuario quiero poder filtrar el listado de las zonas de baño por estado para poder identificar las zonas de baño por estado
RF04	Como usuario registrado quiero poder filtrar el listado de las zonas de baño por favoritos para poder identificar las zonas de baño que he añadido a favoritos.

Tabla 3.1: Historias de usuario de la EP01

ID Historia Usuario	Descripción
RF05	Como usuario quiero poder ver el nombre de la zona de baño, su estado, una pequeña descripción sobre la localidad, fotos proporcionadas por usuarios registrados y la previsión del tiempo para tener información adicional sobre la zona de baño
RF06	Como usuario quiero poder recibir indicaciones al lugar indicado para poder llegar a la zona de baño
RF07	Como usuario quiero poder reservar un alojamiento cercano para poder alojarme en las inmediaciones de la zona de baño

Tabla 3.2: Historias de usuario de la EP02

ID Historia Usuario	Descripción
RF08	Como usuario registrado quiero poder ver los comentarios flash o permanentes publicados por otros usuarios para poder tener mayor información a la hora de decidir
RF09	Como usuario registrado quiero poder publicar un comentario en una zona de baño para ayudar a otros usuarios en su elección

Tabla 3.3: Historias de usuario de la EP03

ID Historia Usuario	Descripción
RF10	Como usuario quiero poder cambiar el idioma de la aplicación para poder entender el contenido de la aplicación en mi lengua materna.
RF11	Como usuario quiero saber si hay actualizaciones de la aplicación para ver si estoy en la última versión de la aplicación.
RF12	Como usuario quiero poder contactar con el desarrollador para notificar errores en la aplicación o para proporcionar comentarios o sugerencias.

Tabla 3.4: Historias de usuario de la EP04

ID Historia Usuario	Descripción
RF13	Como equipo de desarrollo de la aplicación quiero poder avisar a los usuarios cuando haya una actualización de la aplicación disponible para que los usuarios tengan la última versión

Tabla 3.5: Historias de usuario de la EP06

ID Historia Usuario	Descripción
RF14	Como usuario registrado quiero poder iniciar sesión en la aplicación con mi correo electrónico y una contraseña para poder disfrutar de las características extra de la aplicación.
RF15	Como usuario registrado quiero poder iniciar sesión en la aplicación con mi cuenta de Google para poder disfrutar de las características extra de la aplicación.
RF16	Como usuario quiero poder utilizar la aplicación como invitado para poder usar las características básicas de la aplicación
RF17	Como usuario quiero poder registrarme en la aplicación para poder obtener una cuenta
RF18	Como usuario registrado quiero poder eliminar mi cuenta y datos asociados con mi cuenta
RF19	Como usuario registrado quiero poder modificar alguno de los datos asociados a mi cuenta
RF20	Como usuario registrado quiero poder ver y modificar las zonas de baño añadidas favoritos
RF21	Como usuario registrado quiero poder añadir zonas de baño a favoritos
RF22	Como usuario registrado quiero poder añadir imágenes a una zona de baño

Tabla 3.6: Historias de usuario de la EP05

3.3. Requisitos de información como historias de usuario

Los requisitos funcionales de información[15] son los que especifican la funcionalidad que debe quedar almacenada en el sistema.

Los requisitos de información identificados se muestran en la tabla 3.7. Estos se identificarán con la forma **RIXX** donde XX es el número que identifica al requisito de información.

ID	Descripción
RI01	Como equipo de desarrollo quiero guardar los siguientes datos de un usuario : identificador, nombre de usuario, e-mail, proveedor, foto de perfil y contraseña para poder añadir funcionalidad extra a la aplicación.
RI02	Como equipo de desarrollo quiero guardar los siguientes datos de un comentario : identificador, tipo, nombre de la zona de baño, puntuación y observaciones.
RI03	Como equipo de desarrollo quiero guardar los siguientes datos de un favorito : identificador del usuario, nombre, localidad y provincia de la zona de baño y fecha de registro.

Tabla 3.7: Requisitos de Información

3.4. Requisitos no funcionales

Los requisitos no funcionales [124] son los que describen las restricciones o características del sistema.

Se ha identificado los requisitos no funcionales que se pueden ver en la tabla 3.8. Estos se identificarán con la forma **RNFXX** donde XX es el número que identifica al requisito no funcional.

ID	Descripción
RNF01	Como equipo de desarrollo quiero que la aplicación sea compatible para Android 10 o superior para poder llegar a un mayor número de usuarios
RNF02	Como equipo de desarrollo quiero que la aplicación genere logs para tener un registro de todos los fallos que puedan darse
RNF03	Como equipo de desarrollo quiero que la aplicación muestre mensajes de error cuando se produzca algún fallo pero que pueda seguir funcionando para que la aplicación sea más robusta.
RNF04	Como usuario quiero que la aplicación se encuentre en múltiples idiomas: en español, en inglés y en francés, para que más usuarios puedan entender la aplicación.
RNF05	Como usuario quiero que la aplicación tenga una interfaz de usuario usable para que un usuario promedio de la aplicación pueda utilizar toda su funcionalidad sin necesidad de consultar un manual en 5 minutos.
RNF06	Como usuario quiero que la aplicación tenga transiciones entre pantallas para hacer la aplicación más fluida.

Tabla 3.8: Requisitos no funcionales

3.5. Reglas de negocio

Las reglas de negocio [123] son un tipo de requisitos que se refieren a una serie de políticas, normas y condiciones de las organizaciones que deben quedar satisfechas por nuestro sistema.

Para el sistema desarrollado se identifican las reglas de negocio que se pueden observar en la tabla 3.9. Las reglas de negocio se identificarán con el identificador **BRXX** donde XX es el número que identifica a la regla de negocio.

ID	Descripción
BR01	La temporada oficial de baño está establecida entre el 15 de junio y el 15 de septiembre. Fuera de ese periodo el estado de las zonas de baño debe ser “Fuera de temporada”
BR02	La aplicación será gratuita
BR03	La aplicación se distribuirá a través de Google Play
BR04	Los datos deberán ser tratados de acuerdo al Reglamento General de Protección de Datos (RGDP) de la Unión Europea

Tabla 3.9: Reglas de negocio

3.6. Modelo de dominio

El Figura 3.1 se puede observar el modelo de dominio.

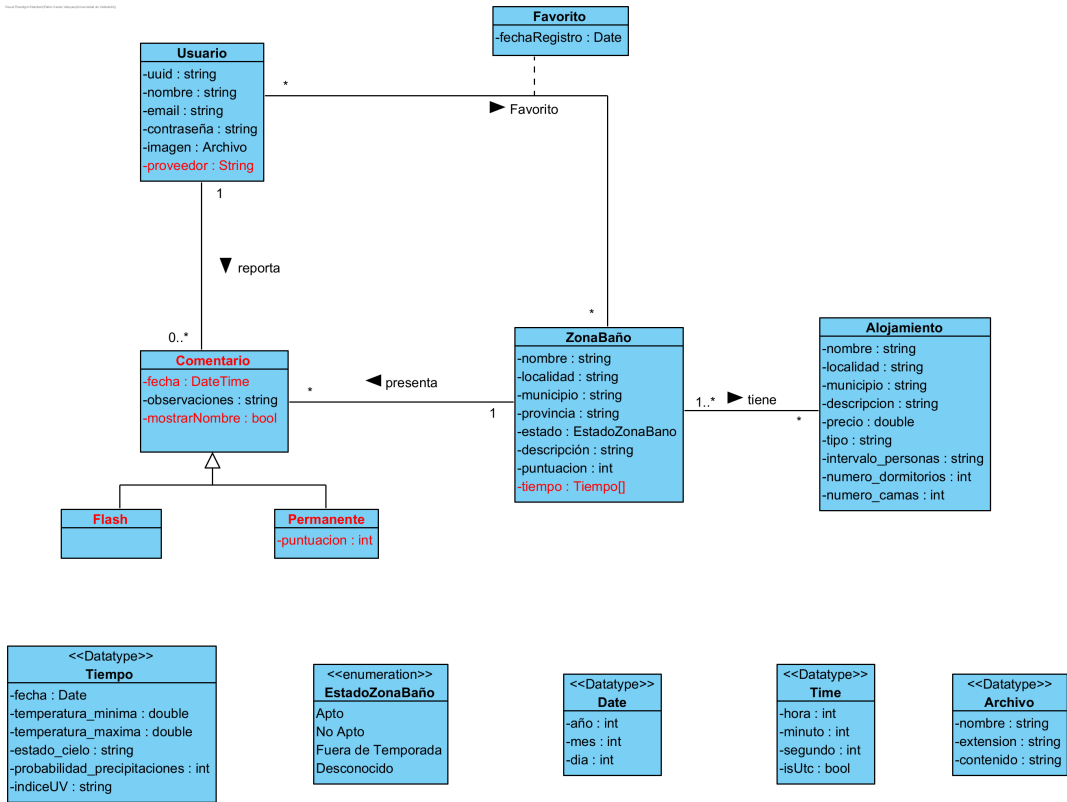


Figura 3.1: Modelo de dominio

Este modelo de dominio que se refleja en esta sección es el resultado final. Se reflejará en rojo las diferencias con respecto al diagrama inicial. Algunos de los cambios que se han modificado con respecto al diagrama inicial han sido:

- En el diagrama inicial, no existía la clase Comentario, sino que era Incidencia. Por tanto se ha producido un cambio de nombre.
- Se han modificado algunos de los atributos de la clase original Incidencia, estos cambios se encuentran reflejados en rojo.
- Se ha añadido las clases Flash y Permanente que representan los dos tipos de comentarios que existen.
- En el Sprint 3, se añadió el apartado para consultar el tiempo. Esto se encuentra reflejado en la zona de baño.

Se ha producido otros cambios en los requisitos que sin embargo no han tenido ningún efecto en el diagrama de dominio inicial.

3.7. Modelo de casos de uso

Dentro de la fase de análisis se ha desarrollado un modelo de casos de uso, donde se analiza los principales casos de uso que nos podemos encontrar en la aplicación que vamos a desarrollar. Esto se puede observar en la Figura 3.2.

Se debe tener en cuenta que los casos de uso “Obtener direcciones de una zona de baño”, “Ver Alojamientos cerca de la zona de baño” y “Consultar la previsión meteorologica” tiene como prerrequisito el caso de uso “Ver detalle de una zona de baño”. Además, todos los casos de uso del stakeholder “Usuario Registrado” tienen como prerrequisito el caso de uso “Identificarse”

Este diagrama se encuentra disponible en el archivo Visual Paradigm disponible en la carpeta *model* del repositorio mencionado en el Apéndice B.

3.7. MODELO DE CASOS DE USO



Figura 3.2: Diagrama de casos de uso de la aplicación

3.8. Modelado de las zonas de baño como máquinas de estado

En el caso de los estados de las zonas de baño, se ha elaborado un diagrama de máquina de estados. En él se detalla que la zona de baño puede tener 5 estados:

- Fuera de Temporada
- Desconocido
- Apta
- No apta
- Agua sin analizar

Debido a la complejidad de las transiciones se ha dividido el diagrama en dos diagramas distintos. Esto se puede observar en las Figuras 3.3 y 3.4.

Estos dos diagramas se pueden observar mejor en el archivo de Visual Paradigm disponible en la carpeta *model* del repositorio mencionado en el Apéndice B

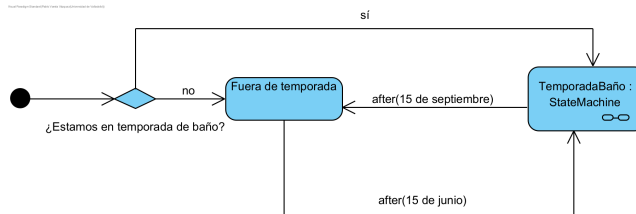


Figura 3.3: Diagrama principal estados zona de baño

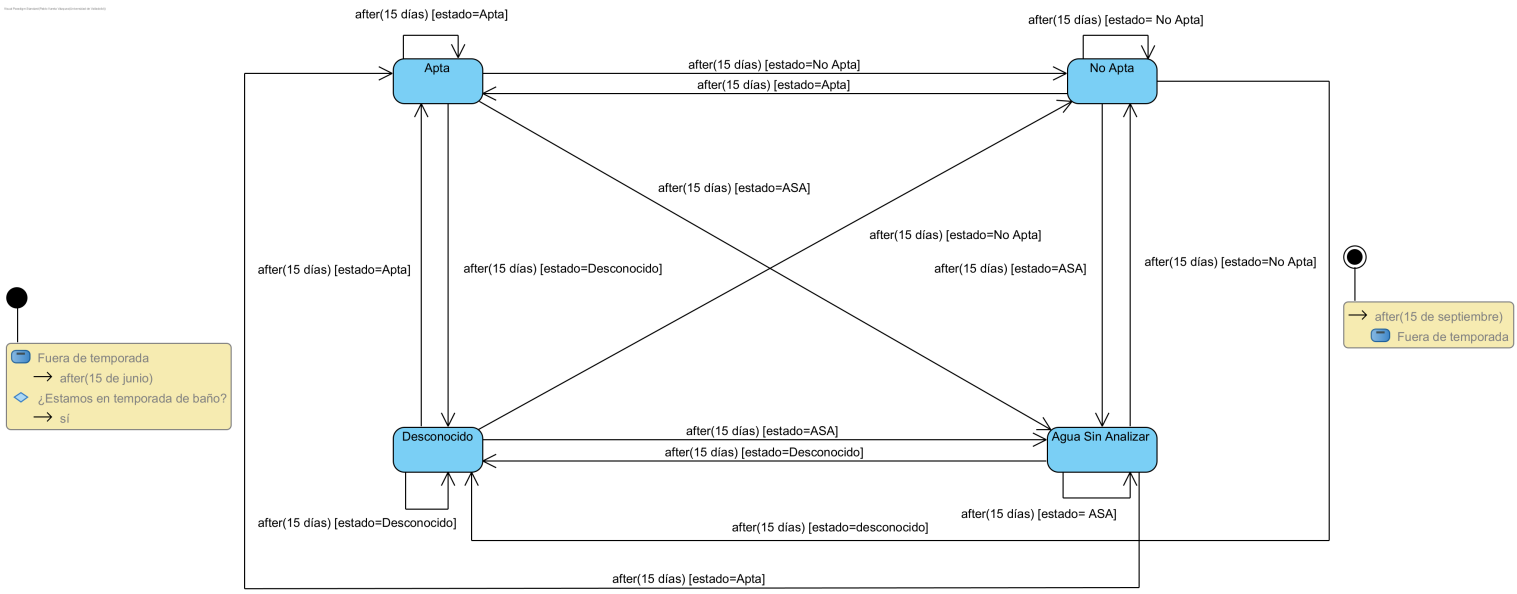


Figura 3.4: Diagrama temporada baño estados zona de baño

Capítulo 4

Tecnologías utilizadas

En este capítulo se explican las tecnologías que se emplean en el desarrollo de este proyecto. Se dividirán en las siguientes categorías: Herramientas de desarrollo, herramientas de gestión de proyectos, herramientas de análisis, herramientas de comunicación y herramientas para la documentación.

4.1. Herramientas de desarrollo

4.1.1. Flutter

Flutter [112] es un framework desarrollado por Google que permite la creación de apps multiplataforma. Algunas de estas plataformas son Android, IOS, Windows, Linux, MacOS y Web. Utiliza el lenguaje de programación Dart, un lenguaje de programación desarrollado por Google. En un principio fue desarrollado como un software interno dentro de Google pero decidieron lanzarlo como proyecto de código libre.

Algunas de las características que presenta Flutter son las siguientes:

- **Compilación en Nativo:** Flutter compila en código nativo tanto para Android como para iOS. Esto significa que las aplicaciones resultantes se ejecutan de manera eficiente y aprovechan las características específicas de cada plataforma.
- **Interfaces Gráficas Flexibles:** Flutter permite crear interfaces gráficas de manera muy flexible. Puedes combinar diferentes Widgets (elementos gráficos) para construir las vistas de tus aplicaciones.
- **Desarrollo Rápido:** Flutter ofrece un ciclo de desarrollo rápido. Se pueden ver los resultados de forma instantánea mientras escribes el código, lo que agiliza el proceso de creación de aplicaciones.

- **Motor Gráfico:** Flutter utiliza el motor gráfico Skia, que renderiza los elementos gráficos en 2D.

La aplicación desarrollada utiliza la versión 3.22.0 de Flutter y la versión 3.4.0 de lenguaje de Programación Dart.

En la aplicación utilizaremos **dependencias** [13], es decir, software externo a nuestro propio software. En Flutter, a las dependencias se las denomina **paquetes** y se encuentran disponibles en Pub.dev, que es el repositorio oficial para paquetes en Flutter. A continuación, se describen los paquetes utilizados en la aplicación:

1. **get**

El paquete `get` [72] es un paquete muy útil ya que se encarga de gestionar los estados de la aplicación, la navegación y las dependencias. Facilita muchas de las tareas que se realizan en Flutter. Es compatible con la versión 3 de Dart y está disponible todas las plataformas: Android, IOS, Windows, Linux, Web y MacOS.

El uso que se le va a proporcionar en esta aplicación es la gestión de estados de la aplicación, la localización y la navegación a través de las diferentes pantallas de la aplicación. La versión que se va a utilizar de este plugin es la 4.6.6

2. **firebase_core**

`firebase_core` [64] es un paquete que permite la inicialización de la aplicación para poder utilizar los servicios de Firebase, como por ejemplo Firebase Authenticator o Firebase Crashlytics. El paquete es compatible con la versión 3 de Dart y se puede usar en Android, IOS, MacOS, Web y Windows.

Nosotros vamos a usar la versión 2.31.0 de este paquete.

3. **firebase_crashlytics**

`firebase_crashlytics` [65] es un paquete se utiliza para registrar y analizar errores y excepciones en la aplicación. Esto ayuda a identificar los problemas y mejorar la calidad de la aplicación. El paquete es compatible con la versión 3 de Dart y se puede usar tanto en Android, IOS, MacOS, Web y Windows.

Nosotros vamos a usar la versión 3.5.5 de este paquete.

4. **in_app_update**

`in_app_update` [75] es un paquete que se utiliza para proporcionar un método de actualización integrado en la aplicación que se activa cuando hay una versión disponible de la aplicación. Este sistema está disponible para Android y utiliza la API que Google ofrece [34]. Esta API presenta dos métodos de actualización distintos:

- **Flujo de actualización flexible:** El flujo de actualización flexible permite que el usuario siga utilizando la aplicación mientras se descarga la actualización de la aplicación. Cuando este lista la actualización a elección del usuario, se continuará con el proceso de instalación reiniciando la aplicación. El flujo de este proceso se puede observar en la Figura 4.1.

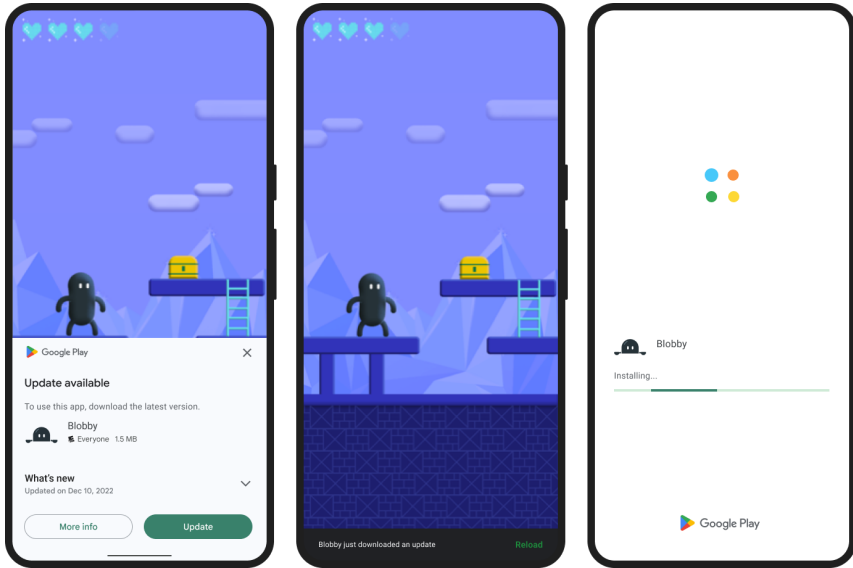


Figura 4.1: Flujo de actualización flexible. Tomada [34]

- **Flujo de actualización inmediato:** El flujo de actualización inmediato muestra el pop-up de actualización a pantalla completa. Esto requiere que el usuario descargue, instale y reinicie la aplicación para seguir usándola. El flujo de este proceso se puede observar en la Figura 4.2.

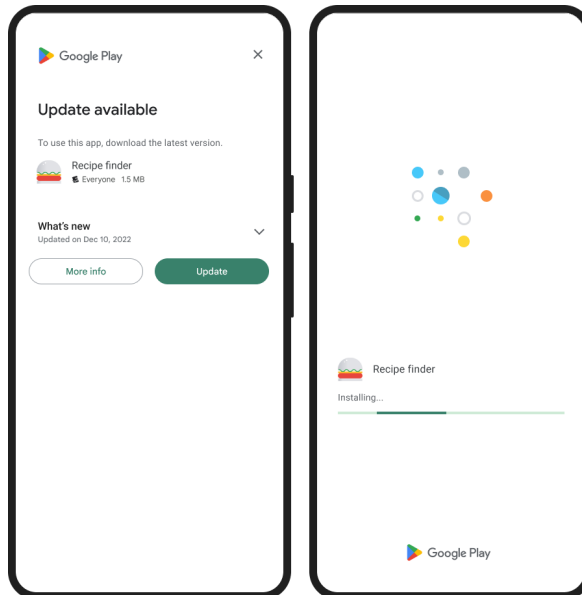


Figura 4.2: Flujo de actualización inmediato. Tomada [34]

En la aplicación, se utilizará la versión de actualización inmediata y se está usando la versión 4.2.2 de este paquete.

5. **dio**

dio [59] es un paquete que permite realizar peticiones http. Algunos de las características que tiene añadido Dio son la configuración global, el uso de timeouts, el uso de interceptores, la descarga o subida de archivos. En la aplicación, usaremos la versión 5.4.3+1 de este paquete.

6. **webview_flutter**

webview_flutter [84] es un paquete que proporciona un webview, es decir, permite cargar una página web u otro contenido externo integrado dentro de nuestra aplicación. La versión del paquete en nuestra aplicación será la 4.7.0.

7. **external_app_launcher**

external_app_launcher [60] es un paquete que permite lanzar otra aplicación especificada si está instalada y si no lo está redirige a Google Play o App Store. La versión del paquete es la 4.0.0.

8. **url_launcher**

url_launcher [83] es un paquete que permite lanzar una url en un navegador externo. También admite el lanzamiento de llamadas (a la aplicación de teléfono), SMS (a la aplicación de mensajes) o correos electrónicos (a la aplicación de correo predeterminada). La versión del paquete es la 6.2.6.

9. **firebase_storage**

firebase_storage [68] es un paquete que permite la conexión con el servicio de Firebase Storage, incluido dentro de Firebase. La versión que se utilizará será la 11.7.5

10. **cached_network_image**

cached_network_image [55] es un paquete que muestra la imagen descargada de internet y la guarda en el directorio de caché para evitar descargarla de nuevo al volverla a usar. La versión de este paquete utilizada es la 3.3.1

11. **package_info_plus**

package_info_plus [77] es un paquete que nos proporciona información acerca del paquete de la aplicación, como su número de versión o su nombre. La versión que vamos a utilizar de este paquete es la 8.0.0.

12. **flutter_email_sender**

flutter_email_sender [69] es un paquete que nos permite enviar un correo electrónico desde Flutter utilizando los mecanismos nativos del sistema operativo. La versión que vamos a utilizar es la 6.0.3.

13. **shared_preferences**

shared_preferences [81] es un paquete que nos permite almacenar datos de manera persistente en memoria. En Android utiliza el sistema SharedPreferences y en IOS utiliza el sistema NSUserDefaults. La versión del paquete que estamos utilizando es la 2.2.3.

14. **firebase_auth**

firebase_auth [63] es un paquete que nos permite hacer la conexión con el servicio Firebase Authentication, incluido dentro de Firebase. La versión que vamos a utilizar es la 4.19.5.

15. **google_sign_in**

google_sign_in [73] es un paquete que permite que podamos iniciar sesión con nuestra cuenta de Google dentro de nuestra aplicación. La versión que vamos a utilizar es la 6.2.1.

16. **permission_handler**

permission_handler [80] es un paquete que permite que podamos comprobar el estado de los permisos y solicitarlos en caso de que los necesitemos. La versión que vamos a utilizar es la 11.3.1.

17. **image_picker**

image_picker [74] es un paquete que permite que podamos utilizar fotografías de nuestra galería o hacer fotos utilizando la cámara. La versión que vamos a utilizar es la 1.1.1.

18. **firebase_database**

firebase_database [66] es un paquete que nos permite hacer la conexión con el servicio Firebase Realtime Database, incluido dentro de Firebase. La versión que vamos a utilizar es la 10.5.5.

19. **file_picker**

file_picker [61] es un paquete que nos permite seleccionar archivos de nuestro dispositivo para poder utilizarlos en nuestra aplicación. La versión que vamos a utilizar es la 8.0.3.

20. **shimmer**

shimmer [82] es un paquete que nos permite añadir a nuestra aplicación el efecto shimmer de manera sencilla. La versión que vamos a utilizar es la 3.0.0.

21. **device_info_plus**

device_info_plus [58] es un paquete que nos permite consultar toda la información acerca de un dispositivo, como el modelo, el sistema operativo, la versión del sistema operativo... La versión que vamos a utilizar es la 10.1.0.

22. **camera**

camera [56] es un paquete que nos permite acceder a las cámaras de los dispositivos y poder realizar ciertas acciones con ellas, como hacer fotos o grabar vídeos. La versión que vamos a utilizar es la 0.11.0.

23. **flutter_local_notifications**

flutter_local_notifications [70] es un paquete que nos permite enviar notificaciones locales en el dispositivo. La versión que vamos a utilizar es la versión 17.1.2.

24. **firebase_messaging**

`firebase_messaging` [67] es un paquete que nos permite hacer la conexión con el servicio Firebase Messaging, incluido dentro de Firebase. La versión que vamos a utilizar es la 14.9.2.

25. **path**

`path` [78] es un paquete que nos permite manipular las rutas a objetos. Algunas acciones que podemos realizar son extraer la extensión, extraer el nombre del fichero... La versión que vamos a utilizar es la 1.9.0.

26. **flutter_rating_bar**

`flutter_rating_bar` [79] es un paquete que nos proporciona un widget que consiste en una barra de valoración de estrellas. La versión que vamos a utilizar es la 4.0.1.

27. **intl**

`intl` [76] es un paquete que nos proporciona facilidades de internacionalización y localización, incluida la traducción de mensajes, plurales y géneros, formateo y análisis sintáctico de fechas y números, y texto bidireccional. La versión que vamos a utilizar es la 0.19.0.

28. **flutter_slidable**

`flutter_slidable` [71] es un paquete que proporciona un widget que permite deslizar hacia derecha e izquierda para realizar una determinada acción. La versión del paquete que vamos a utilizar es la 3.1.0.

29. **firebase_analytics**

`firebase_analytics` [62] es un paquete que permite integrar Google Analytics para Firebase en una aplicación Android o IOS. La versión del paquete que vamos a implementar es la 10.10.5

30. **connectivity_plus**

`connectivity_plus` [57] es un paquete que permite obtener el estado de la red en dispositivos Android e IOS. La versión del paquete que vamos a implementar es la 6.0.3.

4.1.2. **Firestore**

Firestore [9] [33] [41] es un backend online creado por Google. La función principal que tiene Firestore es facilitar el desarrollo de aplicaciones móviles. La plataforma está basada en la nube y está disponible para conectar aplicaciones para los sistemas operativos Android, IOS y para web.

Algunas de las características [116] que presenta Firestore son las siguientes:

- **Almacenamiento de Datos en la Nube:** Firestore dispone de una base de datos en tiempo real. Esta permite almacenar y acceder a datos de forma segura.

- **Autenticación de usuarios:** Permite integrar en las aplicaciones el proceso de autenticación de los usuarios.
- **Análisis y Monitorización:** Firebase ofrece un sistema de análisis que permite a los desarrolladores obtener información sobre el uso o los errores de sus aplicaciones.
- **Integración con otras tecnologías:** Firebase dispone de extensiones que permiten conectarse con otras aplicaciones de Google o de terceros.

A continuación en las siguientes secciones se especificará las diferentes características de Firebase que se van a utilizar en la aplicación:

Firestore Storage

Firestore Storage [23] [6] es un servicio de almacenamiento de objetos, como fotos, vídeos o archivos. Algunas de las funciones claves de Firestore Storage son las siguientes:

- **Operaciones robustas:** Las operaciones de carga y de descarga se realizan sin importar la calidad de red. Si estas operaciones se interrumpen, se reiniciarán en el punto en el que se interrumpieron.
- **Seguridad sólida:** Firestore Storage se integra con otro servicio de Firestore que se llama Firestore Authentication. Esto permite que con las operaciones requieran autenticación por parte de los usuarios.
- **Gran escalabilidad:** Permite aumentar el espacio de almacenamiento si el número de objetos almacenados aumenta considerablemente.

El funcionamiento de Firestore Storage es el siguiente: los desarrolladores usan el SDK de Firestore que es el que permite subir y descargar archivos u otros objetos del servicio. Los archivos se almacenan en buckets en Google Cloud Console. Desde allí los usuarios pueden subir y descargar sus archivos.

Firestore Authentication

Firestore Authentication [24] es un servicio de Firestore que realiza el proceso de inicio de sesión para que el usuario pueda tener una experiencia más personalizada o que queden protegidos los datos y acciones en los que se realiza la aplicación.

Algunas de las funciones claves que realiza este servicio son las siguientes:

- **Autenticación basada en correo electrónico y contraseña:** Permite que los usuarios se autenticquen con su correo electrónico y su contraseña. Se proporcionan varios métodos dentro del SDK de Firestore Authentication para realizar las opciones de registro, inicio de sesión y cierre de sesión. Además, Firestore Authentication se encarga de enviar los correos electrónicos para el restablecimiento de la contraseña.

- **Integración con proveedores de identidad:** Algunos de los proveedores de identidad con los que permite autenticarse Firebase son: Google, Apple, Microsoft, Facebook, X (Twitter), GitHub y Yahoo. Hay que habilitar previamente la configuración en estos proveedores para permitir la autenticación con ellos.
- **Autenticación con números de teléfono:** Permite que se envíe SMS a los números de teléfonos para autenticarlos.
- **Autenticación con sistemas de autenticación personalizados:** Permite la conexión con sistemas de terceros como OpenID connect o SAML para realizar la autenticación.
- **Autenticación anónima:** Permite la creación de una cuenta anónima para permitir el uso a la funciones que requieran autenticación. Existe la opción de convertir una cuenta anónima en una cuenta normal posteriormente.

Firestore Crashlytics

Firestore Crashlytics [26] es un producto de Firestore, que permite visualizar los fallos se produzcan en la aplicación en tiempo de ejecución.

Algunas de las funciones claves de este producto son:

- **Informes de fallas con datos seleccionados:** Agrupa los problemas detectados en una lista de problemas manejable, proporcionando información.
- **Soluciones para una falla común:** Ofrece Crash Insights, que proporciona sugerencias para los posibles errores comunes que pueda ocurrir.
- **Integración con Analytics:** Crashlytics registra los eventos como `app_exception` en Analytics.
- **Alertas en tiempo real:** Permite recibir alertas cuando haya nuevos problemas que puedan requerir una atención urgente.

Firestore Realtime Database

Firestore Realtime Database [28] es una base de datos NoSQL alojada en la nube, que se incluye en Firestore. Los datos en esta base de datos se almacenan en formato JSON y se sincronizan en tiempo real con cada uno de los clientes que se encuentren conectados en ese momento.

Como elemento de seguridad se utilizan una serie de reglas, que se pueden configurar desde la consola de Firestore, que nos indican como se pueden estructurar los datos y quien puede leer y escribir dichos datos en la base de datos.

Algunas de las funciones que dispone Firestore Realtime Database son las siguientes:

- **Tiempo real:** Realtime Database no utiliza las peticiones HTTP, sino que utiliza la sincronización de datos, por lo que al cambiar los datos, los dispositivos que estén conectados reciben esa actualización casi al momento.
- **Sin conexión:** Las apps que usan Realtime Database permiten su utilización sin conexión debido a que los datos se guardan en el disco y cuando vuelve la conexión, el dispositivo recibe los cambios que faltan y sincroniza los cambios que se hayan hecho en local con el estado en el que se encuentre el servidor.
- **Acceso desde dispositivos cliente:** Se puede acceder a Realtime Database desde un dispositivo móvil o desde un navegador web. Podemos configurar reglas de seguridad desde la consola de Firebase para proteger el acceso a los datos que se encuentren almacenados en la base de datos.

Firestore Hosting

Firestore Hosting [27] es un servicio de Firestore que permite alojar contenido web estático o dinámico y microservicios.

Algunas de las funciones clave de este servicio son las siguientes:

- **Entrega de contenido mediante una conexión segura:** Firestore Hosting incluye la conexión mediante SSL por lo que no es necesario hacer ninguna configuración adicional de seguridad.
- **Alojar contenido dinámico y estático, además de microservicios:** Permite alojar todo tipo de contenido para hosting desde contenido dinámico y estático, hasta APIs o microservicios.
- **Obtener contenido con rapidez:** Los archivos proporcionados se almacenan en discos SSD y Firestore se encarga de seleccionar la mejor forma de almacenar dicho contenido y que el usuario pueda obtenerlo.
- **Emular y compartir tus cambios antes de publicarlos:** Podemos observar y probar los cambios en una URL alojada de forma local y después interactuar con un backend emulado.
- **Implementar versiones nuevas con un comando:** Con Firestore CLI se puede poner una aplicación web en funcionamiento con solo ejecutar un comando.

Firestore Cloud Messaging

Firestore Cloud Messaging [25] es un servicio de Firestore que permite enviar mensajes de forma segura y sin costo. Algunos ejemplos para los que se puede usar son: para notificar a un usuario de un correo electrónico o para captar la atención del usuario en un momento.

Un mensaje puede transferir hasta 4000 bytes de información a una aplicación.

Algunas de las funciones clave de este servicio son:

- **Enviar mensajes de notificación o mensajes de datos:** Firebase Cloud Messaging permite enviar mensajes de notificación al usuario y determinar lo que ocurre en el código de la aplicación.
- **Segmentación de los mensajes:** Permite distribuir mensajes de manera personalizada.

4.1.3. Postman

Postman [50] es un programa que permite hacer llamadas a las Application Program Interface (API). Esto es muy útil debido a que permite hacer pruebas con las API o comprobar su correcto funcionamiento.

El origen de este programa fue una extensión disponible para Google Chrome, que más tarde pasó a ser una aplicación disponible para los sistemas operativos Windows, Linux y MacOS. Actualmente, la aplicación es de uso gratuito.

4.1.4. Justinmind

Justinmind [119] es una herramienta de prototipado utilizada para crear prototipos de alta fidelidad para páginas web y aplicaciones móviles. Es conocido por su habilidad de ofrecer prototipos muy próximos al producto final, al igual que ofrecer herramientas de colaboración, interacción y diseño. Los prototipos generados con esta aplicación pueden ser compartidos a través de la nube y ser simulados en dispositivos móviles.

Algunas de las características que dispone Justinmind son las siguientes:

- **Diseño de la interfaz de usuario:** Justinmind permite a los diseñadores crear interfaces de usuario. Por defecto, vienen algunos kits de diseño por defecto con la posibilidad de importar algunos nuevos.
- **Diseño de interacciones web:** Existe una gran variedad de triggers y acciones que permiten añadir a nuestro prototipo desde una pequeña interacción hasta animaciones más complejas.
- **Diseño de los gestos en dispositivos móviles:** Los prototipos pueden tener interacciones usando gestos reales que luego pueden ser probados a través de la app de Justinmind en dispositivos móviles.
- **Diseño de formularios:** Los formularios son totalmente interactivos y pueden usar datos reales para una simulación realista del producto.
- **Visualización de los datos en grids y en tablas:** la herramienta permite importar datos al prototipo y que se puedan visualizar en tablas o en grids.

- **Compartir los prototipos y obtener feedback:** los prototipos se pueden compartir públicamente usando una URL y obtener feedback de otros diseñadores y desarrolladores.
- **Pruebas por parte de los usuarios:** Justinmind está integrado con las principales plataformas de pruebas para usuarios.
- **La aplicación para dispositivos móviles:** Cuando los prototipos se comparten en la nube, es posible acceder a ellos a través de la aplicación móvil. Esto hace posible que los usuarios puedan probar el prototipo en un dispositivo real.

4.1.5. Android Studio

Android Studio [32] [117] es un entorno de desarrollo integrado (IDE) para la plataforma de Android. Fue anunciado durante la conferencia de desarrolladores de Google, Google I/O en el año 2013 y reemplazó a Eclipse como IDE oficial.

Esta basado en IntelliJ de JetBrains. Se encuentra disponible para Windows, Linux, MacOS y Chrome OS.

Algunas de las características de Android Studio son las siguientes:

- Soporte para la construcción específica en Glade.
- Refactorización específica de Android
- Herramientas Lint para detectar problemas de rendimiento, usabilidad y otro tipo de problemas.
- Función de firma de las aplicaciones.
- Editor de diseño.
- Plantillas para crear diseños comunes para Android.
- Soporte con Android Wear
- Integración con Google Cloud Platform
- Soporte para la utilización de emuladores y dispositivos físicos.

En la Figura 4.3 se muestra la interfaz de usuario de Android Studio.

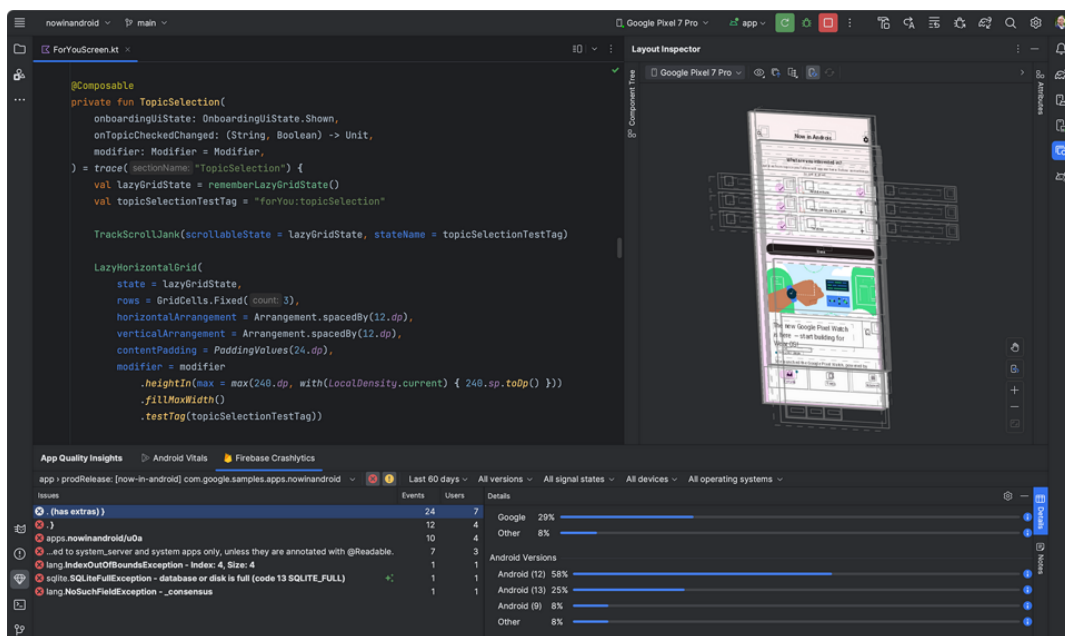


Figura 4.3: Interfaz Android Studio. Tomada [32]

4.1.6. Visual Paradigm

Visual Paradigm [21] es una herramienta de ingeniería de software asistida por computación que proporciona ayuda en el desarrollo de software en las etapas de planificación, análisis, diseño, entre otras. Es decir, soporta el ciclo de vida completo en el desarrollo de software a través de la representación de diagramas.

Algunas de las características de Visual Paradigm son:

- Se encuentra disponible en múltiples plataformas como en Windows y en Linux.
- El diseño está centrado en los casos de uso y nos permite obtener una mayor calidad del software.
- Uso del lenguaje estándar que facilita la comunicación entre el equipo de desarrollo.
- Permite aplicar tanto ingeniería directa como ingeniería inversa. La ingeniería directa es el proceso necesario para crear una aplicación, que comprende, entre otros, la planificación, el análisis, el diseño o la implementación. La ingeniería inversa es el proceso que permite descomponer un sistema o producto para comprender el funcionamiento de este.
- Sincronización entre modelo y código durante todo el ciclo de desarrollo.
- Dispone de dos licencias: la gratuita y la comercial.

- Soporte para Unified Modeling Language (UML).

4.1.7. Bootstrap

Bootstrap [43] es una herramienta orientada al desarrollo de páginas y aplicaciones web. Se basa en los lenguajes de programación HTML, CSS y Javascript. Incluye una amplia gama de elementos de diseño como botones, barras de navegación, enlaces.

Permite tener una página web responsive, es decir, que se adapta al dispositivo que se esté utilizando, ya sea móvil, tablet o un ordenador.

4.1.8. Visual Studio Code

Visual Studio Code [121] es un editor desarrollado por Microsoft, que permite la edición de código con soporte a múltiples lenguajes de programación, integración con Git, soporte para extensiones, entre otros.

Fue lanzado en el año 2015 durante la conferencia de desarrolladores de Microsoft, Microsoft Build 2015. Su código se encuentra publicado en GitHub.

4.2. Herramientas de gestión del proyecto

4.2.1. GitHub

GitHub [122] es una plataforma propiedad de Microsoft que aloja el código que los desarrolladores suben. Este código puede ser tanto de las aplicaciones como de las herramientas que desarrollen. Al usuario le permite no solo descargar la aplicación, sino también entrar al perfil para saber sobre ella o colaborar en su desarrollo.

GitHub utiliza el sistema de control de versiones Git, creado por Linus Torvalds. Más adelante se tratará sobre el uso de Git y las principales características de este sistema de control de versiones.

GitHub ofrece las siguientes características:

- Permite alojar proyectos creando repositorios de forma gratuita.
- Permite observar los proyectos públicos subidos por otro desarrollador.
- Dispone de una inteligencia artificial, llamada GitHub Copilot, que es un asistente de ayuda a la codificación.
- Permite la creación de una Wiki dentro del proyecto

- Dispone de una herramienta de seguimiento de problemas, de forma que otros usuarios puedan realizar sugerencias sobre el proyecto.
- Dispone de una herramienta de revisión de código
- Dispone de herramientas para ver cuanto trabaja un desarrollador en un proyecto

En la Figura 4.4 podemos observar como es la interfaz y como se encuentra alojado nuestro proyecto en GitHub.

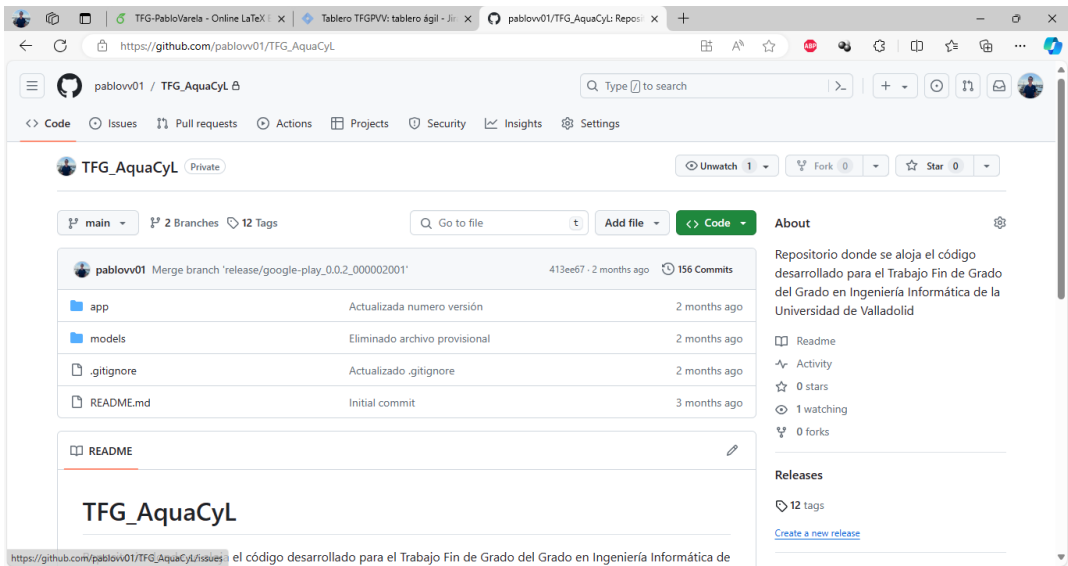


Figura 4.4: Interfaz de GitHub con el proyecto

Git

Git [118] es un software de control de versiones creado por Linus Torvalds. El objetivo de este sistema es llevar el registro de todos los cambios de un archivo y permitir además el trabajo colaborativo con otros usuarios.

Algunas de las características más destacadas de Git es:

- **Desarrollo no lineal:** Permite la gestión de las diferentes versiones en ramas y mezclar las diferentes versiones de un archivo. Git incluye diferentes herramientas para permitirnos navegar y visualizar el historial de desarrollo no lineal.
- **Gestión distribuida:** Git entrega a cada programador una copia local de todo el historial de desarrollo entero y los cambios que vayamos haciendo en nuestra copia local se propagan entre los diferentes repositorios locales.

- Presenta una gestión eficiente de los proyecto de mayor tamaño .

Algunas de las ordenes que utilizaremos de git son las siguientes:

1. **git fetch**: Descarga los cambios realizados en el repositorio remoto.
2. **git merge**: Fusiona la rama en la que nos encontremos en ese momento con la rama indicada en el comando.
3. **git pull**: Obtiene y fusiona los cambios que existan en el repositorio remoto.
4. **git commit**: Confirma los cambios realizados. Adicionalmente, se puede indicar un mensaje con una breve descripción.
5. **git push**: Sincroniza los cambios locales con los del servidor remoto.
6. **git checkout -b**: Crea una nueva rama a partir de la que nos encontramos con el nombre especificado como parámetro.

4.2.2. Jira

Jira [37] [3] es una aplicación web desarrollada por Atlassian, que sirve para gestionar proyectos, tareas y errores. En el desarrollo de software se ha convertido en una herramienta muy importante en el mercado y utilizada en muchas empresas. Facilita la realización de tareas tanto en equipos grandes como en equipos pequeños.

El funcionamiento de Jira es sobre una aplicación web y ella podemos tener proyectos, en los cuales podemos ir creando tickets para diferentes tareas. En nuestro caso, utilizaremos una plantilla disponible dedicada a proyectos SCRUM en el que dentro de cada sprint podemos asignar diferentes tickets, que se corresponden con tareas, historias de usuario o errores.

El 30 de abril de 2024 [4] se anunció que Jira Software y Jira Work Management se fusionarían para crear Jira. Esta nueva versión incluiría una fusión de las características presenten en ambos productos.

En la Figura 4.5 se puede observar la interfaz de Jira utilizando la plantilla de SCRUM correspondiente al tablero de Sprint 4.

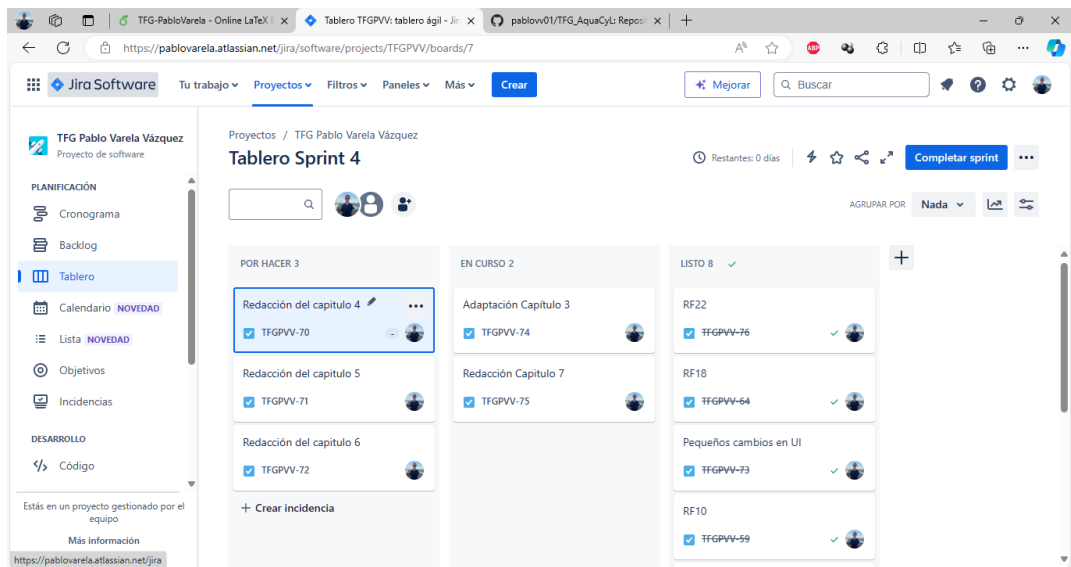


Figura 4.5: Interfaz de Jira

Clockwork Free

Clockwork Free [35] es una aplicación de Jira desarrollada por HeroCoders que se utiliza para medir los tiempos dedicados a cada tarea. También podremos observar el registro de tiempos en una tabla.

Al instalar la aplicación en Jira nos aparecerá un icono de un reloj en las incidencias. Este icono nos desplegará un apartado dentro de las incidencias que nos permite añadir el tiempo de ejecución de una tarea. Tenemos dos opciones para insertar el tiempo: de forma manual o de forma automática. La forma manual es que nosotros registramos el tiempo manualmente mientras que en la forma automática es la propia aplicación la que va contando el tiempo.

En la Figura 4.6 podemos ver la interfaz de la aplicación en una tarea.

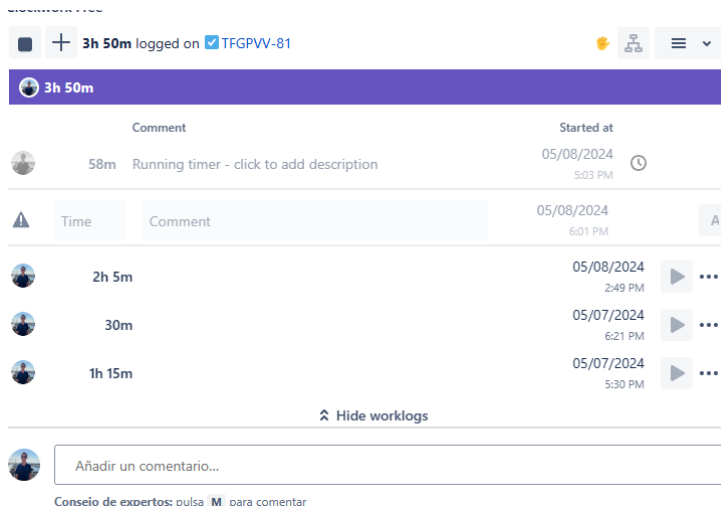


Figura 4.6: Interfaz de la aplicación en una tarea

4.2.3. Sourcetree

Sourcetree [5] [42] es un cliente gratuito desarrollado por Atlassian que nos permite trabajar con nuestros repositorios de código que funcionen bajo la tecnología de Git o Mercurial.

Este cliente nos permite crear ramas de Git, mergear ramas, hacer ramas release, hacer ramas de tipo hotfix, entre otras opciones. Además, se utiliza la opción Gitflow, que es una opción que tiene Sourcetree que nos permite facilitar todo el proceso de creación de ramas.

4.3. Herramientas de análisis

4.3.1. Google Analytics

Google Analytics [11] es una herramienta de análisis creada por Google, que nos permite analizar información sobre sitios webs o aplicaciones, obtener información sobre el rendimiento, conocer cómo los usuarios utilizan una aplicación o sitio web, etcétera.

Nosotros utilizaremos la versión gratuita de este servicio, que además se encuentra integrada con Firebase.

4.4. Herramientas de comunicación

4.4.1. Microsoft Teams

Microsoft Teams [48] es una herramienta de comunicación creada por Microsoft que nos permite comunicarnos a través de mensajes o a través de llamadas y videollamadas. Permite convocar reuniones y tiene una sincronización muy buena con Outlook, el servicio de correo electrónico de Microsoft.

En el caso particular de este TFG, con la tutora de la Universidad de Valladolid, se ha utilizado este medio en ciertos casos puntuales, debido a que las reuniones de seguimiento han sido presenciales. Sin embargo, con el tutor de Getronics, se ha utilizado este medio para todo el seguimiento de proyecto.

4.5. Herramientas para la documentación

4.5.1. Overleaf

Overleaf [53] es una herramienta online para escribir documentos LaTeX. Permite escribir estos documentos de manera colaborativa. Tiene un editor visual para poder editar el documento de manera más sencilla y tiene algunos ajustes que permiten la creación de tablas y figuras de manera más sencilla.

Capítulo 5

Diseño

En este capítulo se abordará las decisiones en cuanto al diseño de la aplicación y en cuanto a su arquitectura.

5.1. Resumen de las decisiones de diseño

Para abordar el desarrollo de este proyecto, se han tomado las siguientes decisiones de diseño:

1. Se utilizará el Framework *Flutter* y el lenguaje de programación *Dart* para el desarrollo de la aplicación. Esto permite que nuestra aplicación pueda ser compilada para diferentes sistemas operativos realizando ajustes mínimos.
2. Se incorporarán diferentes paquetes Flutter a nuestro proyecto para proporcionar funcionalidad ya desarrollada. Estos se encuentran listados en la Sección 4.1.1
3. Para el almacenamiento de archivos y fotos se utilizará *Firebase Storage* incluido dentro de *Firebase*.
4. La gestión de usuarios de la aplicación se realizará usando *Firebase Authenticator*, usando el proveedor de email y contraseña del propio *Firebase* y el proveedor de cuentas de Google integrado en este servicio.
5. Como base de datos usaremos *Firebase Realtime Database*
6. Se decide registrar en Firebase los fallos y errores de cada versión liberada. Para ello, usaremos *Firebase Crashlytics* de *Firebase*.
7. Se quiere obtener información anonimizada sobre cómo utilizan los usuarios la aplicación. Para ello se integrará en la aplicación el servicio *Google Analytics para Firebase*

8. Para la sección de alojamientos de la aplicación, se va a utilizar la plataforma EscapadaRural

5.2. Arquitectura MVVM

Model-View-ViewModel (MVVM) [2] [52] es un patrón arquitectónico de software que se usa para separar la interfaz de usuario en tres componentes distintos, que tienen un rol distinto cada uno de ellos. A continuación, se detalla cada uno de los roles de los componentes:

1. **Modelo (*Model*)**: En este componente nos encontramos todo lo relacionado con los datos y la lógica de negocio. Se encarga de almacenar, obtener y manipular los datos que la aplicación necesita para funcionar.
2. **Vista (*View*)**: Este componente es el responsable de la interfaz de usuario. Se encarga de mostrar los datos al usuario y obtener los datos introducidos por el usuario.
3. **Vista-Modelo (*ViewModel*)**: Este componente se encarga de hacer de puente entre la Vista y el Modelo, conteniendo la lógica relacionada con la transformación de los datos del Modelo para mostrarlos en la Vista.

El funcionamiento de la arquitectura es que la Vista (*View*) se suscribe a los cambios que haya en la Vista-Modelo (*ViewModel*). La Vista-Modelo actualiza la Vista cuando se producen cambios en el Modelo (*Model*). El funcionamiento se puede ver en la Figura 5.1.

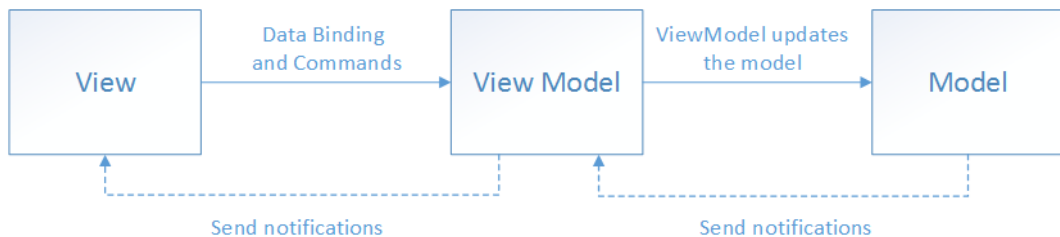


Figura 5.1: Patrón MVVM. Tomada [12]

Cada uno de los componentes es independiente de los demás. Esto permite que el código se pueda testear y mantener de mejor manera.

Con el MVVM se obtiene una serie de ventajas [8] [45] [85] que mejoran la calidad del código y el desarrollo del producto. Estas ventajas son:

- **Código más legible**: Al separarse el código en tres roles (Modelo, Vista y Vista-Modelo), el código es más fácil de entender y de mantener.

- **Código organizado:** La estructura de esta arquitectura provoca que se organice claramente agrupándose cada sección de código según su rol.
- **Código más limpio:** Al separar las responsabilidades de cada uno de los roles, se fomenta la creación de código más conciso y más limpio en la que cada rol se encarga de una tarea en específico.
- **Código reutilizable:** Al separarse los roles permite que se pueda reutilizar componentes por otros elementos del programa.
- Permite realizar **pruebas unitarias** para cada uno de los roles de forma independiente.

5.3. Diseño inicial de la interfaz de usuario

En esta sección se mostrarán los *mock-ups* con los diseños iniciales de la interfaz de usuario de la aplicación móvil realizados con la herramienta Justinmind (ver Sección 4.1.4). Estos diseños se pueden ver de las Figuras 5.2 a 5.16.



Figura 5.2: Pantalla principal



Figura 5.3: Pantalla inicio de sesión

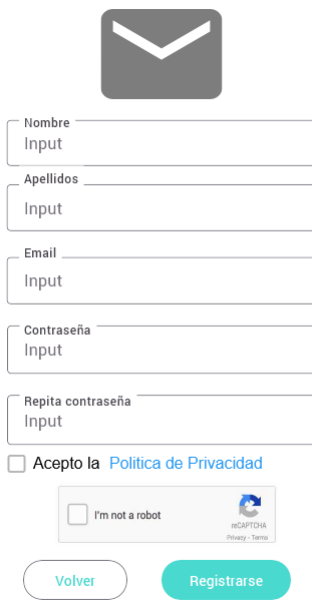


Figura 5.4: Pantalla de registro

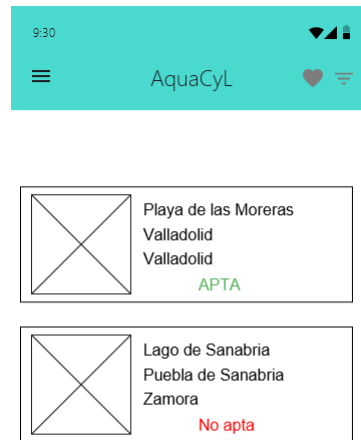


Figura 5.5: Listado de zonas de baño

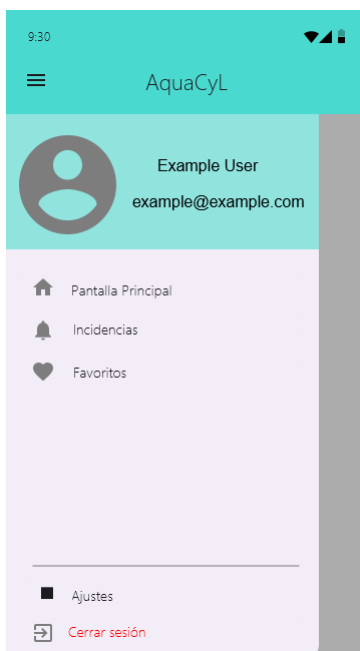


Figura 5.6: Hamburger Menu



Figura 5.7: Filtros

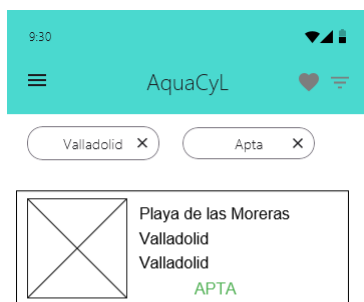


Figura 5.8: Listado de zonas de baño con filtros aplicados

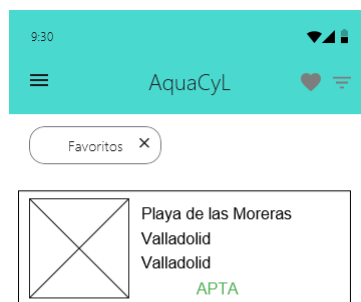


Figura 5.9: Listado de zonas de baño favoritas

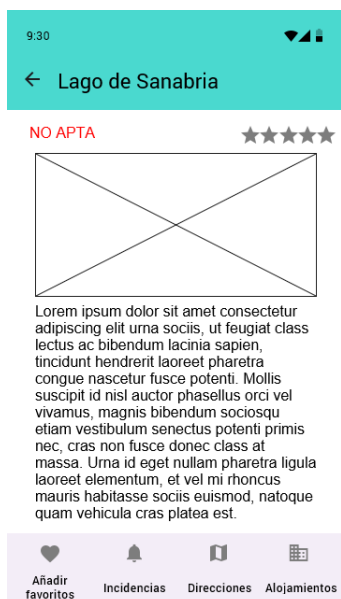


Figura 5.10: Pantalla de detalles

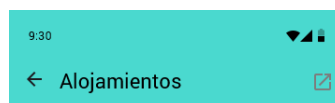


Figura 5.11: Pantalla de alojamientos

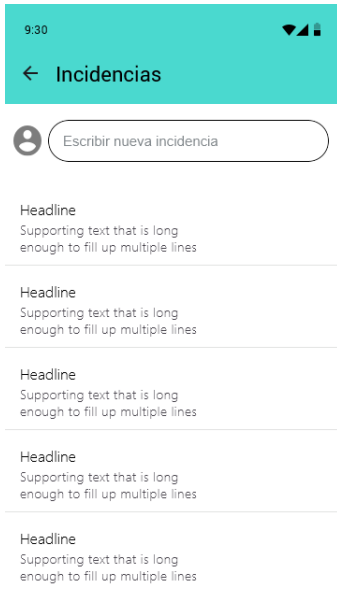


Figura 5.12: Listado de las incidencias de una zona de baño

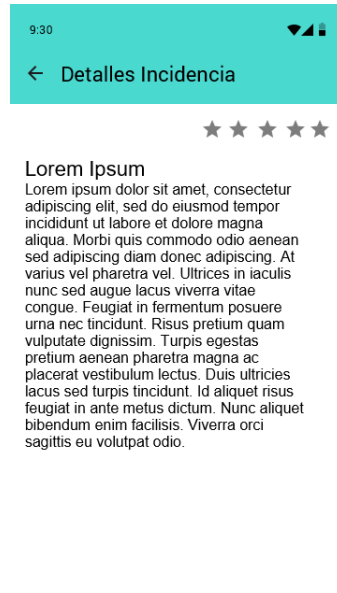


Figura 5.13: Pantalla de detalles de una incidencia

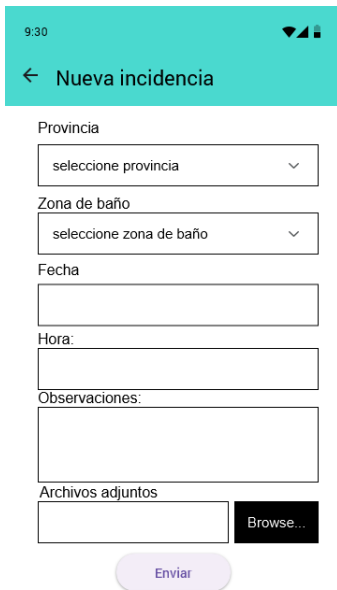


Figura 5.14: Pantalla de nueva incidencia

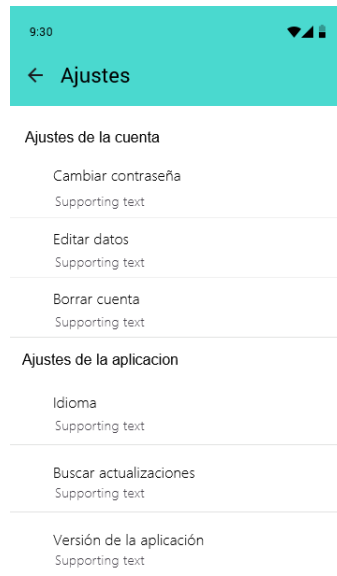


Figura 5.15: Pantalla de ajustes

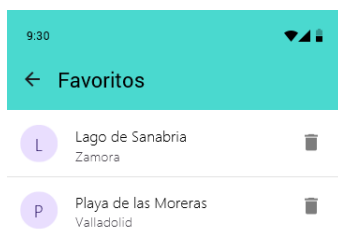


Figura 5.16: Pantalla de favoritos

Durante el desarrollo se han introducido cambios en el diseño de la aplicación. Los cambios en la interfaz de usuario se pueden ver en el manual de usuario de la aplicación (ver Anexo A) o en la propia aplicación (ver enlace de descarga de la aplicación en el Anexo B). En el manual de usuario se incluyen capturas de pantalla de como ha sido el aspecto visual de la aplicación al finalizar el desarrollo.

5.4. Imagen de marca

Para elaborar la imagen de marca de la aplicación se ha utilizado la herramienta *Brand Kit Generator* de Hubspot [36]. Esta herramienta nos permite crear nuestra imagen de marca.

El logotipo elegido para nuestra aplicación es el que se puede observar en la Figura 5.17. Este logotipo dispone de un icono, que representa un río con una persona haciendo uso de él. Se puede ver el nombre de la marca dentro del logotipo y este se encuentra envuelto en un círculo. En la aplicación y la página web se ha utilizado este logotipo o alguna variante del logotipo.



Figura 5.17: Logotipo de la aplicación

En cuanto al color elegido, se ha optado por utilizar el color primario con código hexadecimal #69B4AC, que es el que se puede observar en el logo en la Figura 5.17.

5.5. Diseño arquitectónico

La arquitectura de la aplicación consiste en 5 paquetes principales. Estos se encuentran envueltos por un paquete general que recibe el nombre de paquete de la aplicación: *es.pablo.aquacyl.aqua_cyl*. Estos se detallan a continuación:

- **Entities:** Contiene las entidades reflejadas en el modelo de dominio y con las que se trabaja tanto en el paquete views como en el paquete net.
- **Utils:** Contiene clases que proporcionan una funcionalidad a todo el conjunto de la aplicación.
- **net:** Contiene todos los elementos que realizan operaciones con apis o con la propia red. Esta se subdivide en 3 paquetes:
 - **api:** Contiene las llamadas a las APIs que utiliza nuestra aplicación.
 - **firebase:** Contiene las llamadas a los servicios de Firebase que utilizamos en nuestra aplicación.
 - **interceptors:** Contiene interceptores para la comunicación de red. Un interceptor nos permite capturar los diferentes eventos que suceden durante la comunicación de red.

- repository: Contienen las clases para realizar y procesar las llamadas a las APIs. Funcionan como el modelo de la arquitectura MVVM.
- **views:** Contienen las clases que representan a la vista y a la vista-modelo de la arquitectura MVVM.

La arquitectura general representada en un diagrama *decomposition & Uses Style* se puede ver en la Figura 5.18.

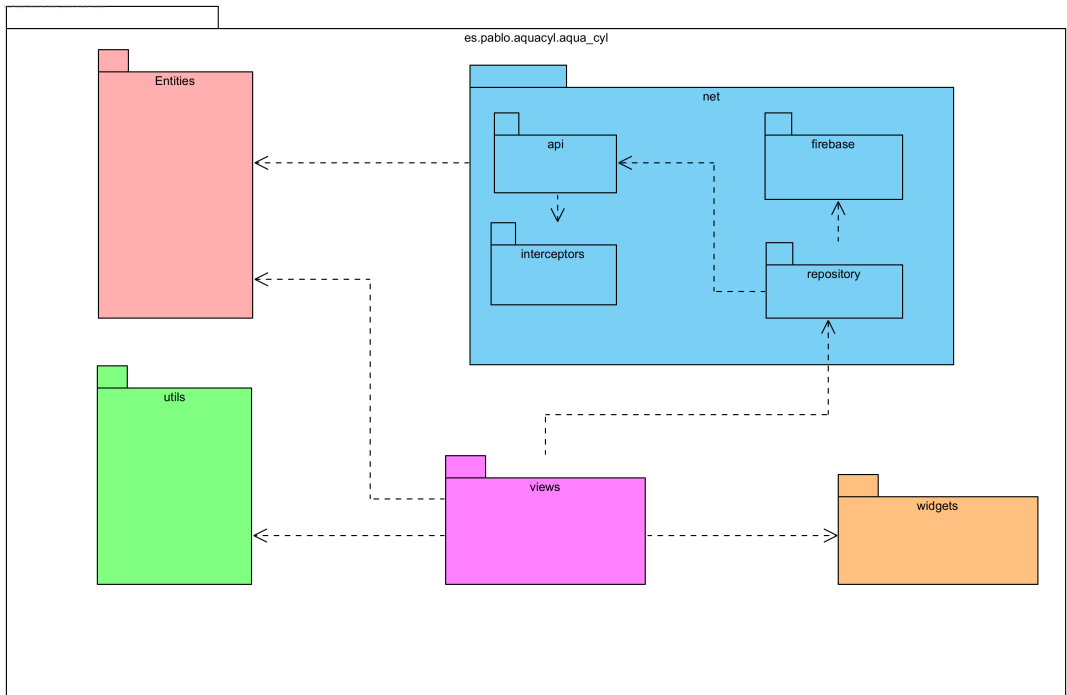


Figura 5.18: Arquitectura general de la aplicación

En la Arquitectura general de la aplicación no se muestran los paquetes internos debido a la gran cantidad de paquetes que dispone la clase views. Si se mostrará en la Figura 5.23, correspondiente al diagrama *decomposition & Uses Styles* del paquete Views.

En la Figuras 5.19 a 5.23 podemos ver los diagramas de clases por cada uno de los paquetes.



Figura 5.19: Decomposition Entities

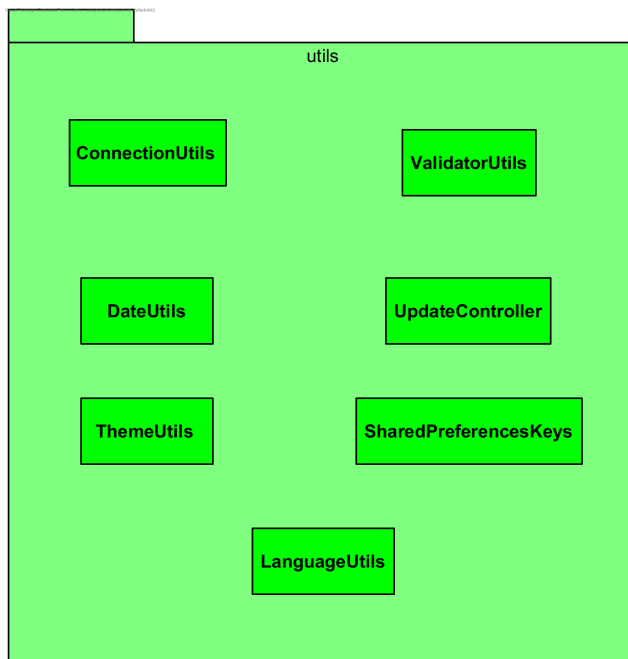


Figura 5.20: Decomposition Utils

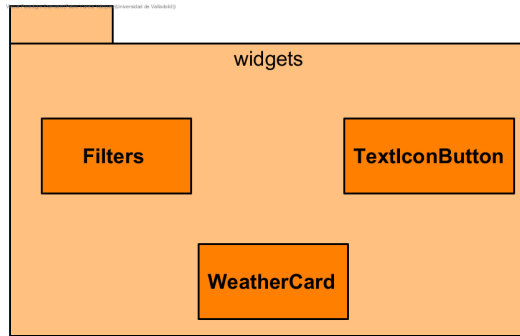


Figura 5.21: Decomposition Widgets

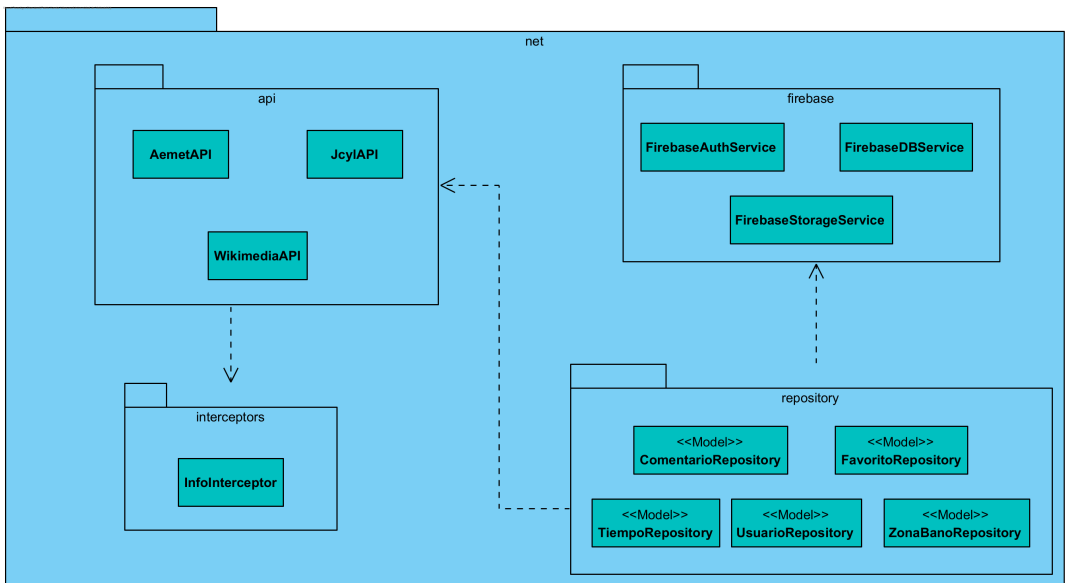


Figura 5.22: Decomposition & Uses Styles Net

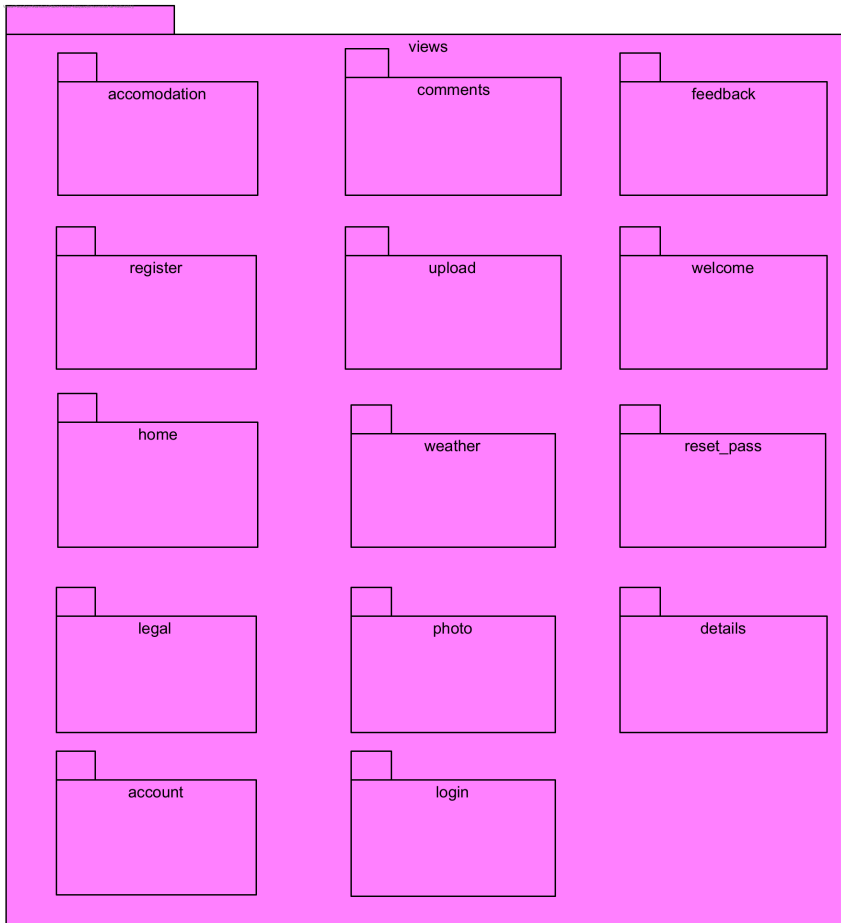


Figura 5.23: Decomposition Views

En las Figuras 5.24 a 5.37 se podrá ver el contenido de cada uno de los paquetes especificados.

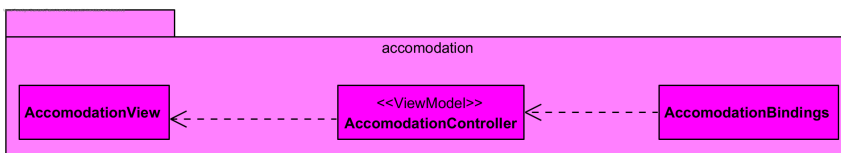


Figura 5.24: Decomposition & Uses Styles Accomodation

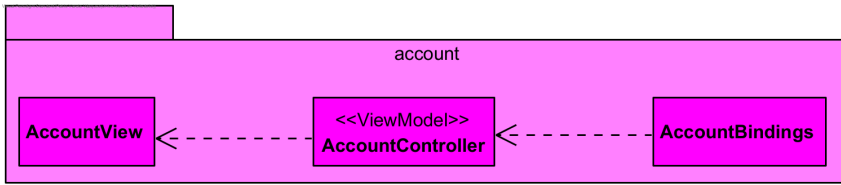


Figura 5.25: Decomposition & Uses Styles Account

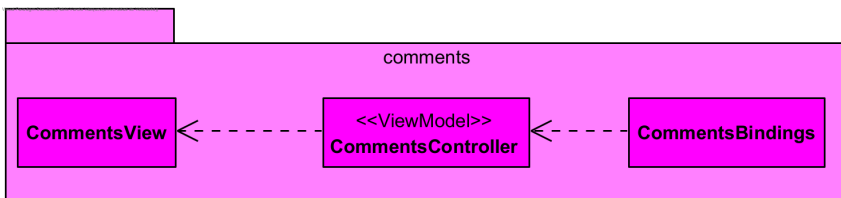


Figura 5.26: Decomposition & Uses Styles Comments

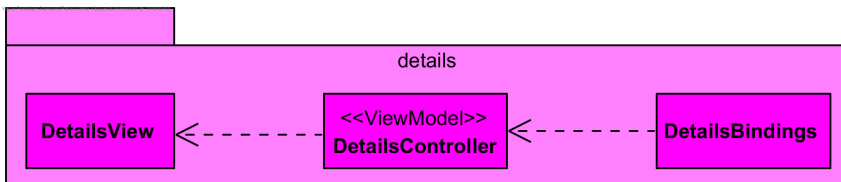


Figura 5.27: Decomposition & Uses Styles Details

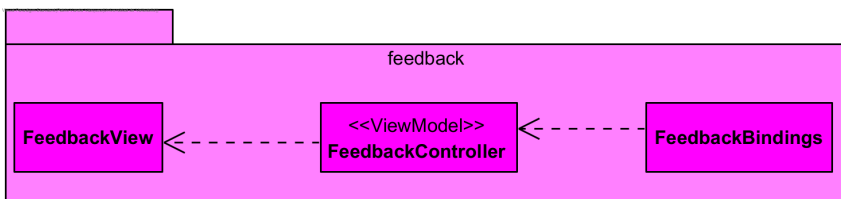


Figura 5.28: Decomposition & Uses Styles Feedback

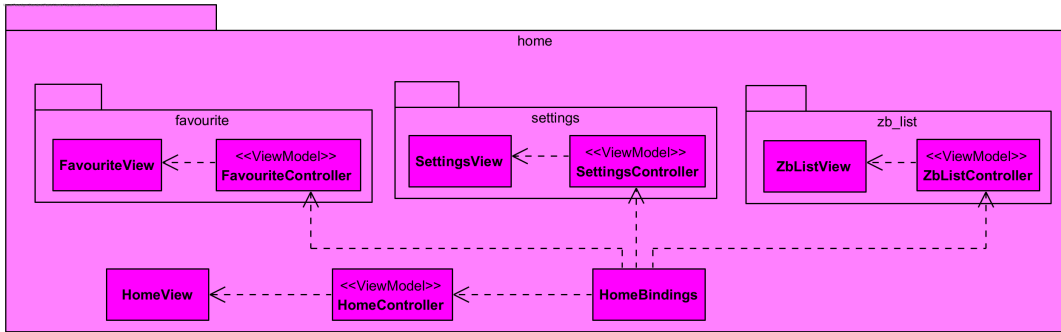


Figura 5.29: Decomposition & Uses Styles Home

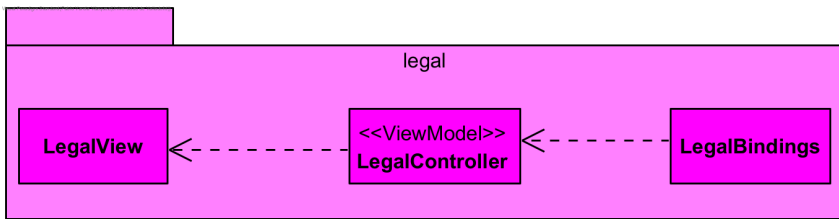


Figura 5.30: Decomposition & Uses Styles Legal

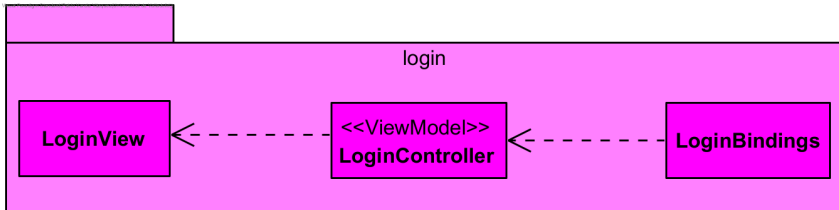


Figura 5.31: Decomposition & Uses Styles Login

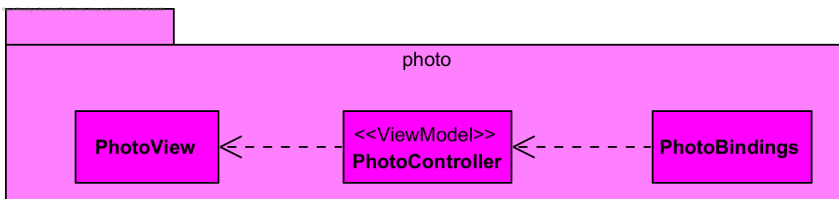


Figura 5.32: Decomposition & Uses Styles Photo

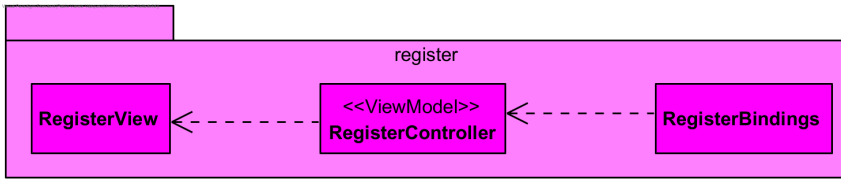


Figura 5.33: Decomposition & Uses Styles Register

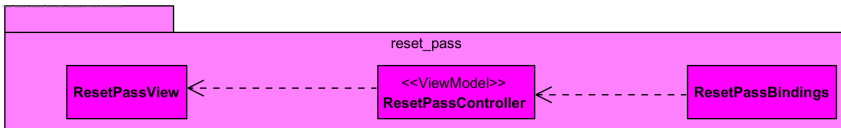


Figura 5.34: Decomposition & Uses Styles Reset Pass

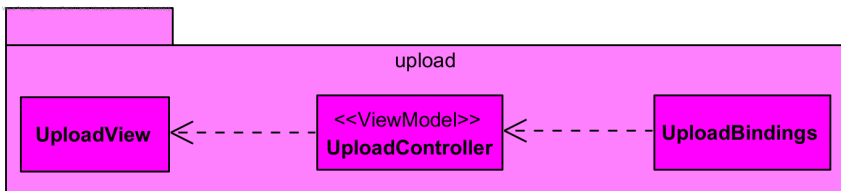


Figura 5.35: Decomposition & Uses Styles Upload

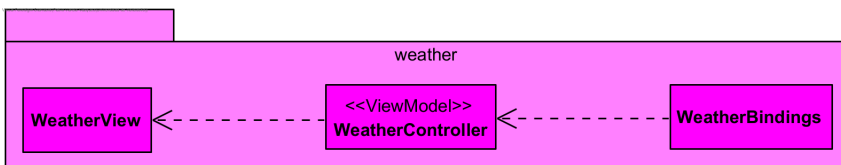


Figura 5.36: Decomposition & Uses Styles Weather

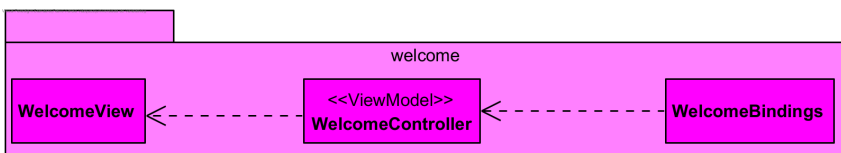


Figura 5.37: Decomposition & Uses Styles Welcome

Los diagramas se podrán ver en mayor resolución en el archivo Visual Paradigm en la carpeta *models* del repositorio disponible en el Anexo B.

5.6. Diseño de la base de datos

En la Figura 5.38 se muestra el diseño de la base de datos. Conviene destacar que se trata de un modelo de base de datos **no relacional**. Al utilizarse Firebase Realtime Database, estos datos se guardan en un formato muy parecido al formato JSON.

Para acceder a estos datos, se ha elaborado unas clases, que se encuentran dentro del paquete `firebase` en `net` (ver Figura 5.22) que realizan la comunicación con dicha base de datos.

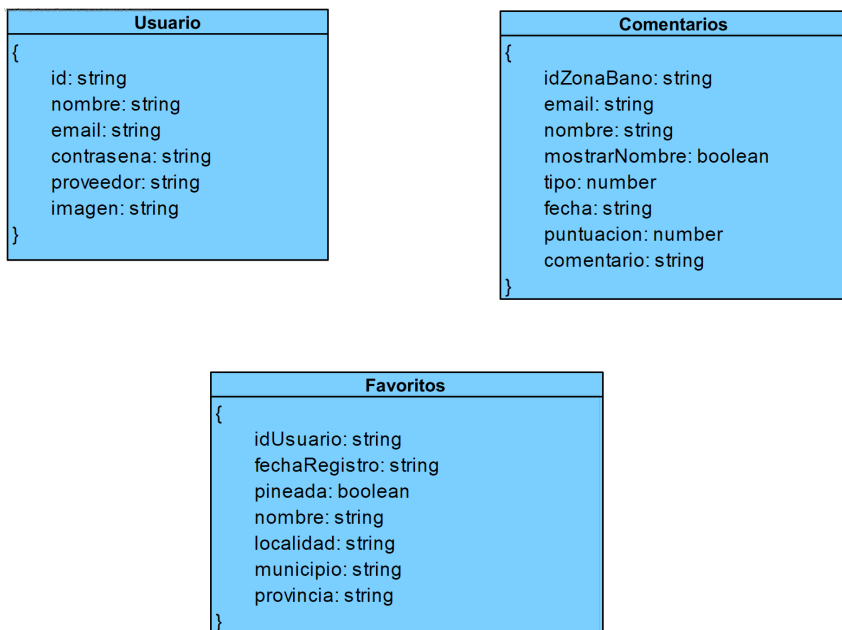


Figura 5.38: Diseño de base de datos

El diagrama se puede consultar en mayor resolución en el archivo de Visual Paradigm en la carpeta `models` del repositorio disponible en el Apéndice B.

5.7. Diseño de comunicación

Siguiendo el patrón MVVM, el controlador de vista (vista-modelo) va a comunicarse con las vistas y con los modelos (repositorios). Las vistas no van a tener comunicación con ningún otro elemento de la aplicación, solo recibirán mensajes por parte del controlador de vista correspondiente. Los modelos (repositorios) se podrán comunicar con las APIs que utiliza nuestra App o con los servicios proporcionados con Firebase. La comunicación con

las API será intervenida por los interceptores para poder mostrar información al equipo de desarrollo acerca de la respuesta obtenida por parte de dicha API.

Para ejemplificar esta comunicación, se ha elaborado un diagrama de secuencia sobre la historia de usuario **RF01: Como usuario quiero poder ver un listado de las zonas de baño disponibles para poder identificar las distintas opciones**. Este diagrama se puede ver en las Figuras 5.39 y 5.40. La versión completa de este diagrama se encuentra en el repositorio disponible en el Apéndice B.

En primer lugar, al iniciar sesión o continuar como invitado, se accede a la ruta Home. Esta ruta es la responsable, a través de Home Binding de cargar los componentes de la pantalla favoritos, los de ajustes y los del listado de las zonas de baño. En esta historia de usuario, nos centraremos en los del listado de zonas de baño.

HomeBindings inicializa el controlador *HomeController* y *ZbListController*. *HomeController* pasa a inicializar la pantalla. Por defecto, la aplicación va a cargar la pantalla del listado de zonas de baño.

ZbListController obtiene los datos llamando a *ZonasBañoRepository*. *ZonasBañoRepository* llamará a *JcylAPI* para obtener las zonas de baño que haya registradas para el año actual. *InfoInterceptor* va a interceptar las llamadas de *JcylAPI* para poder mostrar al desarrollador vía consola la petición y la respuesta de la llamada.

En el caso de que la respuesta tenga un código 200 por parte de la API, se procede a comprobar si el listado obtenido tiene 0 zonas de baño disponibles. Si tiene 0 zonas de baño disponibles se llama de nuevo a la API para obtener las zonas de baño del año anterior. Se procesará la respuesta obtenida y devolverá el listado de zonas de baño procesadas al controlador *ZbListController*. El controlador se encargará de mostrar la lista de zonas de baño por pantalla.

En caso de que la respuesta tenga un código distinto de 200, se producirá una excepción en la API que será devuelta a *ZonasBañoRepository* y esta excepción será relanzada para que la controle el controlador. El controlador *ZbListController* mandará mostrar por pantalla que no hay zonas de baño disponibles y desplegará en un snackbar el mensaje informando del error.

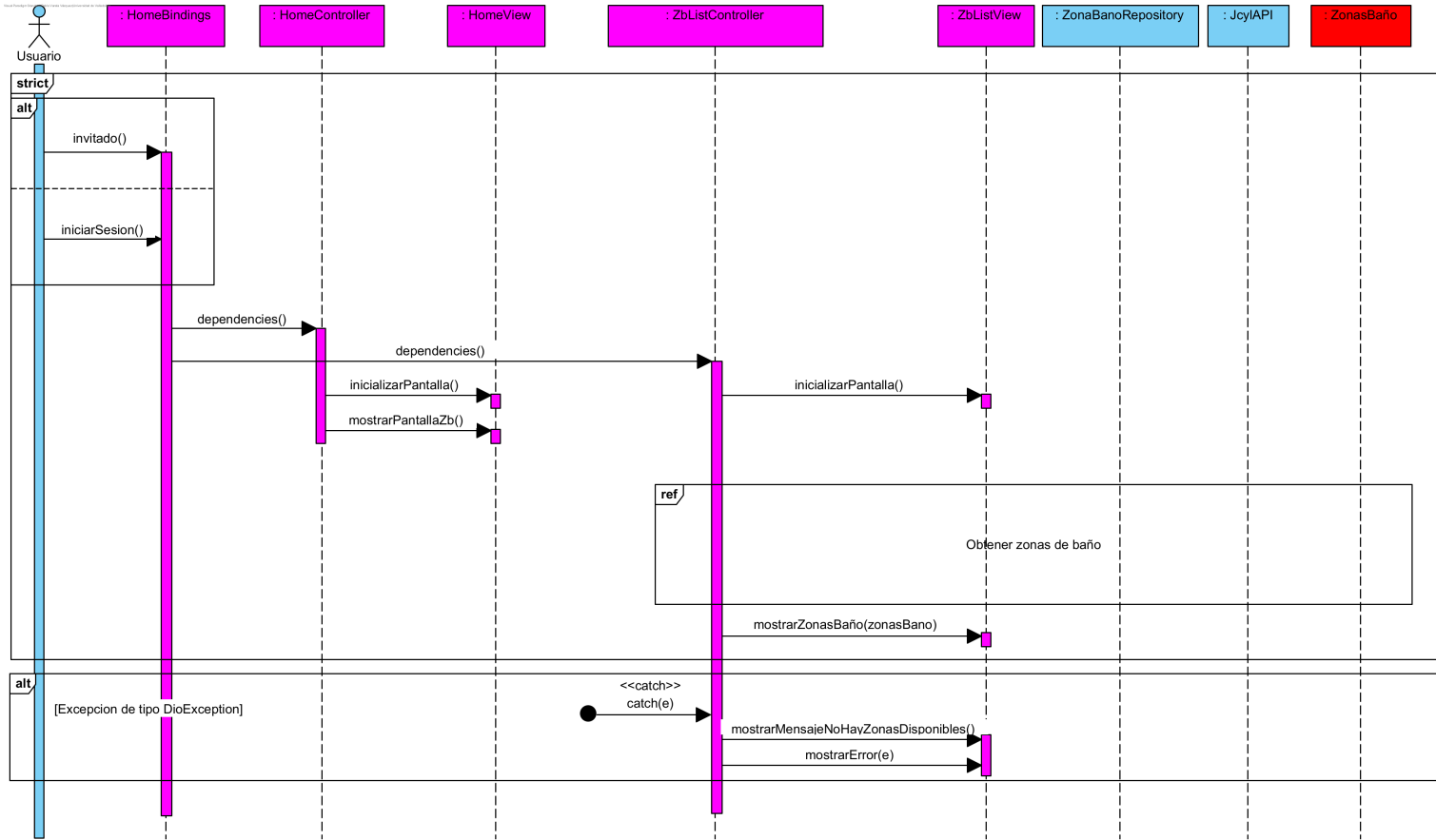


Figura 5.39: Diagrama de secuencia principal para RF01

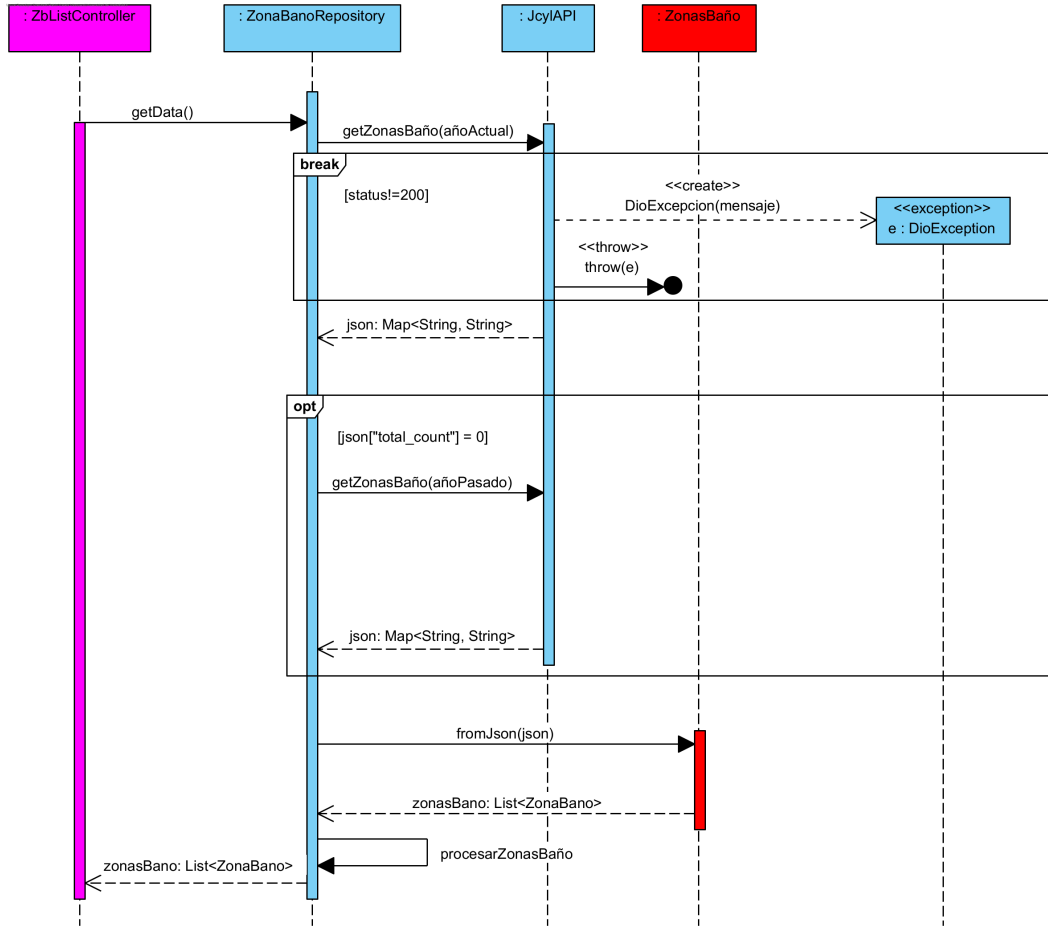


Figura 5.40: Diagrama de secuencia obtener zonas de baño

5.8. Diagrama de despliegue

El despliegue de nuestra aplicación está formado por nuestra aplicación, 3 APIs externas y Firebase. Las APIs externas son la API de la Junta de Castilla y León (JCyL), la API de Wikipedia y la API de la Agencia Estatal de Meteorología (AEMET).

Dentro de Firebase, utilizaremos tres servicios y dos entornos de ejecución. Los servicios son Firebase Crashlytics, Firebase Hosting y Firebase Authentication, y los entornos de ejecución son Firebase Storage y RealTimeDatabase. En Firebase Storage nos encontramos dos artefactos: users y zonas_baño; y en Real Time Database encontramos dos artefactos: Favoritos y Comentarios.

En la Figura 5.41 podemos observar el diagrama de despliegue.

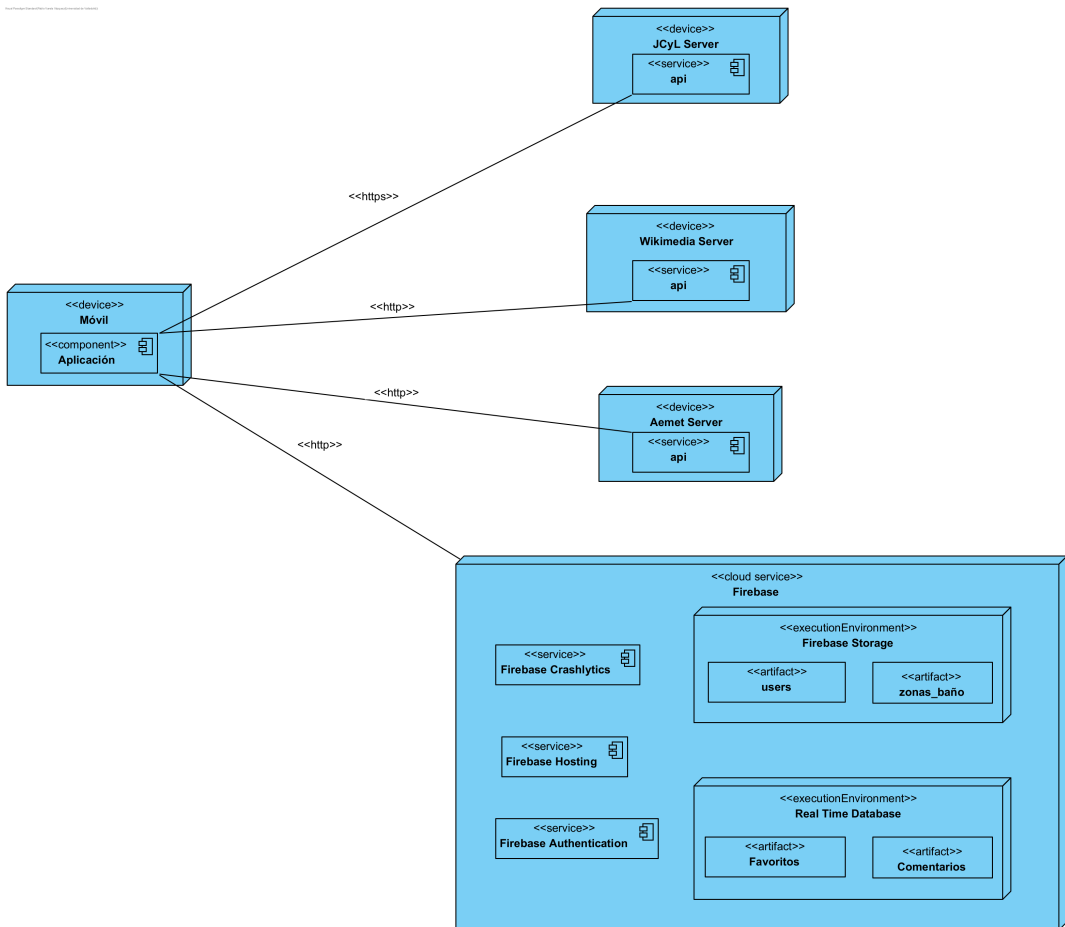


Figura 5.41: Diagrama de despliegue

Capítulo 6

Implementación y pruebas

En este capítulo, se abordará algunos de los detalles durante el proceso de implementación y se expondrá las pruebas realizadas para validar el funcionamiento de la aplicación.

6.1. Implementación

6.1.1. Principales problemas y retos

A la hora de realizar la implementación, se han detectado los siguientes problemas y retos:

- **La fuente de Datos Abiertos de la Junta de Castilla y León no estaba disponible.** Se ha tenido que pedir la disponibilidad de datos abiertos. Mientras tanto, se ha estado trabajando con otros datos disponibles hasta que estuvieran disponibles. La fecha de recepción del correo con la disponibilidad de datos ha sido el 01/04/2024 y se ha implementado dicho cambio en el Sprint 3. Los datos disponibles deberían actualizarse cada 15 días en temporada de baño (15 de junio a 15 de septiembre) por la Junta de Castilla y León, según la solicitud presentada.
- **Los datos de AEMET eran un JSON mal formado:** Disponía de una corchete de apertura como primer carácter del archivo JSON y una corchete de cierre como último carácter de dicho archivo. Se ha tenido que crear un método para eliminar los dos caracteres sobrantes y poder mapear correctamente el archivo JSON.
- **Fallo en el servicio de AEMET desde el 4 de mayo de 2024 al 11 de mayo de 2024:** El 4 de mayo de 2024, durante la realización del Sprint 5 se detectó que el servicio de AEMET para consultar la meteorología no funcionaba correctamente y no cargaba la información del tiempo. Se ha reportado dicho fallo a la AEMET el 8 de mayo de 2024 y se recibió respuesta el 9 de mayo de 2024. El error quedó solucionado el 11 de mayo de 2024 y la aplicación volvió a funcionar correctamente. A raíz de este error,

se ha aprovechado para realizar un cambio en el manejo de los datos en la pantalla de detalles.

- **Subida a Google Play:** El proceso de subida a Google Play de una aplicación era algo que no había realizado nunca y ha sido un reto en cuanto a aprender como se sube y distribuye las aplicaciones por Google Play, así como adaptar el desarrollo de esta aplicación y ampliar el proyecto inicial al desarrollo de dos páginas web estáticas para adaptarse a los requisitos que requiere Google para subir una aplicación a su tienda.
- **Cambio de guía de diseño en la aplicación:** La aplicación se comenzó desarrollando siguiendo la guía de estilos de Material 2. En el Sprint 4, se decidió adaptar la aplicación para seguir la guía de estilos de Material 3. Esto supuso adaptar todos los componentes de la aplicación.

6.1.2. Cambios en requisitos

Se ha producido los siguientes cambios con respecto a los requisitos planteados inicialmente:

- Se ha sustituido toda indicación a la palabra Incidencia por la palabra Comentario, especificando los tipos de comentario que existen.
- Se ha añadido el RF11, que indica que como usuario quiero saber si hay actualizaciones de la aplicación para ver si estoy en la última versión.
- Se ha añadido el RF12, que indica que como usuario quiero poder contactar con el desarrollador para notificar errores en la aplicación o para sugerir sugerencias.
- Se ha añadido el RF15, que indica que como usuario quiero poder iniciar sesión en la aplicación con mi cuenta de Google para poder disfrutar de las características extra de la aplicación

6.1.3. Guías de diseño seguidas

Para el diseño de la aplicación se está aplicando guías de diseño de **Material Design 3 (Material You)** desarrolladas por Google.

Material Design [120] es un lenguaje de diseño desarrollado por Google en el año 2014 y presentado durante la conferencia de desarrolladores Google I/O. El objetivo de Google fue crear un lenguaje de diseño que combinase los principios del buen diseño y la innovación técnica y científica. Google afirmó que este lenguaje de diseño se basa en el papel y en la tinta.

Material Design ha tenido 3 versiones distintas:

- **Material Design (2014):** Fue la versión original de Material Design.

- **Material Design 2 (2018)**: Esta versión enfatiza en los espacios en blanco, las esquinas redondeadas, los iconos coloridos y la barra de navegación inferior.
- **Material Design 3 (2021)**: Esta versión enfatiza en una mayor animación, botones más grandes, la capacidad de una interfaz de usuario personalizada y temas que se generan como fondo de pantalla del usuario.

La aplicación en su etapa de desarrollo inicial seguía las líneas de diseño de Material 2. En el Sprint 4 se decidió adaptar la interfaz de la aplicación a Material 3.

El cambio principal fue eliminar el elemento Drawer de la pantalla principal y sustituirlo por una barra de navegación inferior. Además esto ha conllevado otros cambios menores como el cambio de estilo en los botones, AppBar, Card, texto, diálogos, entre otros elementos.

6.1.4. Licencia

Se va a emplear para el software desarrollado una **Licencia de Software Propietaria** [51], es decir, que el titular limita la copia, distribución, modificación o desembalsado y puede establecer diferentes condiciones sobre el uso de software. Concretamente, vamos a utilizar la licencia **UELA (End User License Agreement)** [38], en la cual se especifican los derechos y las limitaciones sobre dicho software.

El repositorio de GitHub permanecerá privado debido a que no se puede garantizar que los miembros del tribunal disponga de una cuenta en GitHub. Para la evaluación de este Trabajo Fin de Grado se ha creado un repositorio espejo en el GitLab de la Escuela.

La aplicación se distribuirá única y exclusivamente a través de Google Play.

6.1.5. Organización del repositorio

El repositorio dispone de 3 carpetas, un gitignore y un README. Esas tres carpetas serán **app**, que contiene los archivos de la aplicación; **models**, que contiene el archivo de Visual Paradigm con los diagramas de los modelos; y **web**, que contiene el proyecto de la web para la política de privacidad y la eliminación de la cuenta. Esta disposición se puede ver en la Figura 6.1.

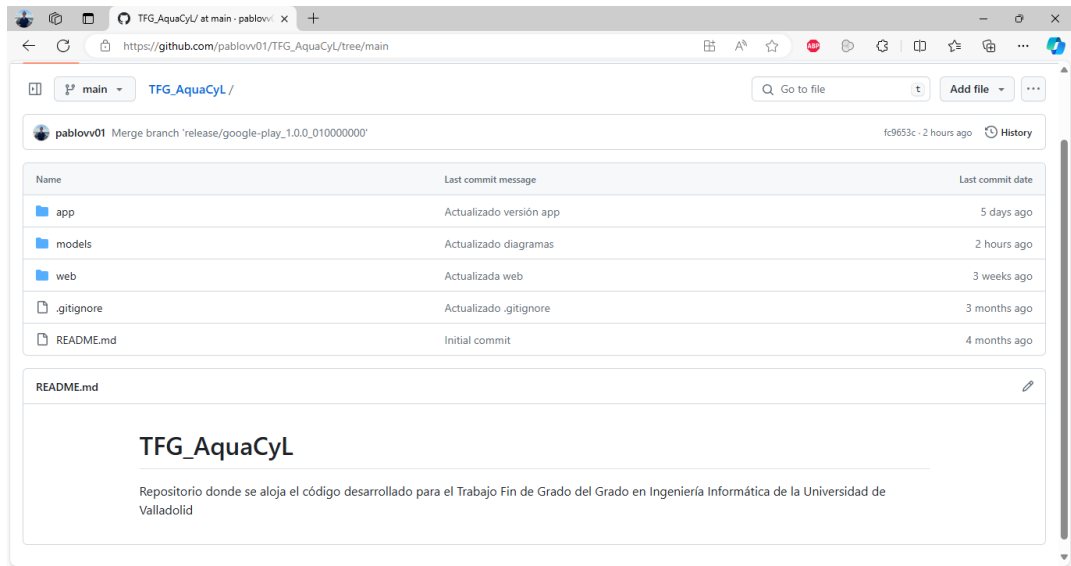


Figura 6.1: Estructura del proyecto

En la Figura 6.2 podemos ver la estructura que tiene el proyecto de la aplicación. En esta carpeta nos encontraremos [7]:

- **android:** Contiene todas las carpetas y archivos necesarios para poder ejecutar la aplicación de flutter en los dispositivos Android. Contiene archivos de configuración de Gradle y archivos Kotlin y Java.
- **assets:** Contiene todos los archivos que se utilizan en nuestro proyecto como imágenes.
- **build:** En este directorio se crean los archivos compilados que son necesarios para ejecutarse en los diferentes dispositivos.
- **ios:** Contiene todo lo relacionado con iOS, ya sea las configuraciones, los ejecutables, etcétera.
- **lib:** En este directorio se encuentra todo el código fuente de nuestra aplicación que se comparte en las diferentes plataformas.
- **.metadata:** Se encarga de rastrear todas las propiedades del proyecto Flutter.
- **analysis_options.yaml:** Es un archivo que contiene todas las reglas y configuraciones que va a tener el analizador a la hora de analizar el código de Flutter.
- **aquacyl.iml:** Es un archivo con formato xml que se utiliza para procesos internos.
- **pubspec.yaml:** Este archivo se utiliza para definir la configuración de nuestro proyecto y para administrar las dependencias o paquetes que instalamos en el proyecto *Flutter*.

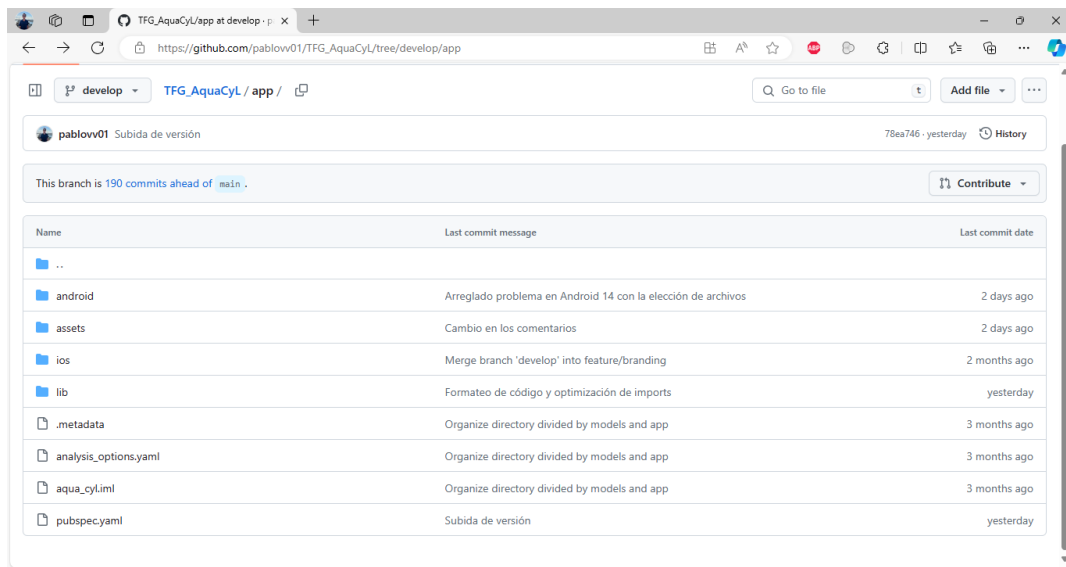


Figura 6.2: Estructura de la carpeta de aplicación

En cuanto a la distribución de la carpeta lib, esta se muestra en la Figura 6.3. En ella podemos observar los siguientes elementos:

- La carpeta **entities** contiene las clases presentadas en el modelo de dominio.
- La carpeta **net** contiene todo lo relacionado con la conexión a internet. En ellas podemos ver las siguientes tres carpetas:
 - api: Contiene las clases que representan a las APIs utilizadas en nuestra aplicación.
 - interceptors: Contiene las clases que representan un interceptor, es decir, [47] fragmentos de código que se activan cada vez que se realiza o se recibe una petición HTTP.
 - repository: Corresponde con la parte de modelo de la arquitectura MVVM.
- La carpeta **utils** contiene 3 clases que nos van a servir en el proyecto.
- La carpeta **views** contiene las vistas (View de la arquitectura MVVM) y los controladores de vista (View-Model de la arquitectura MVVM) divididos en las diferentes carpetas según la vista.
- La carpeta **widgets** contiene diferentes widgets de elaboración propia que son utilizados en más de una vista o que por su complejidad se ha decidido extraerlos en una clase.
- El archivo **app.dart** es la clase de lanzamiento de nuestra aplicación.
- El archivo **firebase_options.dart** contiene la configuración de Firebase de nuestra aplicación.

- El archivo **initial_bindings.dart** contiene la configuración inicial de algunos aspectos comunes que se utilizarán en la aplicación como el paquete Dio.
- El archivo **main.dart** es el punto de lanzamiento de nuestra aplicación. Se encarga de realizar alguna inicialización y configuración y lanza la clase App.
- El archivo **routes.dart** contiene todas las rutas a las diferentes vistas y la inicialización de sus respectivos controladores, así como la configuración de las animaciones.

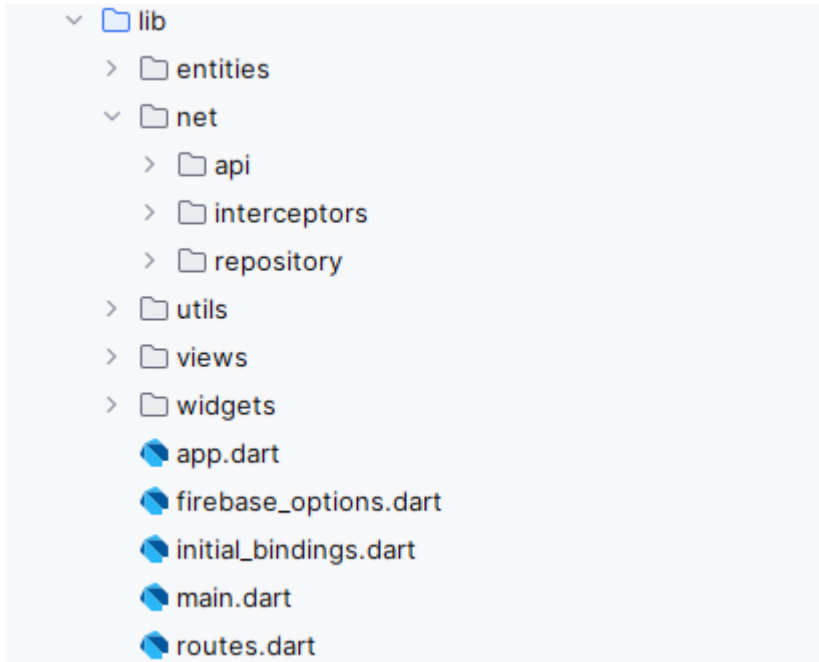


Figura 6.3: Estructura de la carpeta lib dentro de la aplicación

6.2. Pruebas

Además de la implementación de las funcionalidades en nuestra aplicación, se ha realizado una serie de pruebas para verificar el correcto funcionamiento de la aplicación. Las pruebas realizadas han sido **pruebas de aceptación de usuario (UAT)** [115], que son una serie de pruebas que verifican si el producto o servicio cumplen con los requisitos especificados.

Además, se ha distribuido la aplicación a partir del Sprint 6 por el canal de pruebas cerradas de Google Play para que personas reales puedan probar la aplicación en un contexto real de producción. Se ha probado la aplicación con 29 personas.

A mayores, según se han ido implementado la funcionalidad de la aplicación, se ha ido haciendo pequeñas pruebas.

6.2.1. Pruebas de aceptación de usuario (UAT)

En esta sección, se describirá las pruebas de aceptación de usuario llevadas a cabo en los diferentes dispositivos.

Para realizar estas pruebas, se va a utilizar 3 dispositivos diferentes con diferente versión de Android. A continuación, en las Tablas 6.1 a 6.3 se detallan las características de estos dispositivos.

Características Samsung Galaxy A54	
Nombre del dispositivo	Galaxy A54 5G
Fabricante	Samsung
Versión Android	Android 14
Versión One UI (capa personalización)	One UI 6.1
Actualización sistema Google Play	1 de abril de 2024

Tabla 6.1: Características Samsung Galaxy A54

Características Samsung Galaxy A03	
Nombre del dispositivo	Galaxy A03
Fabricante	Samsung
Versión Android	Android 13
Versión One UI (capa personalización)	One UI 5.1
Actualización sistema Google Play	1 de junio de 2023

Tabla 6.2: Características Samsung Galaxy A03

Características Nokia 6.1	
Nombre del dispositivo	Nokia 6.1
Fabricante	HMD (Nokia)
Versión Android	Android 10
Actualización sistema Google Play	1 de marzo de 2020

Tabla 6.3: Características Nokia 6.1

He considerado apropiado llevar a cabo las pruebas en tres dispositivos diferentes con diferente versión de Android debido a las diferencias existentes entre versiones, como por ejemplo, en el caso de los permisos.

Las pruebas de aceptación de usuario se han realizado sobre la versión 0.0.10 de la aplicación y se pueden ver reflejadas en las Tablas 6.4 a 6.39.

6.2. PRUEBAS

Prueba 01	
Título	Continuar como invitado
Descripción	El usuario puede entrar en el listado de las zonas de baño sin necesidad de crearse una cuenta.
Resultados esperados	El usuario accede sin cuenta
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se presiona en el botón de la imagen de perfil y se comprueba que el usuario sea “Invitado”.
Observaciones	

Tabla 6.4: Prueba UAT 01

Prueba 02	
Título	Iniciar sesión con email y contraseña
Descripción	El usuario accede con un correo electrónico y contraseña vinculados a una cuenta previamente creada.
Resultados esperados	El usuario accede con su cuenta de correo electrónico y ve el listado de zonas de baño.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se presiona en el botón de la imagen de perfil y se comprueba que aparece el correo electrónico y el nombre del usuario
Observaciones	

Tabla 6.5: Prueba UAT 02

Prueba 03	
Título	Iniciar sesión con una cuenta de Google
Descripción	El usuario accede usando su cuenta de Google.
Resultados esperados	El usuario accede con su cuenta de Google y se muestra la pantalla de las zonas de baño.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se presiona en el botón de la imagen de perfil y se comprueba que aparece el correo electrónico y el nombre del usuario vinculados a la cuenta de Google del usuario.
Observaciones	Para que se cumpla esta prueba, los usuarios deben tener vinculada una cuenta Google en su telefono de Android.

Tabla 6.6: Prueba UAT 03

Prueba 04	
Título	Registrarse
Descripción	El usuario se registra para poder obtener una cuenta.
Resultados esperados	Se completa correctamente el registro y se envía un correo electrónico para confirmar la creación de la cuenta
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra un popup proporcionando instrucciones y se recibe un correo electrónico
Observaciones	

Tabla 6.7: Prueba UAT 04

Prueba 05	
Título	Contraseña no coincide al registrarse
Descripción	En el proceso de registro las contraseña y la repetición de la contraseña no coinciden.
Resultados esperados	No permite registrarse.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Al pulsar el botón registrarse no realiza ninguna acción y se ilumina de rojo el cuadro de texto de la repetición de la contraseña, indicando que no coinciden las contraseñas
Observaciones	

Tabla 6.8: Prueba UAT 05

Prueba 06	
Título	No se acepta la política de privacidad
Descripción	Al registrarse, no se marca el cuadro de texto donde se indica que se acepta la Política de Privacidad de la aplicación.
Resultados esperados	Se muestra un mensaje indicando que no se ha aceptado la Política de Privacidad.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra un mensaje indicando que no se ha aceptado la Política de Privacidad.
Observaciones	

Tabla 6.9: Prueba UAT 06

6.2. PRUEBAS

Prueba 07	
Título	La cuenta con la que se va a registrar ya existe
Descripción	Al registrarse, se detecta que existe una cuenta registrada con dicho e-mail.
Resultados esperados	Se muestra un mensaje indicando que la cuenta ya existe.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra un mensaje indicando que la cuenta ya existe.
Observaciones	

Tabla 6.10: Prueba UAT 07

Prueba 08	
Título	La contraseña es incorrecta
Descripción	Al iniciar sesión, se introduce una contraseña que es incorrecta
Resultados esperados	Se informa de que la contraseña es incorrecta
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra un mensaje indicando que la información de inicio de sesión no es correcta.
Observaciones	

Tabla 6.11: Prueba UAT 08

Prueba 09	
Título	He olvidado la contraseña
Descripción	Si he olvidado la contraseña, puedo recuperarla a través de un enlace que se envía por correo electrónico.
Resultados esperados	Se recibe el correo electrónico y se cambia la contraseña correctamente.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se recibe el correo y puedo iniciar sesión con la nueva contraseña.
Observaciones	

Tabla 6.12: Prueba UAT 09

Prueba 10	
Título	He olvidado la contraseña
Descripción	Si he olvidado la contraseña, puedo recuperarla a través de un enlace que se envía por correo electrónico.
Resultados esperados	Se recibe el correo electrónico y se cambia la contraseña correctamente.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se recibe el correo y puedo iniciar sesión con la nueva contraseña.
Observaciones	

Tabla 6.13: Prueba UAT 10

Prueba 11	
Título	Listado de zonas de baño
Descripción	Mostrar las zonas de baño disponibles.
Resultados esperados	Se muestra un listado de zonas de baño disponibles.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra las zonas de baño.
Observaciones	

Tabla 6.14: Prueba UAT 11

Prueba 12	
Título	Listado de zonas de baño
Descripción	Mostrar las zonas de baño disponibles.
Resultados esperados	Se muestra un listado de zonas de baño disponibles.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra las zonas de baño.
Observaciones	

Tabla 6.15: Prueba UAT 12

Prueba 13	
Título	Filtrar zonas de baño.
Descripción	Filtrar zonas de baño por estado y por provincia.
Resultados esperados	Se muestra las zonas de baño que están situadas en la provincia seleccionada y que presentan el estado seleccionado.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	La lista de zonas de baño mostradas presentan el estado y la provincia seleccionados.
Observaciones	

Tabla 6.16: Prueba UAT 13

Prueba 14	
Título	Ordenar las zonas de baño.
Descripción	Ordenar las zonas de baño por nombre, localidad, municipio, provincia o estado.
Resultados esperados	Se muestran las zonas de baño ordenados por el parámetro seleccionado.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestran las zonas de baño ordenadas por el parámetro seleccionado.
Observaciones	

Tabla 6.17: Prueba UAT 14

Prueba 15	
Título	Ordenar las zonas de baño.
Descripción	Ordenar las zonas de baño por nombre, localidad, municipio, provincia o estado.
Resultados esperados	Se muestran las zonas de baño ordenados por el parámetro seleccionado.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestran las zonas de baño ordenadas por el parámetro seleccionado.
Observaciones	

Tabla 6.18: Prueba UAT 15

Prueba 16	
Título	Detalles de una zona de baño.
Descripción	Ver los detalles de una zona de baño, como la puntuación, información sobre la localidad, la previsión del tiempo o las fotos.
Resultados esperados	Se debe mostrar la información que esté disponible.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra la información disponible
Observaciones	

Tabla 6.19: Prueba UAT 16

Prueba 17	
Título	Añadir o eliminar las zonas de baño a favoritos.
Descripción	Se debe poder añadir o eliminar la zona de baño de favoritos.
Resultados esperados	Se añade o se elimina la zona de baño.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba que se añade o se elimina de favoritos desde la consola de Firebase.
Observaciones	

Tabla 6.20: Prueba UAT 17

Prueba 18	
Título	Ver alojamientos.
Descripción	Se debe poder acceder a los alojamientos disponibles para esa zona de baño a través de la pestaña 'Alojamientos'.
Resultados esperados	Se accede a la página web que se encarga de la reserva de los alojamientos.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Carga la web embebida dentro de la aplicación
Observaciones	

Tabla 6.21: Prueba UAT 18

Prueba 19	
Título	Obtener direcciones.
Descripción	Obtener direcciones a la localidad donde se sitúa la zona de baño.
Resultados esperados	Abre la aplicación de Google Maps que se encuentra instalada en el teléfono y carga con la información de la localidad.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se carga Google Maps con la localidad de la zona de baño.
Observaciones	

Tabla 6.22: Prueba UAT 19

Prueba 20	
Título	Obtener direcciones.
Descripción	Obtener direcciones a la localidad donde se sitúa la zona de baño.
Resultados esperados	Abre la aplicación de Google Maps que se encuentra instalada en el teléfono y carga con la información de la localidad.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se carga Google Maps con la localidad de la zona de baño.
Observaciones	

Tabla 6.23: Prueba UAT 20

Prueba 21	
Título	Añadir fotos
Descripción	Añadir fotos sobre las zonas de baño.
Resultados esperados	Se añade la foto correctamente, se almacena y se muestra en los detalles de la zona de baño.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Comprobación manual de que la foto se encuentre cargada en Firebase y se muestra la foto en el detalle.
Observaciones	

Tabla 6.24: Prueba UAT 21

Prueba 22	
Título	Ver y añadir comentarios
Descripción	Ver los comentarios publicados en una zona de baño y añadir un nuevo comentario en dicha zona de baño
Resultados esperados	Se espera que se muestre los comentarios publicados en la zona de baño y que al publicar se muestre el comentario nuevo publicado en la parte superior de la lista de comentarios.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se añade el comentario en la aplicación y comprobación manual en Firebase.
Observaciones	

Tabla 6.25: Prueba UAT 22

Prueba 23	
Título	Ver lista de favoritos
Descripción	Ver una lista con los favoritos que tenemos guardados.
Resultados esperados	Se espera que se muestre la lista con todos los favoritos de ese usuario.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra la pantalla de favoritos y se hace comprobación manual en Firebase.
Observaciones	

Tabla 6.26: Prueba UAT 23

Prueba 24	
Título	Anclar favoritos desde la lista de favoritos
Descripción	Anclar una zona de baño favorita para que aparezca en la parte superior.
Resultados esperados	Se espera que esa zona se encuentre elevada sobre las demás, se añada un icono de una estrella y el fondo sea de diferente color que el resto.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba el aspecto visual de la pantalla de favoritos.
Observaciones	

Tabla 6.27: Prueba UAT 24

6.2. PRUEBAS

Prueba 25	
Título	Anclar favoritos desde la lista de favoritos
Descripción	Anclar una zona de baño favorita para que aparezca en la parte superior.
Resultados esperados	Se espera que esa zona se encuentre elevada sobre las demás, se añada un icono de una estrella y el fondo sea de diferente color que el resto.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba el aspecto visual de la pantalla de favoritos.
Observaciones	

Tabla 6.28: Prueba UAT 25

Prueba 26	
Título	Eliminar favoritos desde la lista de favoritos
Descripción	Elimina una zona de baño favorita de la lista de favoritos
Resultados esperados	Esa zona debería desaparecer de la lista de favoritos.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba el aspecto visual de la pantalla de favoritos y se comprueba manualmente Firebase.
Observaciones	

Tabla 6.29: Prueba UAT 26

Prueba 27	
Título	Modificar el nombre de usuario
Descripción	Se modifica el nombre de usuario del usuario de la aplicación.
Resultados esperados	Se debe actualizar el nombre de usuario de la cuenta en toda la aplicación
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba que en la pantalla se haya actualizado el nombre de usuario, al igual que en la pantalla principal.
Observaciones	Esto se puede hacer en las cuentas que no sean una cuenta de Google.

Tabla 6.30: Prueba UAT 27

Prueba 28	
Título	Cambiar foto de perfil
Descripción	Se cambia o se elimina la foto de perfil del usuario.
Resultados esperados	Se cambie o elimine la foto de perfil y se muestre la actualización en todos los lugares de la aplicación.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba el aspecto visual de la pantalla principal y de cuenta.
Observaciones	Esto se puede hacer en las cuentas que no sean una cuenta de Google.

Tabla 6.31: Prueba UAT 28

Prueba 29	
Título	Cambiar contraseña
Descripción	Se cambia la contraseña de la cuenta.
Resultados esperados	La contraseña se ha cambiado.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se cierra sesión y se vuelve a iniciar sesión con la nueva contraseña.
Observaciones	Esto se puede hacer en las cuentas que no sean una cuenta de Google. Además, se puede pedir reautenticación.

Tabla 6.32: Prueba UAT 29

Prueba 30	
Título	Eliminar cuenta
Descripción	Se elimina la cuenta de la aplicación.
Resultados esperados	La cuenta se elimina con éxito
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se intenta iniciar sesión con dicha cuenta y se hace comprobación con el listado de usuarios en Firebase
Observaciones	Se puede pedir reautenticación.

Tabla 6.33: Prueba UAT 30

Prueba 31	
Título	Cambiar idioma
Descripción	Se puede cambiar el idioma de la aplicación a inglés o francés.
Resultados esperados	El idioma de la aplicación debe ser el seleccionado
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba cambiando el idioma y observando que el idioma de la aplicación haya cambiado.
Observaciones	

Tabla 6.34: Prueba UAT 31

Prueba 32	
Título	Actualización no disponible
Descripción	No hay actualización de la aplicación disponible para descargar.
Resultados esperados	Se muestra un dialogo indicando que no hay actualización disponible.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se muestra el dialogo indicado anteriormente
Observaciones	

Tabla 6.35: Prueba UAT 32

Prueba 33	
Título	Actualización disponible
Descripción	Muestra si hay una actualización disponible
Resultados esperados	Descarga de la nueva versión disponible en Google Play.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se sube una versión nueva a Google Play en canal interno y se prueba a pulsar el botón.
Observaciones	

Tabla 6.36: Prueba UAT 33

Prueba 34	
Título	Cambiar tema de la aplicación
Descripción	Cambio del tema de la aplicación a claro, oscuro o predeterminado del sistema
Resultados esperados	Al cambiar el tema, se muestra la aplicación con el tema seleccionado.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se comprueba el aspecto visual de la aplicación.
Observaciones	

Tabla 6.37: Prueba UAT 34

Prueba 35	
Título	Proporcionar Feedback al desarrollador
Descripción	Se proporciona feedback de la aplicación al desarrollador.
Resultados esperados	La aplicación abre la aplicación de correo predeterminada para enviar un correo.
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se envía un correo de prueba con un mensaje sencillo.
Observaciones	

Tabla 6.38: Prueba UAT 35

Prueba 36	
Título	Enviar notificación
Descripción	El desarrollador envía notificaciones para informar sobre aspectos relacionados con la aplicación.
Resultados esperados	Se recibe la notificación
Resultado (Pasa/ No Pasa)	Pasa
Comprobación	Se envía una notificación desde Firebase y se recibe en el dispositivo.
Observaciones	

Tabla 6.39: Prueba UAT 36

En estas pruebas he detectado los siguientes errores:

1. Al guardar un favorito nuevo, la lista de favoritos no se recargaba. Por lo que el nuevo favorito no se mostraba.
2. En Android 14, no se podía adjuntar archivos a la hora de enviar feedback al desarrollador.

6.2. PRUEBAS

Estos fallos se han detectado, corregido y probado previamente a ser liberados en la versión 1.0.0.

6.2.2. Prueba en canal cerrado de Google Play

El día 28 de mayo de 2024, se inició la prueba cerrada después de que Google haya aprobado la documentación y la aplicación para la prueba. La prueba se ha realizado con 27 personas y durante estas pruebas se han reportado las incidencias que se pueden ver en la Tabla 6.40. Para mantener el anonimato de los testers, no se mencionará la identidad de la persona que ha reportado la incidencia.

ID	Descripción
IN01	La pantalla de favoritos no está traducida.
IN02	Las provincias no se ordenan adecuadamente. Por ejemplo, se coloca Ávila después de Zamora.
IN03	En la vista principal, el título de las zonas de baño aparecía en algunos casos alineado a la izquierda y en otros casos centrado.

Tabla 6.40: Fallos detectados en la aplicación

Además se ha propuesto las mejoras que se pueden ver en la Tabla 6.41.

ID	Descripción
PR01	Añadir el símbolo de grados en la previsión meteorológica de las zonas de baño

Tabla 6.41: Propuestas presentadas por testers

Los fallos detectados y los cambios propuestos se han aceptado y se han implementado en la aplicación.

En la versión 0.0.9 de la aplicación se ha implementado la corrección a las incidencias IN01 e IN02. En la versión 0.0.10 de la aplicación se ha implementado la corrección a la incidencia IN03 y la propuesta PR01.

Capítulo 7

Seguimiento del proyecto

En este capítulo, se detalla el seguimiento del proyecto realizado. Durante el tiempo dedicado a cada Sprint, salvo en el Sprint 0, se realizarán Historias de Usuario a la vez con la redacción de la memoria. Adicionalmente, en algunos sprints se realizarán algunas tareas no incluidas en las Historias de Usuario, pero que son necesarias para el producto final.

En cada Sprint, se proporcionará las tareas realizadas indicando el tiempo estimado para la realización de las tareas, el tiempo empleado para la realización de las tareas, algún comentario a destacar de la reunión de *Sprint Retrospective* y las tareas planificadas en la reunión de *Sprint Planning* que se realizarán en el siguiente Sprint.

7.1. Seguimiento de los Sprints

7.1.1. Sprint 0 (20/02/2024 - 29/02/2024)

El Sprint 0 es el sprint en el que se van a realizar algunas tareas de preparación al proyecto que se va a desarrollar en los sucesivos Sprints, así como de formación sobre las tecnologías que se van a usar.

En concreto, se han realizado las siguientes tareas:

1. **Definición del nombre y del resumen del proyecto:** Se ha definido el nombre del proyecto y se ha redactado el resumen inicial en la memoria.
2. **Preparación de las herramientas:** Se ha creado un repositorio en GitHub, se ha creado un proyecto en Firebase para poder usarlo como Backend y se ha definido la plantilla base de la aplicación Flutter. Además, se ha configurado Jira para poder utilizarlo.

3. **Estudio de las tecnologías y metodologías a utilizar:** Se ha estudiado el marco de trabajo ágil Scrum, algunos aspectos de Firebase y algunos aspectos de Flutter.
4. **Calendarización inicial:** Se ha elaborado una planificación inicial para el desarrollo del proyecto.
5. **Redacción de algunos apartados de la introducción de la memoria:** Se ha redactado los apartados relativos al contexto, la motivación y los objetivos.
6. **Definición del backlog inicial:** Se han definido las épicas que forman parte del backlog inicial.
7. **Definición de los requisitos funcionales:** A partir de las épicas del backlog inicial se han definido los requisitos funcionales como historias de usuario.
8. **Elaboración del plan de riesgos:** Se ha elaborado un plan de riesgos que se pueden dar a lo largo del desarrollo de este proyecto.
9. **Elaboración del plan de presupuestos:** Se ha elaborado un plan de presupuestos simulado, que sería el que se daría si este proyecto se desarrollara en la vida real, y el presupuesto real, que es el que se da teniendo en cuenta el entorno académico de este proyecto.
10. **Elaboración del prototipo:** Se ha elaborado un prototipo de la aplicación en la que se muestran algunos aspectos de como funcionaría la aplicación.

En la Tabla 7.1 se detallan las actividades realizadas y las horas que se ha tardado en realizar cada una de las tareas.

Tareas realizadas en el Sprint 0	
Tarea	Tiempo empleado
Definición del nombre y resumen del proyecto	1 hora
Preparación de las herramientas	40 minutos
Estudio de las tecnologías y metodologías	6 horas
Calendarización inicial	3 horas y 30 minutos
Redacción de algunos apartados de la introducción	1 hora y 42 minutos
Definición del Backlog Inicial	1 hora
Definición de los requisitos funcionales	1 hora
Elaboración del Plan de riesgos	2 horas y 18 minutos
Elaboración del Plan de Presupuestos	4 horas y 23 minutos
Elaboración del Prototipo	8 horas y 32 minutos
Total	30 horas y 5 minutos

Tabla 7.1: Tareas realizadas en el Sprint 0

El total de horas ha sido **30 horas y 5 minutos**.

En la reunión que se mantuvo con la tutora el día 29 de febrero se estableció las siguientes tareas para el Sprint 1:

1. Modificación del prototipo para añadir nuevos elementos
2. Redacción de los requisitos no funcionales y de información.
3. Redacción de las reglas de negocio para la aplicación.
4. Redacción de algunos apartados del capítulo 2
5. Redacción de algunos apartados del capítulo 4
6. Completar los apartados del Capítulo 7 relativos a Sprint 0 y Sprint 1.
7. Elaboración del modelo de dominio
8. Desarrollo del RF01
9. Desarrollo del RF16
10. Internacionalización de la aplicación

7.1.2. Sprint 1 (29/02/2024 - 14/03/2024)

En el Sprint 1 se ha realizado tareas de redacción en la memoria, tareas de análisis y desarrollo de algunas historias de usuario de nuestra aplicación.

En la Tabla 7.2 se pueden ver las tareas realizadas durante el Sprint 1.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Modificación del prototipo para añadir nuevos elementos	3 horas	1 hora y 30 minutos	Completo
Redacción de los requisitos no funcionales y de información	3 horas	3 horas y 15 minutos	Completo
Redacción de las reglas de negocio para la aplicación	3 horas	34 minutos	Completo
Redacción de algunos apartados del capítulo 2	4 horas	4 horas y 18 minutos	Completo
Redacción de algunos apartados del capítulo 4	3 horas	50 minutos	Completo
Completar los apartados del Capítulo 7 relativos a Sprint 0 y Sprint 1	3 horas	2 horas y 30 minutos	Completo
Elaboración del modelo de dominio	4 horas	3 horas y 30 minutos	Completo
Desarrollo del RF01	10 horas	9 horas y 30 minutos	Pendiente API definitiva
Desarrollo del RF16	4 horas	2 horas	Completa
Internacionalización de la aplicación	3 horas	3 horas	Completa
Total	40 horas	30 horas y 57 minutos	9:10

Tabla 7.2: Tareas realizadas en el Sprint 1

El tiempo total de trabajo en el Sprint 1 ha sido de **30 horas y 57 minutos**.

En la *Sprint Retrospective* se determino lo siguiente:

- Se ha cumplido los objetivos para el sprint y el incremento se considera valido para el siguiente sprint.
- La API usada para completar el RF01 es provisional. Se sustituirá y adaptará la nueva API en cuanto este disponible.
- Hubo menos tiempo para realizar el trabajo para este sprint debido a la coincidencia de la entrega de un trabajo de la asignatura.

En el *Sprint Planning* para el siguiente sprint se determinó que se iban a realizar las siguientes tareas:

1. Documentar arquitectura
2. Corrección de algunos aspectos pequeños de la memoria
3. Redacción de algunos apartados del capítulo 5: Diseño
4. Redacción de algunos apartados del capítulo 6: Implementación y pruebas
5. Redacción de algunos apartados del capítulo 4: Tecnologías utilizadas.
6. Ampliación del modelo de dominio
7. Desarrollo del RF02
8. Desarrollo del RF03
9. Desarrollo del RF05
10. Desarrollo del RF06
11. Desarrollo del RF07
12. Diseño del logotipo de la aplicación

7.1.3. Sprint 2 (14/03/2024 - 04/04/2024)

En el Sprint 2 se han realizado tareas de redacción de la memoria y de desarrollo de algunas historias de usuario de la aplicación.

En la Tabla 7.3 se pueden ver las tareas planificadas para el Sprint 2 con los tiempos estimados, los tiempos empleados y su estado.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Corrección de algunos pequeños aspectos de la memoria	2 horas	2 horas y 2 minutos	Completo
Redacción de algunos apartados del capítulo 5	3 horas	4 horas	Completo
Redacción de algunos apartados del capítulo 6	3 horas	44 minutos	Completo
Redacción de algunos apartados del capítulo 4	3 horas	3 horas y 21 minutos	Completo
Ampliación del modelo de dominio	1 hora	43 minutos	Completo
Desarrollo del RF02	3 horas	2 horas y 15 minutos	Completo
Desarrollo del RF03	3 horas	2 horas y 30 minutos	Completo
Desarrollo del RF05	20 horas	26 horas	Completo
Desarrollo del RF06	30 minutos	20 minutos	Completo
Desarrollo del RF07	4 horas	3 horas y 10 minutos	Completado
Diseño logotipo aplicación	2 hora	3 horas y 30 minutos	Completado
Total	43 horas y 30 minutos	48 horas y 35 minutos	11:11

Tabla 7.3: Tareas realizadas en el Sprint 2

El tiempo total de trabajo en el Sprint 2 ha sido de **48 horas y 35 minutos**.

En la *Sprint Retrospective* se determino lo siguiente:

- Se ha cumplido los objetivos para el sprint y el incremento se considera valido para el siguiente sprint.
- Se ha realizado una mayor cantidad de tareas y horas debido a que se ha empleado tiempo de las vacaciones de Semana Santa

En el *Sprint Planning* para el siguiente sprint se determinó que se iban a realizar las siguientes tareas:

1. Desarrollo del RF14
2. Desarrollo del RF15
3. Desarrollo del RF17
4. Desarrollo del RF21
5. Completar el RF01
6. Diagrama de despliegue
7. Redacción de algunos apartados del capítulo 5: Diseño
8. Redacción de algunos apartados del capítulo 6: Implementación y pruebas
9. Redacción de algunos apartados del capítulo 4: Tecnologías utilizadas.

7.1.4. Sprint 3 (04/04/2024 - 18/04/2024)

En el Sprint 3 se han realizado tareas de redacción de la memoria y de desarrollo de algunas historias de usuario de la aplicación.

En la Tabla 7.4 se pueden ver las tareas planificadas para el Sprint 3 con los tiempos estimados, los tiempos empleados y su estado.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Desarrollo del RF14	5 horas	8 horas	Completo
Desarrollo del RF15	4 horas	4 horas	Completo
Desarrollo del RF17	3 horas	7 horas y 45 minutos	Completo
Desarrollo del RF21	7 horas	3 horas y 21 minutos	Completo
Completar el RF01	1 hora	3 horas	Completo
Diagrama de despliegue	2 horas	3 horas	Completo
Redacción de algunos apartados del capítulo 5	2 horas	1 horas	Completo
Redacción de algunos apartados del capítulo 6	2 horas	2 horas	Completo
Redacción de algunos apartados del capítulo 4	2 horas	4 horas	Completo
Redacción de algunos apartados del capítulo 7	2 horas	2 horas	Completo
Cambios en UI	1 hora	2 horas	Completo
Total	31 horas	40 horas y 6 minutos	11:11

Tabla 7.4: Tareas realizadas en el Sprint 3

El tiempo total de trabajo en el Sprint 3 ha sido de **40 horas y 6 minutos**.

En la *Sprint Retrospective* se ha determinado que se ha podido llevar a cabo correctamente todas las tareas de este sprint y no se ha manifestado ningún riesgo.

En el *Sprint Planning* para el siguiente sprint se determinó que se iban a realizar las siguientes tareas:

1. Desarrollo del RF10
2. Desarrollo del RF11
3. Desarrollo del RF12
4. Desarrollo del RF18
5. Desarrollo del RF22
6. Adaptación a Material Design 3
7. Redacción de algunos apartados del capítulo 3

8. Redacción de algunos apartados del capítulo 5
9. Redacción de algunos apartados del capítulo 4
10. Redacción de algunos apartados del capítulo 6
11. Redacción de algunos apartados del capítulo 7

7.1.5. Sprint 4 (18/04/2024 - 02/05/2024)

En el Sprint 4 se han realizado tareas de redacción de la memoria y de desarrollo de algunas historias de usuario de la aplicación.

En la Tabla 7.5 se pueden ver las tareas planificadas para el Sprint 4 con los tiempos estimados, los tiempos empleados y su estado.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Pequeños cambios en la Interfaz de Usuario	1 hora	1 hora y 9 minutos	Completo
Diseño de la página de ajustes	1 hora	2 horas	Completo
Desarrollo del RF10	3 horas	3 horas y 15 minutos	Completo
Desarrollo del RF11	3 horas	4 horas	Completo
Desarrollo del RF12	4 horas	4 horas y 17 minutos	Completo
Desarrollo del RF18	2 horas	2 horas y 38 minutos	Completo
Desarrollo del RF22	8 horas	8 horas y 51 minutos	Completo
Adaptación a Material Design 3	7 horas	5 horas	Completo
Redacción de algunos apartados del capítulo 3	1 hora	1 hora y 30 minutos	Completo
Redacción de algunos apartados del capítulo 5	3 horas	0 horas	Incompleto
Redacción de algunos apartados del capítulo 4	3 horas	2 horas	Completo
Redacción de algunos apartados del capítulo 6	3 horas	0 horas	Incompleto
Redacción de algunos apartados del capítulo 7	1 hora	50 minutos	Completo
Total	40 horas	35 horas y 30 minutos	11:13

Tabla 7.5: Tareas realizadas en el Sprint 4

El total del tiempo empleado en este sprint ha sido de **35 horas y 30 minutos**.

En la *Sprint Retrospective* se determino lo siguiente:

- No se ha podido dedicar tiempo a realizar redacción de capítulo 5 y 6 de esta memoria.
- Hubo menos tiempo para realizar el trabajo para este sprint debido a la coincidencia de la entrega de un trabajo de la asignatura, lo cual se encuentra dentro de los riesgos previstos.

En el *Sprint Planning* se ha establecido que se van a llevar a cabo las siguientes tareas:

1. Desarrollo del RF08
2. Desarrollo del RF09
3. Desarrollo del RF13
4. Configuración de las reglas definitivas de Firebase.
5. Redacción del Capítulo 3
6. Redacción del Capítulo 4
7. Redacción del Capítulo 5
8. Redacción del Capítulo 6
9. Redacción del Capítulo 7

7.1.6. Sprint 5 (02/05/2024 - 16/05/2024)

Durante el Sprint 5 nos centraremos más en la redacción de esta memoria y en el desarrollo de 3 requisitos funcionales relacionados con el bloque de comentarios y la configuración de las reglas de Firebase.

Además, se ha solucionado un error no detectado en el anterior sprint consistente en que las cuentas en las que se iniciaba sesión con Google daban un error al eliminarse debido a que buscaba la foto de perfil en Firebase Storage y esta para las cuentas Google no se encuentran almacenadas en ese servicio, sino que en los propios servidores de Google y que en Android 14, por un cambio en los permisos, los usuarios no podían adjuntar archivos por la aplicación al enviar un mensaje de Feedback.

En la Tabla 7.6 se pueden ver las tareas planificadas para el Sprint 5 con los tiempos estimados, los tiempos empleados y su estado al finalizar el sprint.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Desarrollo del RF08	4 horas	7 horas y 12 minutos	Completo
Desarrollo del RF09	4 horas	7 horas y 4 minutos	Completo
Desarrollo del RF13	4 horas	6 horas y 8 minutos	Completo
Configuración de las reglas definitivas de Firebase	3 horas	2 horas y 21 minutos	Completo
Redacción de algunos apartados del Capítulo 3	4 horas	4 horas y 11 minutos	Completo
Redacción de algunos apartados del Capítulo 4	4 horas	6 horas	Completo
Redacción de algunos apartados del Capítulo 5	7 horas y 30 minutos	30 minutos	Completo
Redacción de algunos apartados del Capítulo 6	8 horas	5 horas y 27 minutos	Completo
Redacción de algunos apartados del Capítulo 7	1 hora y 30 minutos	1 hora	Completo
Solucionar problemas detectados en Android 14 y con Cuentas de Google	1 hora	1 hora y 47 minutos	Completo
Total	41 horas	41 horas y 30 minutos	10:10

Tabla 7.6: Tareas realizadas en el Sprint 5

El tiempo empleado para la realización del Sprint 5 ha sido de **41 horas y 30 minutos**.

En la *Sprint Retrospective* se ha determinado que se ha podido llevar a cabo correctamente todas las tareas de este sprint y no se ha manifestado ningún riesgo. El tiempo utilizado a mayores en los requisitos funcionales se debe a cambios estructurales y de diseño en los requisitos funcionales y en no conocer a fondo el sistema de notificaciones de Firebase.

En el *Sprint Planning* para el siguiente sprint se determinó que se iban a realizar las siguientes tareas:

- Finalizar redacción del Capítulo 5
- Finalizar redacción del Capítulo 3
- Finalización de los requisitos funcionales restantes (RF04, RF19 y RF20)
- Páginas web de privacidad, cierre de cuenta y condiciones de uso.
- Preparación despliegue en Google Play (Canal de prueba cerrado).

7.1.7. Sprint 6 (16/05/2024 - 30/05/2024)

En el Sprint 6, se ha dedicado el tiempo a finalizar la implementación de la aplicación, desarrollar la página web del servicio (donde está alojadas las páginas necesarias), finalizar

7.1. SEGUIMIENTO DE LOS SPRINTS

la redacción del capítulo 3 y 5 de esta memoria y comenzar con el despliegue en Google Play, aportando toda la información requerida por Google.

Durante el desarrollo de este Sprint y en el *Scrum Weekly* se ha añadido las siguientes tareas:

- Corrección algunos errores detectados
- Adaptar la interfaz de usuario al modo tablet
- Creación de una vista principal para la página web.
- Actualización del Capítulo 4. En este capítulo se añaden las herramientas utilizadas para el desarrollo web, la incorporación de Google Analytics en la aplicación y la actualización de algunos paquetes de Flutter.
- Estudio de la posibilidad de guardar las zonas de baño en base de datos o normalizar. Al final, me he decantado por normalizar debido al alto consumo de tasa de descargas de Firebase Realtime Database.
- Ordenar las zonas de baño según los campos.

En la Tabla 7.7 se pueden ver las tareas desarrolladas en el Sprint 6 con los tiempos estimados, los tiempos empleados y su estado al finalizar el sprint.

Tareas	Tiempo estimado	Tiempo empleado	Estado
Desarrollo del RF19	6 horas	10 horas y 21 minutos	Completo
Desarrollo del RF20	6 horas	5 horas y 46 minutos	Completo
Desarrollo del RF04	5 horas	5 horas y 8 minutos	Completo
Desarrollo de las páginas web	4 horas	8 horas	Completo
Preparación del despliegue en Google Play	3 horas	2 horas y 27 minutos	Completo
Estudio de guardar zonas de baño en Base de Datos o normalizar	4 horas	6 horas y 28 minutos	Completo
Adaptación a tablet	3 horas	2 horas y 36 minutos	Completo
Ordenar las zonas de baño en pantalla principal	2 horas	2 horas y 27 minutos	Completo
Solución de errores en app	1 hora	1 hora y 35 minutos	Completo
Redacción del Capítulo 3	1 hora	1 hora y 20 minutos	Completo
Actualización del Capítulo 4	1 hora	1 hora y 38 minutos	Completo
Redacción del Capítulo 5	4 horas	5 horas y 45 minutos	Completo
Redacción del Capítulo 7	1 hora	50 minutos	Completo
Total	41 horas	54 horas y 21 minutos	13:13

Tabla 7.7: Tareas realizadas en el Sprint 6

El total de tiempo empleado para el Sprint 6 ha sido de **54 horas y 21 minutos**.

En el *Sprint Retrospective* se ha determinado que se ha podido llevar a cabo correctamente las tareas y no se ha manifestado ningún riesgo.

En el *Sprint Planning* se ha acordado que se va a terminar de redactar la memoria del TFG y a pulir la aplicación de cara a liberar la aplicación en Google Play.

7.1.8. Sprint 7 (30/05/2024 - 13/06/2024)

Se ha dedicado el Sprint 7 a terminar de redactar la memoria y a pulir la aplicación de cara a ser liberada a un entorno de producción.

En la Tabla 7.8 se puede ver las tareas desarrolladas en el Sprint 7 con los tiempos empleados para realizar dichas tareas.

Tareas	Tiempo empleado
Corrección de fallos de la aplicación e implementación de las propuestas	1 hora y 22 minutos
Revisión Capítulo 1	1 hora y 15 minutos
Revisión Capítulo 2	1 hora y 42 minutos
Redacción Capítulo 3	2 horas y 39 minutos
Redacción Capítulo 5	5 horas y 10 minutos
Redacción Capítulo 6	4 horas y 14 minutos
Redacción Capítulo 7	2 horas y 24 minutos
Redacción Capítulo 8	2 horas y 21 minutos
Redacción Anexo A	3 horas y 35 minutos
Revisión del código de la aplicación	50 minutos
Primera revisión de la memoria completa	1 hora y 30 minutos
Total	27 horas y 2 minutos

Tabla 7.8: Tareas realizadas en el Sprint 7

El tiempo empleado para el Sprint 7 ha sido **27 horas y 2 minutos**.

En el *Sprint Retrospective* se ha determinado que se ha completado con éxito las tareas previstas para este Sprint. Se ha manifestado el riesgo de coincidencia con la asignatura del grado Evaluación y Rendimiento de Sistemas Software, debido a que durante el desarrollo de este sprint se desarrolló el examen ordinario de dicha asignatura.

No se ha realizado *Sprint Planning* debido a ser el último sprint previsto y haber finalizado todas las tareas. No se requiere, por tanto, de ningún sprint adicional.

7.2. Resumen del desarrollo del proyecto

Una vez finalizado el proyecto, se resume en esta sección los riesgos que se han presentado a lo largo del desarrollo del proyecto, el tiempo empleado, los costes y un análisis de los costes que pueden venir en un futuro.

7.2.1. Riesgos

En el desarrollo del proyecto, se han presentado los siguientes riesgos de los enumerados en el análisis inicial de riesgos:

1. **RK11: Coincidencia con la asignatura.** En algunos sprints no se ha podido dedicar más tiempo o el tiempo planificado del sprint debido a actividades de dicha asignatura.
2. **RK09: Uso de datos abiertos.** Hasta el Sprint 3 no se pudo completar la implementación del RF01 debido a que los datos abiertos no estaban actualizados.
3. **RK02: Desconocimiento de la tecnología a emplear.** En algunos caso, no se controlaba bien las tecnologías necesarias para implementar una funcionalidad y se ha requerido más tiempo para completar una tarea.

7.2.2. Calendarización y tiempo empleado

Se ha seguido la calendarización establecida en el Sección 2.3 y no ha sido necesario emplear los dos sprints extras.

El tiempo previsto de desarrollo de proyecto inicial era de 300 horas. El proyecto se ha desarrollado en **308 horas y 6 minutos**. Esto supone 8 horas y 6 minutos más que el tiempo previsto.

7.3. Costes

En cuanto a los costes, la duración del desarrollo del proyecto ha sido 4 meses. Por tanto, podemos recalcular los costes teniendo en cuenta las horas dedicadas a este proyecto y la duración total de desarrollo.

7.3.1. Coste simulado final

El coste final simulado experimenta una variación muy pequeña con el planificado debido a que dicho precio ya incluía ciertos costes a 4 meses. Esto se puede ver en la Tabla 7.9.

Concepto	Precio Unitario	Cantidad	Total
Sueldo Full Stack	1.785,71€/mes	4 meses	7.142,84€
Sueldo Product Owner	2.857,14€/mes	4 meses	11.428,56€
Sueldo Scrum Master	2.571,42€/mes	4 meses	10.285,68€
Equipo PC	27,25€/mes	4 meses	109€
Nokia 6.1	4,13€/mes	4 meses	16,5€
Samsung Galaxy A03	6,25€/mes	4 meses	25€
Samsung Galaxy A54	20,8€/mes	4 meses	83,2€
Complemento teletrabajo	51€/mes	4 meses	204€
Subscripción Justinmind	8,53€/mes	4 meses	34,12€
Subscripción Visual Paradigm	31,99€/mes	4 meses	127,96€
Subscripción Microsoft Teams	3,66€/mes	4 meses	14,64€
Tasa Google Play			23,81€
Total			29495,31€
+25 % previsión de riesgos			36869,14€

Tabla 7.9: Coste simulado final

7.3.2. Coste real final

Teniendo en cuenta que un portátil suele gastar entre 200 y 250 W [39]. Realizamos la siguiente operación (tomando el máximo del intervalo):

$$kWh = \frac{308 \cdot 250}{1000} = 77$$

En cuanto a los móviles, teniendo en cuenta que usamos 3 dispositivos y que cada dispositivo consume 45Wh [111], realizamos la misma operación que en el caso anterior:

$$kWh = \frac{3 \cdot 45Wh \cdot 308}{1000} = 41,58$$

Por tanto, los kWh consumidos han sido 118,58 kWh. Podemos ver el coste real final en la Tabla 7.10

Concepto	Precio Unitario	Cantidad	Total
Electricidad	0,149842 €/kWh	118,58	17,77€
Tasa Google Play			23,81€
Equipo PC	27,25€/mes	4 meses	109€
Nokia 6.1	4,13€/mes	4 meses	16,5€
Samsung Galaxy A03	6,25€/mes	4 meses	25€
Samsung Galaxy A54	20,8€/mes	4 meses	83,2€
Total			275,28€

Tabla 7.10: Coste real final

7.3.3. Evaluación de Costes Futuros

La aplicación tiene como servicio de backend a Firebase, un servicio de Google cuyo coste es gratuito hasta que se alcanzan unos límites fijados por Google. A partir de ese momento, se tendría que contratar el plan de prepago que dispone Firebase.

En caso de que nuestra aplicación fuese un éxito y muchos usuarios quisieran descargarla y probarla, podríamos llegar a esta situación y que nuestra aplicación proporcione pérdidas. Esta situación se podría dar, sobre todo, en temporada estival, que es cuando se prevé que la aplicación pueda tener un mayor número de descargas.

Utilizando la herramienta calculadora de Firebase para el Plan Blaze [22], se estima el posible incremento de costes en Firebase:

- Realtime Database
 - Incremento de almacenamiento a 20 GB: 95\$/mes
 - Incremento a 195 GB de transferencia: 185\$/mes
- Authentication: incremento a 110.000 usuarios activos por mes: 276\$/mes.
- Cloud Storage
 - Incremento a 180 GB de almacenamiento: 4,95\$/mes.
 - Incremento a 60 GB de transferencia: 3,60\$/mes.

El coste estimado por mes sería de 564\$/mes, que en euros serían 523€.

Por lo tanto, para que el producto sea rentable, modificaría la aplicación para que utilice el servicio de anuncios de Google y muestre publicidad a los usuarios. También incluiría la opción de “Compras en la aplicación”, ofreciendo un pago anual para eliminar los anuncios y algunas de las nuevas características podrían pasar a ser de pago mediante un modelo de suscripción. En resumen, adoptaríamos un modelo de negocio **freemium**.

Capítulo 8

Conclusiones

Después de haber finalizado la implementación del proyecto, se puede concluir que he sido capaz de implementar la aplicación **AquaCyl** con éxito. Durante el proyecto, se ha utilizado algunos de los conceptos vistos en las asignaturas del grado, concretamente en las de la mención de Ingeniería de Software, así como en las habilidades adquiridas en la asignatura de Practicas en Empresa y en mi experiencia laboral.

En cuanto a los objetivos de desarrollo:

- Se ha conseguido implementar con éxito la consulta del estado de las zonas de baño de Castilla y León.
- Se ha conseguido implementar la consulta de detalles de una zona de baño.
- Se ha conseguido implementar el hecho de guardar las zonas como favoritos y permitir consultar dicho listado.
- Se ha conseguido que el usuario sea capaz de realizar comentarios en las zonas de baño.

Además, se ha añadido características adicionales como el hecho de poder recibir notificaciones, cambiar el idioma de la aplicación, añadir fotografías de la zona de baño, consultar la previsión meteorológica, entre otras.

En cuanto a los objetivos personales:

- He adquirido un mayor conocimiento de la tecnología Flutter, iniciando yo mismo el proyecto desde cero.
- He utilizado nuevos servicios de Firebase que nunca había utilizado como Firebase Analytics, Firebase Crashlytics, Firebase Storage o Firebase Authentication.
- He aplicado en el desarrollo de proyecto un marco de trabajo ágil basado en Scrum, como se puede ver en el Capítulo 7.

- He adquirido una mayor habilidad en cuanto a la experiencia de usuario y el diseño de aplicaciones, aplicando por norma general en la aplicación, las líneas de diseño basadas en Material Design 3 y habiendo hecho el cambio de Material Design 2 a Material Design 3, con los cambios de diseño que ha supuesto.
- He conseguido publicar en Google Play la aplicación y he aprendido a utilizar la consola de Google Play aprendiendo a usar los canales de pruebas internas, los canales de pruebas cerradas y el proceso de paso a producción, al igual que he aprendido a completar y a reportar toda la información que requiere Google para proceder a la publicación de la aplicación.

8.1. Líneas de trabajo futuras

Algunas de las posibles líneas de trabajo futuras a corto y medio-largo plazo para la aplicación son:

1. **Despliegue de aplicación en IOS:** La aplicación está disponible actualmente para los usuarios que disponen de un dispositivo Android. La idea sería adaptar la aplicación para que pueda ser ejecutado en IOS. El proceso sería más sencillo debido a la utilización de Flutter y nos tendríamos que preocupar únicamente de la integración y adaptación de la aplicación ya existente con el sistema operativo de Apple.
2. **Desarrollo de un backend para la aplicación:** La idea detrás del desarrollo de backend es reducir el número de tareas que se está realizando ahora mismo desde la aplicación y que éstas pasen a realizarse desde el backend. Esto nos permitiría que la aplicación pasara a tener una única API en lugar de las 3 que presenta ahora mismo, reducir la dependencia con Firebase y se abriría la posibilidad de ofrecer nuevas funcionalidades en la aplicación.
3. **Permitir que otras entidades de Castilla y León añadan sus zonas de baño:** Actualmente solo se están mostrando las zonas de baño que se proporcionan a través de Datos Abiertos por la Junta de Castilla y León. La idea a futuro es que otras administraciones puedan mostrar sus zonas de baño en la aplicación. El primer paso en este caso sería eliminar la integración con Datos Abiertos y que estas zonas de baño se pasen a almacenar en una base de datos propia. Para administrar las zonas de baño, se crearía unas cuentas especiales para las administraciones o instituciones que les permitiría añadir y administrar sus zonas de baño en la aplicación.
4. **Única aplicación para los roles de administración y usuario:** Al introducir el rol de administrador en la aplicación, la aplicación se debería comportar de forma diferente en función del rol. La vista para el rol de usuario se mantendría igual que la que se muestra actualmente y la vista para el rol de administrador sería totalmente diferente y quedaría pendiente por definir la estructura y las vistas de esta última.
5. **Pantalla de alertas:** Se introduciría en la aplicación una nueva pantalla en las que el usuario pueda ver las alertas enviadas por los administradores de sus zonas de baño favoritas. Dicha alerta se mostraría también como una notificación.

Bibliografía

- [1] AEMET. El avance implacable del verano en españa: análisis sobre su progresivo alargamiento. <https://aemetblog.es/2023/06/24/el-avance-implacable-del-verano-en-espana-analisis-sobre-su-progresivo-alargamiento/>, junio 2023. Acceso: 2024-02-25.
- [2] Victor Ahmad. Build a flutter app using mvvm. <https://medium.com/@Victor.Ahmad/implementing-model-view-viewmodel-mvvm-in-a-flutter-app-a-step-by-step-guide-92b05e6e8192>. Acceso: 2024-03-20.
- [3] Atlassian. Jira. <https://www.atlassian.com/software/jira>. Acceso: 2024-03-13.
- [4] Atlassian. Introducing a new jira for all teams. <https://community.atlassian.com/t5/Jira-articles/Introducing-a-new-Jira-for-all-teams/ba-p/2683613>. Acceso: 2024-05-08.
- [5] Atlassian. Sourcetree. <https://www.atlassian.com/software/sourcetree>. Acceso: 2024-03-13.
- [6] Back4app. Firebase storage: qué es y cómo funciona. <https://blog.back4app.com/es/que-es-firebase-storage-y-como-funciona/>. Acceso: 2024-04-03.
- [7] Jaime Alberto Téllez Bohórquez. Flutter — estructura de proyecto. <https://medium.com/@jaimetellezb/flutter-estructura-de-proyecto-96ae96becf33>. Acceso: 2024-05-15.
- [8] Sandra Camero. Arquitecturas de software. mvc y mvvm. <https://www.abatic.es/arquitecturas-de-software-mvc-y-mvvm/>. Acceso: 2024-04-01.
- [9] Manuel Perez Cardona. Firebase, qué es y para qué sirve la plataforma de google. <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>. Acceso: 2024-04-02.
- [10] Maria Coppola. Desarrollador full stack: qué es, funciones y salario promedio. <https://blog.hubspot.es/website/que-es-desarrollador-full-stack>. Acceso: 2024-03-04.
- [11] María Eugenia Coppola. Qué es google analytics, para qué sirve y cómo utilizarlo. <https://blog.hubspot.es/marketing/google-analytics>. Acceso: 2024-05-27.

- [12] Microsoft Corporation. Modelo-vista-modelo de vista (mvvm). <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>. Acceso: 2024-03-20.
- [13] Matias de Andrea. Eligiendo dependencias (packages) en flutter. <https://medium.com/flutter-espa%C3%B1a/eligiendo-dependencias-packages-en-flutter-8c8bf2136c4b>. Acceso: 2024-04-02.
- [14] Junta de Castilla y León. Zonas de agua de baño | ciudadanos. <https://www.saludcastillayleon.es/es/zonasdebaño>. Acceso: 2024-02-25.
- [15] Universidad de Sevilla. Requisitos para un sistema de información. <http://www.lsi.us.es/docencia/get.php?id=6845>. Acceso: 2024-03-06.
- [16] Escuela Ingeniería Informática. Universidad de Valladolid. Calendario actividades 2º cuatrimestre 3º grado curso 2023-2024. https://www.inf.uva.es/wp-content/uploads/2024/02/CAD_23-24_3Curso_2C.pdf. Acceso: 2024-03-03.
- [17] Escuela Ingeniería Informática. Universidad de Valladolid. Calendario exámenes 3º grado curso 2023-2024. https://www.inf.uva.es/wp-content/uploads/2023/09/CalendarioExámenes-2324_Grado_Tercero_IS.pdf. Acceso: 2024-03-03.
- [18] Universidad de Valladolid. Calendario académico curso 2023-2024. https://www.uva.es/export/sites/uva/2.estudios/_documentos/Calendario-23-24.pdf. Acceso: 2024-03-03.
- [19] Boletín Oficial del Estado. Ley 10/2021, de 9 de julio, de trabajo a distancia. <https://www.boe.es/boe/dias/2021/07/10/pdfs/BOE-A-2021-11472.pdf>. Acceso: 2024-03-13.
- [20] Boletín Oficial del Estado. Resolución de 13 de julio de 2023, de la dirección general de trabajo, por la que se registra y publica el xviii convenio colectivo estatal de empresas de consultoría, tecnologías de la información y estudios de mercado y de la opinión pública. <https://www.boe.es/boe/dias/2023/07/26/pdfs/BOE-A-2023-17238.pdf>. Acceso: 2024-03-13.
- [21] Ecured. Visual paradigm. https://www.ecured.cu/Visual_Paradigm. Acceso: 2024-04-21.
- [22] Firebase. Calculadora del plan blaze. <https://firebase.google.com/pricing?authuser=0&hl=es-419#blaze-calculator>. Acceso: 2024-06-10.
- [23] Firebase. Cloud storage para firebase. <https://firebase.google.com/docs/storage?hl=es-419>. Acceso: 2024-04-03.
- [24] Firebase. Firebase authentication. <https://firebase.google.com/docs/auth?hl=es-419>. Acceso: 2024-04-03.
- [25] Firebase. Firebase cloud messaging. <https://firebase.google.com/docs/cloud-messaging?hl=es-419>. Acceso: 2024-05-08.
- [26] Firebase. Firebase crashlytics. <https://firebase.google.com/docs/crashlytics?hl=es-419>. Acceso: 2024-04-03.

- [27] Firebase. Firebase hosting. <https://firebase.google.com/docs/hosting?hl=es-419>. Acceso: 2024-05-08.
- [28] Firebase. Firebase realtime database. <https://firebase.google.com/docs/database?hl=es-419>. Acceso: 2024-05-08.
- [29] Glassdoor. Sueldos para el puesto de full stack developer en españa. https://www.glassdoor.es/Sueldos/full-stack-developer-sueldo-SRCH_K00,20.htm. Acceso: 2024-03-04.
- [30] Glassdoor. Sueldos para el puesto de product owner en españa. https://www.glassdoor.es/Sueldos/product-owner-sueldo-SRCH_K00,13.htm. Acceso: 2024-03-04.
- [31] Glassdoor. Sueldos para el puesto de scrum master en españa. https://www.glassdoor.es/Sueldos/scrum-master-sueldo-SRCH_K00,12.htm. Acceso: 2024-03-04.
- [32] Google. Android studio. <https://developer.android.com/studio?hl=es-419>. Acceso: 2024-03-13.
- [33] Google. Firebase. <https://firebase.google.com/solutions?authuser=0&hl=es-419>. Acceso: 2024-03-13.
- [34] Google. In app update. <https://developer.android.com/guide/playcore/in-app-updates?hl=es-419>. Acceso: 2024-04-02.
- [35] Herocoders. Clockwork. <https://docs.herocoders.com/clockwork/>. Acceso: 2024-05-08.
- [36] Hubspot. Brand kit generator. <https://www.hubspot.com/brand-kit-generator>. Acceso: 2024-05-28.
- [37] IONOS. ¿qué es jira? todo lo que necesitas saber sobre el software de gestión de proyectos. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-jira/>. Acceso: 2024-05-08.
- [38] iubenda. ¿qué es un eula y cómo puedes generar uno? <https://www.iubenda.com/es/help/43961-que-es-un-eula-y-como-puedes-generar-uno>. Acceso: 2024-05-15.
- [39] Javier Jiménez. Esto es lo que consume tu ordenador encendido. <https://www.redeszone.net/noticias/power/consumo-ordenador-encendido/>. Acceso: 2024-06-10.
- [40] Justinmind. Pricing. <https://www.justinmind.com/pricing>. Acceso: 2024-03-13.
- [41] Redacción KeepCoding. Qué es firebase. <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>. Acceso: 2024-04-02.
- [42] Luis Llamas. Sourcetree. <https://www.luisllamas.es/sourcetree/>. Acceso: 2024-05-08.
- [43] Pablo Londoño. Qué es bootstrap, para qué sirve y cómo funciona. <https://blog.hubspot.es/website/que-es-bootstrap>. Acceso: 2024-05-27.

- [44] Back Market. Nokia 6.1. <https://www.backmarket.es/es-es/p/nokia-nokia-61-64-gb-negro-libre/c452930d-cad3-4e6b-be81-448bdfc62020>. Acceso: 2024-03-13.
- [45] Mateo Martin. Mvc vs mvvm: diferencia entre ellos. <https://www.guru99.com/es/mvc-vs-mvvm.html#advantages-of-mvvm>. Acceso: 2024-04-01.
- [46] Aurora Martínez. Definición de riesgo. <https://conceptodefinicion.de/riesgo/>. Acceso: 2024-02-28.
- [47] Muhammad Mateen. How to set up interceptors for http calls in flutter app. <https://medium.com/upday-devs/how-to-set-up-interceptors-for-http-calls-in-flutter-app-b0560985b33>. Acceso: 2024-05-15.
- [48] Microsoft. Microsoft teams. <https://www.microsoft.com/es-es/microsoft-teams/group-chat-software/>. Acceso: 2024-05-08.
- [49] Microsoft. Microsoft teams pricing. <https://www.microsoft.com/es/microsoft-teams/compare-microsoft-teams-business-options?market=bz>. Acceso: 2024-03-13.
- [50] Assembler Institute of Technology. ¿qué es postman? características y ventajas. <https://assemblerinstitute.com/blog/que-es-postman/>. Acceso: 2024-04-03.
- [51] Orange. ¿qué son las licencias de ‘software’ y qué tipos hay? <https://blog.orange.es/consejos-y-trucos/que-son-las-licencias-de-software-y-que-tipos-hay/>. Acceso: 2024-05-15.
- [52] José Manuel Montero Ortega. La arquitectura mvvm y sus componentes. <https://openwebinars.net/blog/la-arquitectura-mvvm-y-sus-componentes/>. Acceso: 2024-03-20.
- [53] Overleaf. Features. <https://www.overleaf.com/about/features-overview>. Acceso: 2024-05-08.
- [54] Visual Paradigm. Pricing. <https://www.visual-paradigm.com/shop/vp.jsp?license=monthly>. Acceso: 2024-03-13.
- [55] pub.dev. Cached network image. https://pub.dev/packages/cached_network_image. Acceso: 2024-04-02.
- [56] pub.dev. Camera. <https://pub.dev/packages/camera>. Acceso: 2024-05-07.
- [57] pub.dev. Connectivity plus. https://pub.dev/packages/connectivity_plus. Acceso: 2024-05-27.
- [58] pub.dev. Device info plus. https://pub.dev/packages/device_info_plus. Acceso: 2024-05-07.
- [59] pub.dev. Dio. <https://pub.dev/packages/dio>. Acceso: 2024-04-02.
- [60] pub.dev. External app launcher. https://pub.dev/packages/external_app_launcher. Acceso: 2024-04-02.
- [61] pub.dev. File picker. https://pub.dev/packages/file_picker. Acceso: 2024-05-07.

- [62] pub.dev. Firebase analytics. https://pub.dev/packages/firebase_analytics. Acceso: 2024-05-27.
- [63] pub.dev. Firebase authenticator. https://pub.dev/packages/firebase_auth. Acceso: 2024-05-07.
- [64] pub.dev. Firebase core. https://pub.dev/packages/firebase_core. Acceso: 2024-03-13.
- [65] pub.dev. Firebase crashlytics. https://pub.dev/packages/firebase_crashlytics. Acceso: 2024-03-13.
- [66] pub.dev. Firebase database. https://pub.dev/packages/firebase_database. Acceso: 2024-05-07.
- [67] pub.dev. Firebase messaging. https://pub.dev/packages/firebase_messaging. Acceso: 2024-05-08.
- [68] pub.dev. Firebase storage. https://pub.dev/packages/firebase_storage. Acceso: 2024-04-02.
- [69] pub.dev. Flutter email sender. https://pub.dev/packages/flutter_email_sender. Acceso: 2024-05-07.
- [70] pub.dev. Flutter local notifications. https://pub.dev/packages/flutter_local_notifications. Acceso: 2024-05-08.
- [71] pub.dev. Flutter slidable. https://pub.dev/packages/flutter_slidable. Acceso: 2024-05-27.
- [72] pub.dev. Get. <https://pub.dev/packages/get>. Acceso: 2024-03-13.
- [73] pub.dev. Google sign in. https://pub.dev/packages/google_sign_in. Acceso: 2024-05-07.
- [74] pub.dev. Image picker. https://pub.dev/packages/image_picker. Acceso: 2024-05-07.
- [75] pub.dev. In app update. https://pub.dev/packages/in_app_update. Acceso: 2024-04-02.
- [76] pub.dev. Intl. <https://pub.dev/packages/intl>. Acceso: 2024-05-14.
- [77] pub.dev. Package info plus. https://pub.dev/packages/package_info_plus. Acceso: 2024-05-07.
- [78] pub.dev. Path. <https://pub.dev/packages/path>. Acceso: 2024-05-14.
- [79] pub.dev. Path. https://pub.dev/packages/flutter_rating_bar. Acceso: 2024-05-14.
- [80] pub.dev. Permission handler. https://pub.dev/packages/permission_handler/. Acceso: 2024-05-07.

- [81] pub.dev. Shared preferences. https://pub.dev/packages/shared_preferences. Acceso: 2024-05-07.
- [82] pub.dev. Shimmer. <https://pub.dev/packages/shimmer>. Acceso: 2024-05-07.
- [83] pub.dev. Url launcher. https://pub.dev/packages/url_launcher. Acceso: 2024-04-02.
- [84] pub.dev. Webview flutter. https://pub.dev/packages/webview_flutter. Acceso: 2024-04-02.
- [85] Eduardo Rodriguez. Mvvm – qué es y como funciona. <https://inmediatum.com/blog/ingenieria/mvvm-que-es-y-como-funciona/>. Acceso: 2024-04-01.
- [86] Samsung. Samsung galaxy a03. <https://www.samsung.com/es/smartphones/galaxy-a/galaxy-a03-black-64gb-sm-a035gzkgeub/?msockid=34acc777ceb3680e11d8d32bcf0a69d7>. Acceso: 2024-06-09.
- [87] Samsung. Samsung galaxy a54. <https://www.samsung.com/es/smartphones/galaxy-a/galaxy-a54-5g/buy/?msockid=34acc777ceb3680e11d8d32bcf0a69d7>. Acceso: 2024-06-09.
- [88] Scrum. La guía scrum. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>. Acceso: 2024-03-04.
- [89] Scrum. The scrum artifacts. <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts>. Acceso: 2024-03-04.
- [90] Scrum. The scrum events. <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events>. Acceso: 2024-03-04.
- [91] Scrum. Scrum framework 9.29.23. <https://scrumorg-website-prod.s3.amazonaws.com/drupal/inline-images/2023-09/scrum-framework-9.29.23.png>. Acceso: 2024-03-04.
- [92] Scrum. The scrum team. <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team>. Acceso: 2024-03-04.
- [93] Scrum. Scrum values. <https://s3.amazonaws.com/static.scrum.org/web/Scrum+Values.png>. Acceso: 2024-03-04.
- [94] Scrum. What is a definition of done? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-a-definition-of-done>. Acceso: 2024-03-04.
- [95] Scrum. What is a developer? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-developer>. Acceso: 2024-03-04.
- [96] Scrum. What is a product backlog? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-a-product-backlog>. Acceso: 2024-03-04.

- [97] Scrum. What is a product goal? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-a-product-goal>. Acceso: 2024-03-04.
- [98] Scrum. What is a product owner? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-product-owner>. Acceso: 2024-03-04.
- [99] Scrum. What is a scrum master? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team/what-is-a-scrum-master>. Acceso: 2024-03-04.
- [100] Scrum. What is a sprint? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint>. Acceso: 2024-03-04.
- [101] Scrum. What is a sprint backlog? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-a-sprint-backlog>. Acceso: 2024-03-04.
- [102] Scrum. What is a sprint goal? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-a-sprint-goal>. Acceso: 2024-03-04.
- [103] Scrum. What is a sprint retrospective? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint-retrospective>. Acceso: 2024-03-04.
- [104] Scrum. What is an increment? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-artifacts/what-is-an-increment>. Acceso: 2024-03-04.
- [105] Scrum. What is daily scrum? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-daily-scrum>. Acceso: 2024-03-04.
- [106] Scrum. What is scrum? <https://www.scrum.org/learning-series/what-is-scrum/>. Acceso: 2024-03-04.
- [107] Scrum. What is sprint planning? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-sprint-planning>. Acceso: 2024-03-04.
- [108] Scrum. What is sprint review? <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-events/what-is-a-sprint-review>. Acceso: 2024-03-04.
- [109] Guillem Hernandez Sola. ¿existe el sprint 0 en scrum? <https://www.scrum.org/resources/blog/existe-el-sprint-0-en-scrum>. Acceso: 2024-02-28.
- [110] tarifaluzhora.es. ¿cuánto cuesta el kilovatio hora de luz (kwh) en españa? <https://tarifaluzhora.es/info/precio-kwh>. Acceso: 2024-03-13.
- [111] Tecnodroid. Cuantos watts consume cargar un celular: descubre el gasto de energía. <https://tecnodroid.co/cuantos-watts-consume-cargar-un-celular>. Acceso: 2024-06-14.
- [112] Aures tic. ¿qué es flutter? <https://aurestic.es/que-es-flutter/>. Acceso: 2024-03-13.
- [113] Universidad de Valladolid. Proyecto docente del trabajo de fin de grado 2023-2024 (Mención Ingeniería de Software). https://apps.stic.uva.es/guias_docentes/uploads/2023/545/46976/1/Documento.pdf. Acceso: 2024-02-24.

- [114] Dominio Virtual. Portátil hp laptop 15-da0071ns. <https://www.dominiovirtual.es/ordenadores-portatiles/6379/3zu33earabe/portatil-hp-laptop-15-da0071ns-0193015050747.html>. Acceso: 2024-03-13.
- [115] wearetesters. Pruebas de aceptación de usuario (uat): qué son y por qué es importante hacerlas. <https://www.wearetesters.com/investigacion-ux/pruebas-aceptacion-usuario-uat-que-son/>. Acceso: 2024-05-15.
- [116] webdesigncusco. Características principales de firebase: La plataforma de desarrollo de aplicaciones en la nube. <https://webdesigncusco.com/caracteristicas-principales-de-firebase/>. Acceso: 2024-04-02.
- [117] Wikipedia. Android studio. https://es.wikipedia.org/wiki/Android_Studio. Acceso: 2024-04-21.
- [118] Wikipedia. Git. <https://es.wikipedia.org/wiki/Git>. Acceso: 2024-05-01.
- [119] Wikipedia. Justinmind. [https://en.wikipedia.org/wiki/Justinmind_\(software\)](https://en.wikipedia.org/wiki/Justinmind_(software)). Acceso: 2024-04-21.
- [120] Wikipedia. Material design. https://es.wikipedia.org/wiki/Material_Design. Acceso: 2024-04-21.
- [121] Wikipedia. Visual studio code. https://es.wikipedia.org/wiki/Visual_Studio_Code. Acceso: 2024-05-27.
- [122] Xataka. Qué es github y qué es lo que le ofrece a los desarrolladores. <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>. Acceso: 2024-05-01.
- [123] Ángel Arias. Aprende sobre la ingeniería del software: Reglas de negocio. <https://zoboko.com/text/65jdm2mv/aprende-sobre-la-ingenieria-del-software/50>. Acceso: 2024-03-06.
- [124] Ángel Arias. Aprende sobre la ingeniería del software: Requisitos no funcionales. <https://zoboko.com/text/65jdm2mv/aprende-sobre-la-ingenieria-del-software/49>. Acceso: 2024-03-06.

Apéndice A

Manuales

A.1. Manual de instalación

Para instalar la aplicación se requiere de un **teléfono o tablet Android** con versión **superior a Android 10** (Android 10 incluido).

Esta aplicación se encuentra disponible en Google Play, por lo que para poder instalarla, debe entrar en el enlace de descarga en Google Play indicada en el Apéndice B. Si ha entrado desde el ordenador, puede iniciar sesión con su cuenta de Google e indicarle a Google Play que instale la aplicación en el dispositivo que elija. Si ha entrado a dicho enlace desde su teléfono o tablet, se abrirá Google Play y podrá instalar la aplicación pulsando en el botón “Instalar”.

A.2. Manual de mantenimiento y de despliegue en Google Play

Para poder realizar mantenimiento de la aplicación o añadir nuevas características y mejorar otras, debe solicitar acceso al repositorio GitHub. Una vez concedido el permiso, podrá clonar el repositorio y subir los cambios realizados en él.

Para desplegar la aplicación en Google Play, hay que solicitar permiso al desarrollador de la aplicación para que le añada como usuario colaborador dentro de la cuenta de desarrollador de Google. Si se aprueba la solicitud, tendrá desde su cuenta acceso unicamente a la aplicación de AquaCyL y no podrá ver el resto de aplicaciones del desarrollador. Podrá desplegar nuevas versiones de la aplicación.

A.3. Manual de usuario

La primera vez que se accede a la aplicación, se muestra una pantalla como la que se puede ver en la Figura A.1. Las opciones de acceso se explican en la Sección A.3.1 de este manual. En caso de que tengamos previamente iniciada sesión, nos cargará directamente la pantalla Home de la aplicación (ver Figura A.5).

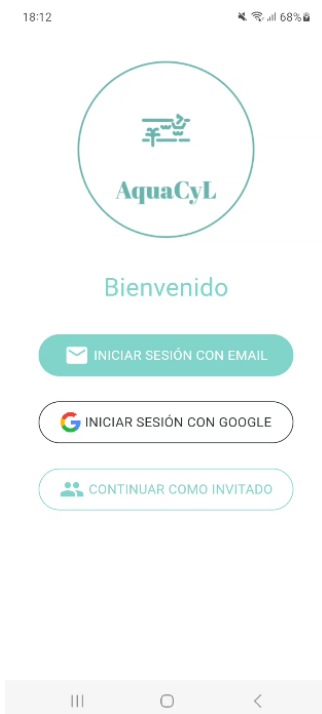


Figura A.1: Pantalla principal

A.3.1. Acceso a la aplicación

Tenemos tres opciones para acceder a la aplicación: Continuar como invitado, iniciar sesión con Google o iniciar sesión con email. A continuación, se detalla cada una de ellas.

1. **Continuar como invitado:** Al continuar como invitado, se está accediendo a funcionalidades limitadas de la aplicación. Habrá funciones que si que disfruten los usuarios registrados que no se puedan disfrutar con este modo. Al pulsar *Continuar como invitado*, la aplicación dirigirá a la pantalla Home (Figura A.5).
2. **Iniciar sesión con Google:** Esta opción permite iniciar sesión en la aplicación con las cuentas de Google disponibles en el dispositivo. Al pulsar *Iniciar sesión con Google*,

se mostrará las cuentas disponibles de Google en su dispositivo y se iniciará sesión con la que desee. Pasaremos a la pantalla Home (Figura A.5).

3. **Iniciar sesión con email:** Esta opción permite crear o iniciar sesión con un email y contraseña. Al pulsar *Iniciar sesión con email*, se mostrará la pantalla de iniciar sesión (Figura A.2) donde se podrá iniciar sesión. Además disponemos de la pantalla para registrarnos (Figura A.3) y para recuperar la contraseña en caso de habernos olvidado de ella (Figura A.4). Al iniciar sesión, se nos dirigirá a la pantalla Home (Figura A.5)

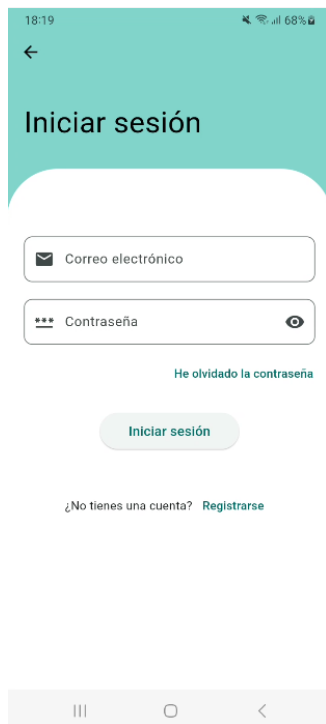


Figura A.2: Pantalla Iniciar Sesión



Figura A.3: Pantalla Registrarse



Figura A.4: He olvidado la contraseña

A.3.2. Pantalla Home

En la pantalla Home nos encontraremos un listado con las zonas de baño disponibles (Figura A.5). En esta pantalla nos encontramos varios elementos a mayores.

En la barra superior tenemos el nombre de la aplicación y dos iconos en la parte derecha. El primer icono nos muestra un popup para ordenar las zonas de baño (Figura A.7) y el segundo icono nos sirve para cerrar sesión (Figura A.6).

Disponemos de un botón flotante que nos va a permitir filtrar las zonas de baño por

provincia y por estado. En caso de que esté registrado, también nos permitirá filtrar por favorito (Figura A.8).

La barra inferior de navegación nos permitirá ir a la pantalla donde se muestran los favoritos (Figura A.12) y a los ajustes de la aplicación (Figura A.13).



Figura A.5: Pantalla Home

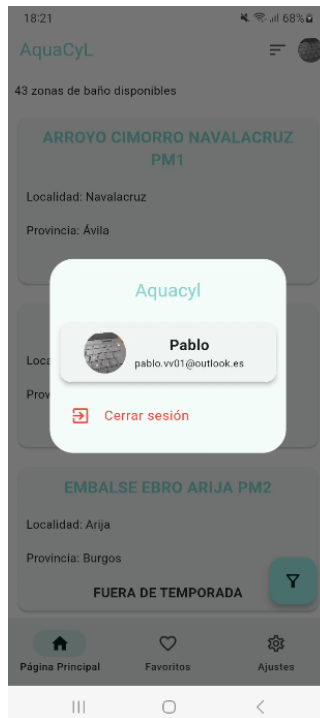


Figura A.6: Cerrar sesión

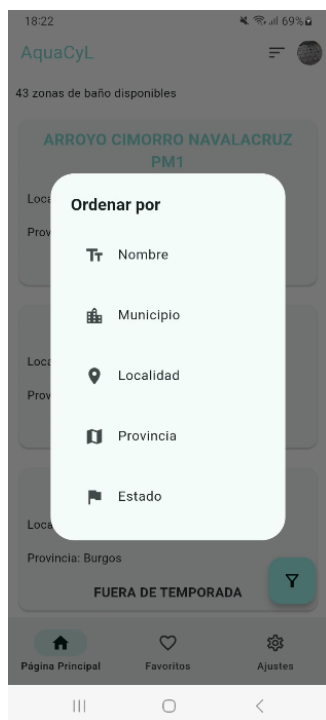


Figura A.7: Ordenar zonas de baño



Figura A.8: Filtrar zonas de baño

A.3.3. Pantalla de detalles

Al pulsar sobre una zona de baño, se nos mostrarán los detalles de dicha zona de baño, entre lo que se incluye: el estado, la puntuación de la zona de baño, una descripción de la localidad, las fotos de la zona de baño y la previsión meteorológica para dicha zona de baño. Al pulsar sobre cada elemento de la previsión del tiempo podemos ver los detalles de dicha previsión.

En la barra superior de la página nos encontramos con un botón para marcar una zona de baño como favorita.

Además se podrá acceder a los alojamientos (Figura A.10), a las direcciones y a los comentarios (en caso de estar registrado).

Los comentarios se pueden acceder pulsando sobre el botón flotante de comentarios. En el se podrá ver un listado de comentarios ordenados de más reciente a más antiguo (Figura A.11). En la parte superior se podrá pulsar para redactar un nuevo comentario.



Figura A.9: Pantalla de detalles



Figura A.10: Pantalla de alojamientos

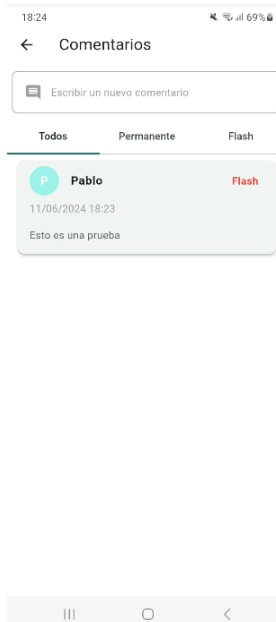


Figura A.11: Pantalla de comentarios

A.3.4. Pantalla de favoritos

En la pantalla de favoritos, podemos ver las zonas de baño favoritas del usuario. Esta pantalla solo se podrá ver si se ha iniciado sesión.

Por cada zona de baño, si desplazamos a la izquierda podemos eliminar dicha zona de baño y si desplazamos a la derecha podemos anclar la zona de baño. Las zonas de baño ancladas se muestran en la parte superior de la pantalla.

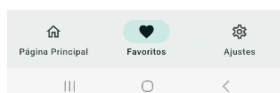


Figura A.12: Pantalla de favoritos

A.3.5. Ajustes

En la pantalla de ajustes se puede ver todas las opciones de los ajustes (Figura A.13). Si se está registrado como usuario, se podrá ver las opciones de la cuenta (Figura A.14). Para todos los usuarios, se podrá cambiar de idioma, de tema de la aplicación, buscar actualizaciones, enviar feedback (Figura A.15) y consultar los aspectos legales (Figura A.16).

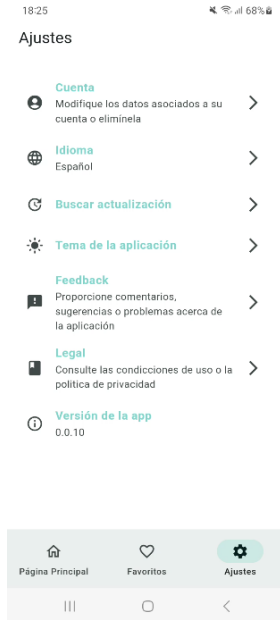


Figura A.13: Pantalla de ajustes

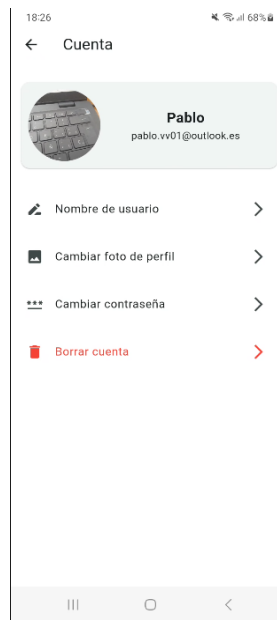


Figura A.14: Pantalla de cuenta

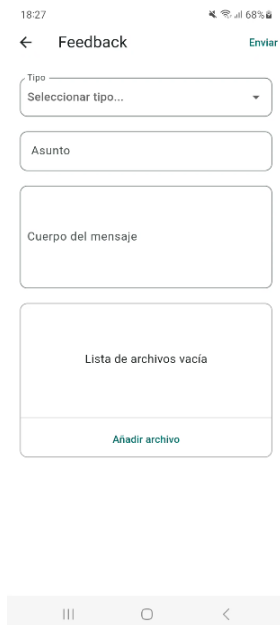


Figura A.15: Pantalla de feedback

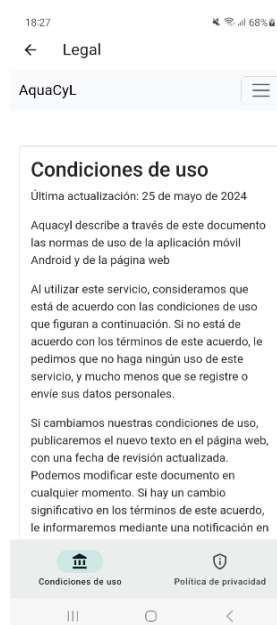


Figura A.16: Pantalla de aspectos legales

Apéndice B

Resumen de enlaces adicionales

Los enlaces útiles de interés en este Trabajo Fin de Grado son:

- Repositorio del código en Gitbub: https://github.com/pablovv01/TFG_AquaCyL.
En este repositorio, en la carpeta *app* podemos encontrar el código desarrollado, en la carpeta *models* se encuentran los diagramas que se encuentran en la memoria y en la carpeta *web* se encuentran los archivos html y css de la páginas de Política de Privacidad y de Cierre de Cuenta.
Sobre este repositorio se ha realizado el control de versiones.
- Repositorio de código en GitLab: https://gitlab.inf.uva.es/pabvare/tfg_aquacyl.
Este repositorio contiene una copia idéntica de lo que contiene el repositorio alojado en GitHub.
Se proporcionará acceso a los miembros del tribunal con sus cuentas de la Escuela.
- Enlace para descargar la aplicación AquaCyL en Google Play: https://play.google.com/store/apps/details?id=es.pablo.aquacyl.aqua_cyl
- Enlace a la página web de la aplicación: <https://aquacyl-d89d7.firebaseio.com/>