



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

**Find Your Project: Red social para incubar
proyectos de desarrollo de software**

Alumno: Gonzalo Calvo Sanz

Tutor: Yania Crespo González-Carvajal

A mi familia, por apoyarme tanto durante todos estos años

Agradecimientos

A mi padres, por estar siempre apoyándome en cada decisión que he tomado.

A mi hermano Daniel, por acompañarme durante toda la vida.

A mis amigos de la carrera, por hacer que estos años se hayan pasado mas rápido que ningún otro.

A mi novia Irene, por aguantarme y ser mi lugar seguro durante estos meses tan duros.

A Five Flames Mobile, por darme la oportunidad de comenzar mi vida laboral a su lado.

A mi tutora Yania, por todo el apoyo y la ayuda que me ha ofrecido, por sacar siempre tiempo cuando lo he necesitado, y por ser la mejor profesora que he tenido a lo largo de estos años.

Muchas gracias a todos

Resumen

El objetivo de este proyecto es desarrollar una aplicación para ayudar a la comunidad *Open Source* en la búsqueda de equipos de trabajo para proyectos incipientes de desarrollo de software. Además, una vez se tenga el equipo inicial, la aplicación se encargará de crear un repositorio en *Github* con los miembros de este equipo como colaboradores para poder comenzar a trabajar cuando se desee y redirigir a los usuarios interesados al proyecto ya creado.

Para conseguir el objetivo se ha ejecutado un proyecto de desarrollo de software en el que se ha desarrollado una app para el sistema operativo Android, utilizando Kotlin como lenguaje de programación. También se ha utilizado *Scrum* como marco de trabajo ágil adaptándolo a las características del proyecto y su contexto. Se ha intentado seguir las mejores practicas para conseguir un producto de alta calidad.

Abstract

The goal of this project is to develop an application to help the Open Source Community in search of work teams for fledgling software development projects. Also, when the initial team is completed, the application will take care of creating a Github repository with team members as collaborators, to be able to start working whenever they want and redirect other interested users to it.

To achieve the goal, a software project has been executed, in which an Android app has been developed. Scrum has also been used as agile framework, adapting it to project features and context. It has been tried to follow best practices to achieve a high quality project.

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. Estudio de alternativas	2
1.3.1. Twago	2
1.3.2. Fiverr	3
1.3.3. Malt	3
1.3.4. Conclusiones obtenidas	4
1.4. Objetivos	4
1.5. Estructura de la memoria	4
2. Requisitos y Planificación	7
2.1. Scrum	7

2.1.1. Roles en Scrum	7
2.1.2. Artefactos	8
2.1.3. Eventos	8
2.2. Scrum en este proyecto	9
2.3. Planificación Inicial	9
2.4. Product Backlog inicial	10
2.5. Product Backlog final	13
2.6. Plan de riesgos	15
2.7. Plan de presupuestos	16
2.7.1. Presupuesto simulado	21
2.7.2. Presupuesto real	22
3. Tecnologías utilizadas	25
3.1. GitLab	25
3.1.1. Git	25
3.1.2. GitLab Issue Board	25
3.2. Android	28
3.2.1. Material Design	28
3.2.2. Fragments	28
3.2.3. Navigation	29
3.3. Firebase	29
3.3.1. Realtime Database	29
3.3.2. Authentication	29
3.4. Retrofit	31
3.5. Herramientas para el análisis, diseño y documentación	31
3.5.1. Astah Professional	31
3.5.2. Figma	31

3.5.3. Overleaf	31
3.6. Herramientas para la comunicación	31
3.6.1. Microsoft Teams	31
4. Análisis	33
4.1. Requisitos funcionales como historias de usuario	33
4.2. Requisitos de información como historias de usuario	33
4.3. Requisitos no funcionales como historias de usuario	33
4.4. Restricciones como historias de usuario	34
4.5. Modelo de dominio Inicial	34
4.6. Modelo de proceso de Negocio	34
5. Diseño	39
5.1. Decisiones de diseño	39
5.2. Arquitectura MVVM	39
5.3. Patrones de diseño utilizados	40
5.3.1. Patrón Observador	40
5.3.2. Método Factoría	41
5.4. Diseño de la interfaz de usuario	42
5.4.1. Decisiones de diseño de la interfaz de usuario	42
5.4.2. Mockups	43
5.5. Diseño Arquitectónico	46
5.6. Diseño de la comunicación entre objetos	51
5.7. Diagrama de despliegue	52
5.8. Diseño de la Base de Datos	52
5.8.1. Usuarios	52
5.8.2. Proyectos	52

5.8.3. Traducción JSON a POJO por Firebase	54
5.9. Retrofit	55
6. Implementación y pruebas	59
6.1. Implementación	59
6.1.1. Filtrado por habilidad	59
6.1.2. Problemas en la implementación	59
6.2. Pruebas	62
6.2.1. Tipos de pruebas	62
6.2.2. Casos de prueba	62
6.2.3. Resultados de las pruebas	72
6.3. Copyright	72
7. Seguimiento del proyecto	75
7.1. Introducción	75
7.2. Seguimiento de los sprints realizados	76
7.2.1. Sprint 0 (01/02/22 - 17/02/22)	76
7.2.2. Sprint 1 (17/02/22 - 03/03/22)	76
7.2.3. Sprint 2 (03/03/22 - 17/03/22)	77
7.2.4. Sprint 3 (17/03/22 - 31/03/22)	79
7.2.5. Sprint 4 (31/03/22 - 14/03/22)	79
7.2.6. Sprint 5 (21/04/22 - 05/05/22)	81
7.2.7. Sprint 6 (05/04/22 - 19/05/22)	82
7.2.8. Sprint 7 (19/05/22 - 02/06/22)	82
7.3. Resumen de la ejecución del proyecto	83
7.3.1. Calendarización	84
7.4. Tiempo empleado	84

7.5. Costes	84
7.5.1. Coste simulado final	84
7.5.2. Coste real final	85
8. Conclusiones	87
8.1. Líneas de trabajo futuras	87
Bibliografía	89
A. Manuales	93
A.1. Manual de instalación	93
A.2. Manual de mantenimiento	93
A.3. Manual de usuario	94
A.3.1. Usuarios que quieran crear un proyecto	94
A.3.2. Usuarios que quieran unirse a un proyecto	95
B. Resumen de enlaces adicionales	101

Lista de Figuras

3.1. Issue Board 1	27
3.2. Issue Board 2	27
3.3. Ejemplo Gráfico de Navegación	30
4.1. Modelo de Dominio Inicial	36
4.2. Modelo de Proceso de Negocio	37
5.1. Relaciones entre partes en el patrón arquitectónico MVVM	40
5.2. Estructura Patrón Observador	41
5.3. Estructura Método Factoría	42
5.4. Vista LogIn	44
5.5. Vista Proyectos	44
5.6. Vista Nuevo Proyecto	44
5.7. Vista Detalles del Proyecto	44
5.8. Vista Perfil	45
5.9. Vista Editar Perfil	45
5.10. Vista solicitudes	45
5.11. Vista Mis Proyectos	45
5.12. Arquitectura general de la aplicación	47
5.13. <i>Decomposition&Uses style</i> del paquete <i>entities</i>	48

5.14. <i>Decomposition&Uses style</i> del paquete <i>interfaces</i>	48
5.15. <i>Decomposition&Uses style</i> del paquete <i>ui</i>	49
5.16. <i>Decomposition&Uses style</i> del paquete <i>api</i>	49
5.17. <i>Decomposition&Uses style</i> del paquete <i>viewModels</i>	50
5.18. <i>Decomposition&Uses style</i> del paquete <i>repositories</i>	50
5.19. Diagrama de secuencia principal de la EP1H7	53
5.20. Diagrama de secuencia del patrón Observador	54
5.21. Diagrama de secuencia del patrón Observador (Continuación)	55
5.22. Diagrama de despliegue	56
5.23. Estructura Lista de Proyectos en la BD	57
5.24. Estructura Lista de Usuarios en la BD	57
6.1. Ticket enviado a Github	61
6.2. Ticket de respuesta de Github	62
A.1. Vista LogIn	96
A.2. Vista Proyectos	96
A.3. Vista Nuevo Proyecto	96
A.4. Vista Detalles del Proyecto	96
A.5. Vista Perfil	97
A.6. Vista Editar Perfil	97
A.7. Vista mis solicitudes	97
A.8. Vista Mis Proyectos	97
A.9. Vista Mis proyectos iniciados	98
A.10. Vista solicitudes	98
A.11. Vista Detalles Proyecto completo	98
A.12. Vista Detalles Repositorio Creado	98

LISTA DE FIGURAS

A.13.Vista Detalles Iniciado	99
A.14.Vista Enviar Solicitud	99
A.15.Vista Detalles Pendiente	99
A.16.Vista Detalles Aceptado	99

Lista de Tablas

2.2. <i>Product Backlog</i> inicial	10
2.3. Historias de usuario	11
2.1. Planificación Inicial de Sprints	12
2.4. Prioridad Historias de usuario	13
2.5. Historias de usuario Final	14
2.6. Prioridad Historias de usuario Final	15
2.7. Riesgo R01. Ausencia temporal del estudiante por enfermedad	16
2.8. Riesgo R02. Fallos técnicos en el equipo de trabajo del estudiante	17
2.9. Riesgo R03. Ausencia temporal de la tutora por enfermedad	18
2.10. Riesgo R04. Confinamiento domiciliario debido a COVID-19	18
2.11. Riesgo R05. Mala estimación de tiempo y trabajo de algunas actividades	19
2.12. Riesgo R06. Desconocimiento de la arquitectura a utilizar	19
2.13. Riesgo R07. Menor tiempo empleado que el previsto debido a otros asuntos académicos	20
2.14. Riesgo R09. Cambio en los requisitos del proyecto	20
2.15. Riesgo R09. Necesidad de aprendizaje de nuevas herramientas debido a actualizaciones en las tecnologías utilizadas	21
2.16. Presupuesto Simulado	23
2.17. Presupuesto Real	23
4.1. Requisitos de información	34

4.2. Requisitos no funcionales	34
4.3. Restricciones	35
6.1. CP-01. Registro	63
6.2. CP-02. Inicio de sesión	63
6.3. CP-03. Cerrar sesión	64
6.4. CP-04. Crear un proyecto	64
6.5. CP-05. Filtrar Proyectos por categoría	65
6.6. CP-06. Filtrar Proyectos por habilidad	65
6.7. CP-07. Ver detalles de proyecto	66
6.8. CP-08. Ver información de perfil	66
6.9. CP-09. Editar Perfil	67
6.10. CP-10. Solicitar unirse a proyecto	68
6.11. CP-11. Editar Proyecto	68
6.12. CP-12. Eliminar proyecto	69
6.13. CP-13. Ver información de perfil de un solicitante	69
6.14. CP-14. Aceptar solicitantes	70
6.15. CP-15. Ver mis solicitudes	70
6.16. CP-16. Ver mis solicitudes	71
6.17. CP-17. Ver mis proyectos	71
6.18. CP-18. Crear Repositorio	72
6.19. CP-19. Iniciar Proyecto	72
6.20. Resultados y soluciones de los casos de prueba	73
7.1. Tareas Sprint 1	77
7.2. Tareas Sprint 2	78
7.3. Tareas Sprint 3	79

LISTA DE TABLAS

7.4. Tareas Sprint 4	80
7.5. Tareas Sprint 5	81
7.6. Tareas Sprint 6	82
7.7. Tareas Sprint 7	83
7.8. Coste Simulado	85
7.9. Coste Real	86

Capítulo 1

Introducción

1.1. Contexto

Casi desde el inicio de internet, ha existido una filosofía que promueve la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Esta corriente se denomina **Open Source** o **Código Abierto** [29], y cada vez es más y más común entre desarrolladores de todo el mundo.

Los **contributors** son aquellos que intentan aportar a la comunidad *Open Source* de cualquier manera. Todo el mundo puede aportar algo, que siempre se valorará positivamente el intento. Puede ser desde encontrar *bugs*, ayudar con la documentación, realizar propuestas de cambios, etc.

Todo este trabajo se suele realizar en repositorios de código, es decir, un lugar donde todo el mundo pueda acceder al proyecto a la vez, siendo este visible para todos los usuarios. El más conocido actualmente es *Github* [23], el alojamiento de referencia de casi toda la comunidad *Open Source*. Permite que todos sus usuarios puedan ver el código fuente de los proyectos de otros usuarios. Estos proyectos pueden estar formados por uno o varios usuarios, pudiendo todos ellos modificar y actualizar el código. Además, permite realizar *forks* o bifurcaciones de dichos repositorios para el uso y modificación del resto de usuarios, sin que afecte al repositorio original. También, se pueden proponer mejoras al autor o al equipo de trabajo encargado del repositorio. Estas propuestas se realizan mediante las denominadas *pull requests*, permitiendo al autor, o equipo, decidir si aceptar o no la propuesta.

Find Your Project nace para proporcionar una herramienta de incubación para todos los proyectos incipientes de desarrollo software en la plataforma de *Github*.

1.2. Motivación

La idea surge después de encontrarme confinado en casa en la primavera de 2020, como toda la población, teniendo una gran cantidad de tiempo libre. En esos momentos, me hubiera gustado poder ayudar a iniciar algún nuevo proyecto, ya que yo me encontraba sin ideas, pudiendo colaborar así con la comunidad y ayudando otros desarrolladores a llevar a cabo sus proyectos. Ahí fue cuando se me ocurrió crear una aplicación que pudiera hacer de intermediario entre promotores de proyectos o como les llamaremos a lo largo de la aplicación **organizadores**, cuyas ideas no hayan podido ver la luz puesto que necesitaban ayuda en el desarrollo del proyecto, con los desarrolladores cuyo objetivo es poder ayudar y contribuir a la comunidad, ayudando a desarrollar y llevar a cabo estos proyectos como colaboradores o *contributors*.

Viendo que los dispositivos móviles son las herramientas utilizadas por casi toda la población en su día a día, y fundamentalmente por el segmento de población al que iría destinada este proyecto, y teniendo en cuenta lo accesibles que se han convertido, se decidió crear una red social mediante una aplicación para estos dispositivos, permitiendo que esta sea usada en cualquier sitio y lugar. Se espera poder ayudar a crecer a la comunidad abarcando todo los campos posibles del desarrollo de software, como pueden ser análisis, diseño, programación, etc.

1.3. Estudio de alternativas

En esta sección se mostrarán algunos productos similares al proyecto que vamos a realizar. Para ello, se ha realizado una búsqueda exhaustiva por internet hasta llegar a todos aquellos productos que nombraremos a continuación.

1.3.1. Twago

Twago [46] es una página web de búsqueda de desarrolladores *freelance* (por cuenta propia). Tiene dos apartados, tanto para empresas que buscan desarrolladores, pudiendo filtrarles y ver sus perfiles; como para desarrolladores que buscan entrar en un proyecto.

Ventajas

- Gran cantidad de proyectos
- Gran cantidad de desarrolladores
- Aparecen los salarios asociados a cada proyecto

Desventajas

- Interfaz poco vistosa y usable
- No dispone de aplicación móvil
- No está destinada a la incubación de proyectos *open source*

1.3.2. Fiverr

Fiverr [20] es una plataforma para que trabajadores *freelance* vendan su trabajo. Cada usuario introduce los “productos” (desde diseño de una página web hasta un dibujo) que vende, el precio, y las fechas de entrega, al igual que otra información relevante.

Ventajas

- Amplio catálogo de servicios
- Amplia información de cada producto y vendedor
- Interfaz muy usable
- Dispone de aplicación para móvil

Desventajas

- Soluciones únicas o a corto plazo
- Posibilidad de obtener un servicio insuficiente o inferior a lo acordado
- Obligatorio pago
- No está destinada a la incubación de proyectos *open source*

1.3.3. Malt

Malt [35] al igual que Twago, es una página que sirve para conectar a trabajadores *freelance* directamente con Empresas. En el caso de las empresas, dispone de diferentes planes para poder adaptar tus necesidades a cada uno de ellos.

Ventajas

- Gran cantidad de empresas
- Diferentes planes de servicio
- Interfaz intuitiva

Desventajas

- Solo disponible bajo suscripción a alguno de los planes por parte de la empresa
- No está destinada a la incubación de proyectos *open source*

1.3.4. Conclusiones obtenidas

Como se puede observar, todas las aplicaciones mencionadas anteriormente están enfocadas en proyectos con clientes o relacionados con empresas. Ninguna de ellas se enfoca en la comunidad *Open Source*, por lo que creo que nuestra aplicación puede ser un punto de inflexión en esta comunidad, ayudándola a interconectar entre ella y a ayudar a la creación de un mayor número de proyectos.

La observación de estas aplicaciones no obstante permite obtener algunas lecciones para tener en cuenta al realizar la concepción del producto que se desarrollará en este trabajo como pueden ser cuestiones de diseño de interfaz, usabilidad, cómo plasmar las “habilidades” a ofrecer por los usuarios y necesitadas para la organización de un proyecto en incubación.

1.4. Objetivos

A continuación, detallaremos los objetivos de este proyecto:

- Desarrollar una app que sirva como nexo de unión para desarrolladores de todas las partes del mundo con el inicio de sus proyectos
- Enfrentarme a la creación de un proyecto de Desarrollo Software real desde cero
- Publicar mi primera aplicación en una tienda virtual, en este caso Google Play.

1.5. Estructura de la memoria

Este documento se estructura de la siguiente forma:

Capítulo 1: Introducción. En este capítulo se describe el origen de la idea del proyecto, la motivación para llevarla a cabo y los objetivos que se pretenden conseguir con el proyecto.

Capítulo 2: Requisitos y planificación. En este capítulo se muestra el marco de trabajo elegido y la adaptación del proyecto. Se describen los requisitos en forma de historias de usuario y la planificación del proyecto. También se expondrá el plan de riesgos y los presupuestos asociados al proyecto.

Capítulo 3: Tecnologías utilizadas. En este capítulo se explican las tecnologías y herramientas que se han utilizado a lo largo del desarrollo del proyecto.

Capítulo 4: Análisis En este capítulo se documenta el resultado de las tareas de análisis realizadas a lo largo del proyecto a partir de todo lo recogido y explicado en el capítulo de requisitos y planificación.

Capítulo 5: Diseño. En este capítulo se muestran las decisiones de diseño tomadas, así como los diagramas, modelos elaborados y bocetos de la aplicación.

Capítulo 6: Implementación y pruebas. En este capítulo se explican algunos aspectos relacionados con la implementación del sistema y las distintas pruebas realizadas. También se expondrán todas las dificultades que se han encontrado en la implementación.

Capítulo 7: Seguimiento del proyecto. En este capítulo se mostrará cómo ha ido avanzando el proyecto a lo largo del tiempo de forma detallada.

Capítulo 8: Conclusiones. En este capítulo se expondrán las conclusiones finales obtenidas, al igual que algunas propuestas de mejora para llevar a cabo en el futuro.

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

Anexo B Resumen de enlaces adicionales: Incluye enlaces de interés sobre el proyecto como el repositorio de código y el enlace a la Google Play Store.

Capítulo 2

Requisitos y Planificación

2.1. Scrum

Para la realización de este proyecto, se ha elegido el marco de trabajo **Scrum**. Se trata de un marco que permite el trabajo colaborativo entre equipos. En él se anima a los equipos a aprender a través de la experiencia y a organizarse de forma autónoma a la vez que se aborda un problema, y también a analizar y reflexionar sobre lo realizado anteriormente en el proyecto, ya sea bueno o malo, para realizar una mejora continua [12].

Dentro de Scrum, podemos distinguir tres partes indispensables para el desarrollo del proyecto: roles, eventos y artefactos.

2.1.1. Roles en Scrum

El equipo Scrum esta compuesto por tres roles principales [13]:

- **Product Owner.** Es el responsable de comunicar la visión de producto del cliente al equipo de desarrollo. Se deben respetar sus decisiones y estas se pueden encontrar en el *Product Backlog*, artefacto que será explicado más adelante (Sección 2.1.2).
- **Scrum Master.** Es el encargado de comprobar que se cumple Scrum. También es el nexo de unión entre el equipo de trabajo y la organización. Es un facilitador que se encarga de desbloquear cualquier situación y allanar el camino al equipo de desarrollo.
- **Equipo de desarrollo.** Es el resto de personas que componen el equipo Scrum. Se encargan de desarrollar el producto mientras aportan valor en cada *incremento*, artefacto que será explicado más adelante (Sección 2.1.2).

2.1.2. Artefactos

Como resultado a las actividades en Scrum, se crean los llamados artefactos, los cuales representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave. Los artefactos considerados son los siguientes [34]:

- **Product Backlog.** Es una lista ordenada por prioridad en las que se encuentran las características y mejoras que se buscan añadir al producto. Esto se realiza en forma de historias de usuario. El principal responsable es el *Product Owner*. El *Product Goal* es el objetivo que se persigue, que define un estado futuro del producto al que se quiere llegar.
- **Sprint Backlog.** Es la lista de historias de usuario en las que trabajará el equipo de desarrollo en el sprint. Los responsables de este artefacto son el equipo de desarrollo. El objetivo de este artefacto es el *Sprint Goal*, el cual define los objetivos a cumplir en el sprint. Este es más flexible respecto al trabajo realizado para conseguirlo.
- **Incremento.** Es el resultado de todo el trabajo realizado en un sprint. Todos los incrementos son acumulativos con los sprint anteriores, hasta intentar acercarse al *Product Goal*.

2.1.3. Eventos

Los eventos en Scrum tienen una periodicidad y dirección definida, y son cuatro [34]:

- **Sprint.** Es el contenedor del resto de eventos de Scrum. Son periodos de tiempo de entre 1 y 4 semanas, cuya ocurrencia es secuencial. Es decir, ocurren uno detrás de otro mientras dure el proyecto. Durante un sprint, no se pueden cambiar los requisitos. Los sprints deben ser de la misma duración a lo largo del proyecto
- **Sprint Planning.** Evento que da comienzo de un sprint. En esta reunión, participan todos los miembros del equipo Scrum. Sirve para planificar el contenido y los objetivos del sprint.
- **Daily Scrum.** Evento diario de duración máxima 15 minutos donde cada miembro del equipo informa sobre su progreso desde la última reunión.
- **Sprint Review.** Evento que da comienzo al finalizar un sprint, donde se discute sobre los objetivos cumplidos. Es posible hacer ajustes en el *Product Backlog*
- **Sprint Retrospective.** Este evento finaliza el sprint. En él se evalúa el sprint, para poder sacar conclusiones sobre el trabajo realizado y mejorar de cara a los siguientes sprints.

2.2. Scrum en este proyecto

Para el desarrollo de este proyecto, se va a adaptar Scrum a las condiciones y al contexto para poder trabajar de la manera más cómoda y eficiente posible.

Respecto a los roles, se asignarán de la siguiente manera: el estudiante ostentará los roles de *Product Owner* y equipo de desarrollo, mientras que la tutora ostentará el rol de *Scrum Master*.

En cuanto a los artefactos, todos estarán a cargo del estudiante, puesto que asume los roles de los responsables de dichos artefactos.

Respecto a los eventos, los sprint tendrán una duración de 2 semanas. Las reuniones de *Sprint Planning*, *Sprint Review* y *Sprint Retrospective* se realizarán cada dos semanas, coincidiendo el mismo día que se termina un sprint para aprovechar y comenzar el siguiente. Por otro lado, las *Daily Scrum* son incompatibles con el contexto e innecesarias dado el equipo de trabajo formado por una sola persona, por lo que en este proyecto contaremos con *Weekly Scrum*, realizadas una vez por semana para supervisar el avance y permitir desbloquear situaciones y aclarar dudas.

2.3. Planificación Inicial

A continuación, se mostrará la planificación inicial del proyecto. Siguiendo la guía docente [47], el Trabajo de Fin de Grado se corresponde con 12 ECTS, lo que equivale a una cantidad de **300 horas**. Por ellos, hemos decidido realizar un plan basado en 8 sprints. Como hemos comentado anteriormente en la Sección 2.2 cada sprint tendrá una duración de 2 semanas, siendo previsto dedicar entre 15 y 20 horas semanales, lo que nos da una duración de entre 30 y 40 horas por sprint. Para realizar la estimación, utilizaremos la media de las dos duraciones, es decir, **35 horas**.

Para obtener las historias del Product Backlog que se asignarán a cada sprint, utilizaremos un sistema de puntos de historia lineal, donde cada **punto de historia** equivale a **5 horas** de trabajo. Está elección de historias y asignación de puntos se realizará en la reunión *Sprint Planning*.

Además, también se incluye en esta planificación un **sprint 0**, cuya duración no está prefijada y ha sido de 20 horas. En este sprint se han realizado las tareas de preparación inicial y planificación, para que directamente en el sprint 1 se pueda comenzar a desarrollar de forma eficaz. A su vez, también se han añadido 2 sprints extra, para que sean utilizados en caso de necesitarlo.

Por tanto, sumando todo lo comentado anteriormente, la estimación inicial sería una cantidad de **300 horas**. En caso de necesitar alguno de los sprints adicionales, la estimación inicial variaría hasta las **335 horas** o **370 horas**. Como podemos ver, en todos los casos se superan las 300 horas de trabajo del estudiante correspondientes a los 12 ECTS.

En la Tabla 2.1, podemos ver el calendario del proyecto.

2.4. Product Backlog inicial

Como se ha explicado anteriormente en la Sección 2.2, es el propio alumno el que ostenta el cargo de *Product Owner*, por lo que es el encargado de elaborar el *Product Backlog*. Después de trabajar en ello y gracias a las aportaciones de la tutora, se ha elaborado el *Product Backlog* que aparece en la Tabla 2.2.

El *Product Backlog* inicial se elabora mediante historias denominadas como épicas. Una épica es un conjunto de trabajo grande que puede dividirse en subtarefas mas concretas (historias de usuario), en función de la necesidades o solicitudes realizadas por los clientes o usuarios finales [36].

Cada épica tendrá un código, el cual estará indicado de la forma **EPX** donde **X** será remplazada por el número de épica. Todas las épicas estarán escritas de la siguiente forma: “Como *stakeholder* quiero *objetivo* para *razón que aporte valor como stakeholder*”, haciendo referencia *stakeholder* a los interesados en el proyecto. Esta frase en la que se introduce valor permite analizar las prioridades a la hora de ordenar el product backlog e ir avanzando en la construcción del producto mínimo viable.

Número	Épica
EP1	Como usuario quiero poder crear proyectos incipientes para llevar a cabo una idea de proyecto
EP2	Como usuario quiero poder unirme a proyectos incipientes para ayudar a llevarlos a cabo
EP3	Como usuario quiero poder tener un control sobre mi perfil para poder gestionar mi información personal

Tabla 2.2: *Product Backlog* inicial

A continuación, se ha realizado una división por **historias de usuario** de cada épica. Cada historia tendrá un código asignado, que será de la forma **EPXHY** donde **X** es el número de épica al que pertenece la historia e **Y** es el número de historia. Todo esto se puede ver en la Tabla 2.3.

Número	Historia
EP1H1	Como usuario quiero crear un proyecto para que otros usuarios puedan solicitar unirse
EP1H2	Como usuario y administrador de un proyecto quiero poder escoger a un usuario para el puesto requerido de entre todos los que hayan solicitado, para elegir a la persona más adecuada para el proyecto

EP1H3	Como usuario y organizador de un proyecto quiero poder ver la información de perfil de los aplicantes a un puesto de mi proyecto para ver toda la información de dicho perfil, así como las tecnologías que conoce y la forma de contacto.
EP1H4	Como usuario y organizador de un proyecto quiero poder crear un repositorio de GitHub directamente desde la aplicación con los miembros del equipo inicial, para establecer un repositorio donde poder comenzar a trabajar
EP1H5	Como usuario y organizador de un proyecto ya iniciado, quiero dejar constancia del repositorio en GitHub para que otros usuarios puedan unirse desde allí
EP1H6	Como usuario y organizador del proyecto, quiero poder modificar la información y requisitos del proyecto para mejorar la explicación y adaptar los requisitos tecnológicos
EP1H7	Como usuario y organizador del proyecto, quiero poder eliminar una propuesta de proyecto, para poder desecharlo y notificar a los solicitantes la eliminación de dicho proyecto
EP2H1	Como usuario quiero poder ver todas las propuestas de proyectos y filtrarlas para poder elegir el proyecto que mejor se adapte a mí
EP2H2	Como usuario quiero poder ver la información de un proyecto de forma detallada, así como todas las tecnologías necesarias para poder decidir si es adecuado para mí
EP2H3	Como usuario quiero poder solicitar unirme a un proyecto para formar parte del equipo inicial
EP2H4	Como usuario quiero poder ver todas las solicitudes de unión a proyectos que haya realizado y poder cancelarlas, para tener un mayor control sobre mi cuenta frente a cambios de opinión
EP3H1	Como usuario quiero poder iniciar sesión y registrarme con mi cuenta de GitHub, para poder utilizar mi información de usuario de Github en la aplicación
EP3H2	Como usuario quiero poder ver y modificar la información de mi perfil que contendrá información personal y las tecnologías que conozco o soy experto, para poder mantenerla al día
EP3H3	Como usuario quiero poder ver todos los proyectos a los que pertenezco y los que he creado, para poder llevar un control sobre ellos
EP3H4	Como usuario quiero poder cerrar mi sesión en la aplicación, para poder acceder nuevamente con el mismo u otro usuario, o recargar el token de autenticación

Tabla 2.3: Historias de usuario

Además, se ha realizado un tabla con el orden de prioridad con el que se realizaran las historias de usuario. Todo esto, por supuesto, está sujeto a cambios a lo largo del proyecto. Se puede ver en la Tabla 2.4.

2.4. PRODUCT BACKLOG INICIAL

Nº Sprint/Evento	Fecha inicio	Fecha Fin	Anotaciones
Sprint 0	01/02/2022	17/02/2022	
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	17/02/2022		
Sprint 1	17/02/2022	03/03/2022	
<i>Scrum Weekly</i>	24/02/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	03/02/2022		
Sprint 2	03/03/2022	17/03/2022	
<i>Scrum Weekly</i>	10/03/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	17/03/2022		
Sprint 3	17/03/2022	31/03/2022	
<i>Scrum Weekly</i>	24/03/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	31/03/2022		
Sprint 4	31/03/2022	14/04/2022	
<i>Scrum Weekly</i>	07/04/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	14/04/2022		
-	14/04/2022	21/04/2022	Vacaciones de Semana Santa y del estudiante
Sprint 5	21/04/2022	05/05/2022	
<i>Scrum Weekly</i>	28/04/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	05/05/2022		
Sprint 6	05/05/2022	19/05/2022	
<i>Scrum Weekly</i>	12/0/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	19/05/2022		
Sprint 7	19/05/2022	02/06/2022	
<i>Scrum Weekly</i>	26/05/2022		
<i>Sprint Review, Sprint Retrospective Y Sprint Planning</i>	02/06/2022		
Sprint 8	02/06/2022	16/06/2022	
<i>Scrum Weekly</i>	09/06/2022		
<i>Sprint Review, Sprint Retrospective</i>	16/06/2022		Se realizará el <i>Sprint Planning</i> en caso de necesitar el próximo sprint
Sprint Extra 1	16/06/2022	30/06/2022	
<i>Scrum Weekly</i>	23/06/2022		
Límite solicitud defensa convocatoria ordinaria	24/06/2022		Los tribunales se realizarán antes del 11/07/2022
<i>Sprint Review, Sprint Retrospective</i>	30/06/2022		Se realizará el <i>Sprint Planning</i> en caso de necesitar el próximo sprint
Sprint Extra 2	30/06/2022	14/07/2022	
<i>Scrum Weekly</i>	07/07/2022		
<i>Sprint Review, Sprint Retrospective</i>	14/07/2022		
Límite solicitud defensa convocatoria extraordinaria	15/07/2022		Los tribunales se realizarán antes del 28/07/2022

Tabla 2.1: Planificación Inicial de Sprints

Orden	Historia
1	EP3H1
2	EP3H4
3	EP1H1
4	EP2H1
5	EP2H2
6	EP3H2
7	EP2H3
8	EP1H6
9	EP1H7
10	EP1H3
11	EP1H2
12	EP2H4
13	EP1H4
14	EP1H5
15	EP3H3

Tabla 2.4: Prioridad Historias de usuario

2.5. Product Backlog final

Según ha ido avanzando el proyecto, se han realizado cambios en el *Product Backlog* inicial y en el orden de prioridad de las historias. A continuación, en las Tablas 2.5 y 2.6 podemos ver las versiones finales de las mismas.

Número	Historia
EP1H1	Como usuario quiero crear un proyecto para que otros usuarios puedan solicitar unirse
EP1H2	Como usuario y organizador de un proyecto quiero poder escoger a un usuario para el puesto requerido de entre todos los que hayan solicitado, para elegir a la persona más adecuada para el proyecto
EP1H3	Como usuario y organizador de un proyecto quiero poder ver la información de perfil de los aplicantes a un puesto de mi proyecto para ver toda la información de dicho perfil, así como las tecnologías que conoce y la forma de contacto.
EP1H4	Como usuario y organizador de un proyecto quiero poder crear un repositorio de GitHub directamente desde la aplicación con los miembros del equipo inicial, para establecer un repositorio donde poder comenzar a trabajar
EP1H5	Como usuario y organizador de un proyecto ya iniciado, quiero dejar constancia del repositorio en GitHub para que otros usuarios puedan unirse desde allí
EP1H6	Como usuario y organizador del proyecto, quiero poder modificar la información y requisitos del proyecto para mejorar la explicación y adaptar los requisitos tecnológicos

EP1H7	Como usuario y organizador del proyecto, quiero poder eliminar una propuesta de proyecto, para poder desecharlo y notificar a los solicitantes la eliminación de dicho proyecto
EP2H1	Como usuario quiero poder ver todas las propuestas de proyectos y filtrarlas para poder elegir el proyecto que mejor se adapte a mí
EP2H2	Como usuario quiero poder ver la información de un proyecto de forma detallada, así como todas las tecnologías necesarias para poder decidir si es adecuado para mí
EP2H3	Como usuario quiero poder solicitar unirme a un proyecto para formar parte del equipo inicial
EP2H4	Como usuario quiero poder ver todas las solicitudes de unión a proyectos que haya realizado, para tener un mayor control sobre mi cuenta frente a cambios de opiniones
EP2H5	Como usuario quiero poder filtrar todas las propuestas de proyecto por categoría para poder elegir el proyecto que mejor se adapte a lo que busco
EP2H6	Como usuario quiero poder cancelar todas las solicitudes de unión a proyectos que haya realizado, para tener un mayor control sobre mi cuenta frente a cambios de opiniones
EP3H1	Como usuario quiero poder iniciar sesión y registrarme con mi cuenta de GitHub, para poder utilizar mi información de usuario de Github en la aplicación
EP3H2	Como usuario quiero poder ver la información de mi perfil que contendrá información personal y las tecnologías que conozco o soy experto, para poder comprobar que esta actualizada
EP3H3	Como usuario quiero poder ver todos los proyectos a los que pertenezco y los que he creado, para poder llevar un control sobre ellos
EP3H4	Como usuario quiero poder cerrar mi sesión en la aplicación, para poder acceder nuevamente con el mismo u otro usuario, o recargar el token de autenticación
EP3H5	Como usuario quiero poder modificar la información de mi perfil que contendrá información personal y las tecnologías que conozco o soy experto, para poder mantenerla al día
EP3H6	Como usuario quiero poder recibir notificaciones cuando haya algún cambio sobre un proyecto o solicitud, para poder conocer su estado sin necesidad de acceder a la aplicación

Tabla 2.5: Historias de usuario Final

Como podemos observar, se han añadido las historias de usuario EP2H5, EP2H6, EP3H5 y EP3H6. Las creación de las tres primeras historias mencionadas anteriormente se debe a que sus historias iniciales eran demasiado extensas como para realizarse en un único sprint o en una única historia. Sin embargo, la historia EP3H6 se añadió durante el desarrollo del proyecto, al observar que podría ser una funcionalidad muy interesante para la aplicación. Todo esto ha repercutido en el orden de prioridad de las historias de usuario. Las tres primeras, se han situado a continuación de sus historias previas, mientras que a EP3H6 ha sido

Orden	Historia
1	EP3H1
2	EP3H4
3	EP1H1
4	EP2H1
5	EP2H2
6	EP2H5
7	EP3H2
8	EP3H5
9	EP2H3
10	EP1H6
11	EP1H7
12	EP1H3
13	EP1H2
14	EP2H4
15	EP2H6
16	EP3H3
17	EP1H4
18	EP1H5
19	EP3H6

Tabla 2.6: Prioridad Historias de usuario Final

colocada al final. También ha habido una reorganización en el orden de las últimas historias de cara a los sprints finales. Las historias asociadas a cada sprint podrán verse en el Capítulo 7.

2.6. Plan de riesgos

En la fase de planificación, hemos realizado un plan de riesgos.

Un riesgo [13] es un suceso o acción relacionada con el futuro, cuyo aparición afectará de alguna forma al desarrollo del proyecto. Para ello, hemos creado este plan de riesgos, cuyo propósito es controlar dichos riesgos a lo largo del ciclo de vida de vida del proyecto, tanto su ocurrencia como sus consecuencias.

En nuestro caso, vamos a tener en cuenta cuatro aspectos:

- **Probabilidad.** Se trata de la posibilidad de que suceda el riesgo. Se le asignará uno de los siguientes valores: baja, media, alta.
- **Impacto.** Se trata de la el efecto que tendrá el riesgo en caso de materializarse Se le asignará uno de los siguientes valores: bajo, medio, alto.

Riesgo R01	
Título	Ausencia temporal del estudiante por enfermedad
Descripción	El estudiante no puede dedicar tiempo a trabajar en el TFG durante un período determinado debido a causas médicas que le impidan trabajar con normalidad
Probabilidad	Media
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ■ Llevar una vida saludable y buena alimentación ■ Evitar actividades que supongan un riesgo para la salud
Plan de contingencia	<ul style="list-style-type: none"> ■ Si la duración de la ausencia es unos días, tratar de recuperar el tiempo perdido distribuyéndolo entre los siguientes días ■ Si la duración es más prolongada, pasar tareas al siguiente sprint ■ Si ocurre en los últimos sprints y las dos anteriores no se pueden realizar, hacer uso de los sprints extra de refuerzo

Tabla 2.7: Riesgo R01. Ausencia temporal del estudiante por enfermedad

- **Plan de mitigación.** Se trata del conjunto de acciones a seguir para reducir la probabilidad de que ocurra el riesgo.
- **Plan de contingencia.** Se trata del conjunto de acciones a seguir para reducir el impacto de un riesgo una vez se ha materializado

Los riesgos identificados a continuación se dividirán en dos tipos: **generales**, los cuales se pueden aplicar a cualquier proyecto, y **particulares**, los cuales son específicos de nuestro proyecto.

Los riesgos generales se encuentran entre las Tablas 2.7 y 2.13, mientras que los riesgos particulares se encuentran entre las Tablas 2.14 y 2.15.

2.7. Plan de presupuestos

Se ha realizado un plan presupuestario como ultima parte de la fase de planificación. Este se dividirá en dos presupuestos bien diferenciados: un **presupuesto simulado**, el cual

Riesgo R02	
Título	Fallos técnicos en el equipo de trabajo del estudiante
Descripción	El equipo de trabajo del estudiante sufre algún fallo técnico y no se puede continuar trabajando con normalidad
Probabilidad	Baja
Impacto	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Hacer una copia de seguridad de los documentos importantes cada poco tiempo ■ Mantener una copia de los documentos importantes en algún repositorio online ■ Trabajar en lo posible de forma online con los documentos (Overleaf, Microsoft Teams...)
Plan de contingencia	<ul style="list-style-type: none"> ■ Si se ha realizado copia de seguridad de los documentos, recuperarlos en otro dispositivo para poder seguir trabajando ■ Si no se han realizado copias de seguridad, intentar recuperar dichos documentos con los distintos mecanismos que ofrecen los propios programas

Tabla 2.8: Riesgo R02. Fallos técnicos en el equipo de trabajo del estudiante

2.7. PLAN DE PRESUPUESTOS

Riesgo R03	
Título	Ausencia temporal de la tutora por enfermedad
Descripción	La tutora no puede continuar con el seguimiento del TFG durante un período determinado debido a causas médicas que le impidan trabajar con normalidad
Probabilidad	Media
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ■ Tener contacto asiduamente para poder saber si el riesgo se materializa en algún momento
Plan de contingencia	<ul style="list-style-type: none"> ■ Continuar con el trabajo hasta poder realizar la siguiente reunión con la tutora, apuntando dudas o problemas que surjan ■ Comunicarse mediante otros medios que no sea videoconferencia si es posible, para enviar dudas o un resumen del contenido a tratar en la reunión.

Tabla 2.9: Riesgo R03. Ausencia temporal de la tutora por enfermedad

Riesgo R04	
Título	Confinamiento domiciliario debido a COVID-19
Descripción	Se produce una orden de confinamiento domiciliario debido al empeoramiento de la situación o a un posible contagio por COVID-19 determinado debido a causas médicas que le impidan trabajar con normalidad
Probabilidad	Baja
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ■ Conocer y saber utilizar herramientas de comunicación telemática (videoconferencias, chats...)
Plan de contingencia	<ul style="list-style-type: none"> ■ Realizar el seguimiento con la tutora mediante videoconferencia u otras herramientas.

Tabla 2.10: Riesgo R04. Confinamiento domiciliario debido a COVID-19

Riesgo R05	
Título	Mala estimación de tiempo y trabajo de algunas actividades
Descripción	La estimación de algunas actividades ha sido errónea y el tiempo empleado no se asemeja a lo estimado
Probabilidad	Media
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ■ Establecer uno o más sprint de refuerzo al final del proceso de desarrollo del proyecto que puedan cubrir estas necesidades de tiempo adicional ■ Ser realistas con las estimaciones (se mejora con la practica) ■ Centrarse más en las partes importantes del proyecto que en los detalles
Plan de contingencia	<ul style="list-style-type: none"> ■ Utilizar los sprints de refuerzo ■ Recortar el alcance del proyecto en caso de no tener más sprints de refuerzo.

Tabla 2.11: Riesgo R05. Mala estimación de tiempo y trabajo de algunas actividades

Riesgo R06	
Título	Desconocimiento de la arquitectura a utilizar
Descripción	La arquitectura a utilizar no se conoce y por tanto hay que aprender a utilizarla
Probabilidad	Media
Impacto	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Familiarizarse lo antes posible con la arquitectura que se va a utilizar
Plan de contingencia	<ul style="list-style-type: none"> ■ Buscar documentación oficial o tutoriales donde se explique el uso de dicha arquitectura ■ Tratar de buscar alternativas factibles y ya conocidas para utilizar en lugar de la nueva arquitectura

Tabla 2.12: Riesgo R06. Desconocimiento de la arquitectura a utilizar

2.7. PLAN DE PRESUPUESTOS

Riesgo R07	
Título	Menor tiempo empleado que el previsto debido a otros asuntos académicos
Descripción	En algún sprint no se puede dedicar tanto tiempo como se ha estimado inicialmente en la reunión de planificación debido a otros eventos académicos que se solapan con la duración del sprint (prácticas en empresa, exámenes, entregas...)
Probabilidad	Alta
Impacto	Medio
Plan de mitigación	<ul style="list-style-type: none"> ■ Intentar planificar cada sprint conociendo que eventos académicos sucederán durante el tiempo que dure dicho sprint
Plan de contingencia	<ul style="list-style-type: none"> ■ Pasar tareas sin finalizar al sprint siguiente ■ Recuperar el tiempo no empleado en futuros sprints, cuando se disponga de más tiempo

Tabla 2.13: Riesgo R07. Menor tiempo empleado que el previsto debido a otros asuntos académicos

Riesgo R08	
Título	Cambio en los requisitos del proyecto
Descripción	Los requisitos del proyecto cambian por parte del estudiante, que asume el cargo de <i>Product Owner</i> , cuando el desarrollo ya ha sido iniciado
Probabilidad	Media
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ■ Intentar aclarar todas las funcionalidades propuestas desde el principio
Plan de contingencia	<ul style="list-style-type: none"> ■ Si es un cambio estético, de IU o similar, se puede llevar a cabo ■ Si es un cambio que supone mucho trabajo, en principio no se efectuaría ya que el objetivo final del TFG no es comercializarlo en principio

Tabla 2.14: Riesgo R09. Cambio en los requisitos del proyecto

Riesgo R09	
Título	Necesidad de aprendizaje de nuevas herramientas debido a actualizaciones en las tecnologías utilizadas
Descripción	Algunas de las herramientas o tecnologías utilizadas (API Android, programas utilizados...) reciben una actualización durante el transcurso del proyecto y cambian su aspecto y/o comportamiento
Probabilidad	Baja
Impacto	Bajo
Plan de mitigación	<ul style="list-style-type: none"> ▪ Definir desde un principio las versiones de las tecnologías y herramientas a utilizar.
Plan de contingencia	<ul style="list-style-type: none"> ▪ Si se llega a actualizar la herramienta a una nueva versión con características que supongan problemas, volver a la versión inicial

Tabla 2.15: Riesgo R09. Necesidad de aprendizaje de nuevas herramientas debido a actualizaciones en las tecnologías utilizadas

estima los costes que tendría el proyecto de tratarse de un proyecto real cuyo propósito es la salida al mercado, y un **presupuesto real**, el cual estima los costes reales de este Trabajo de Fin de Grado.

2.7.1. Presupuesto simulado

Lo primero a tener en cuenta es el salario de los trabajadores. Contrastando los datos en diferentes paginas [33] [31] que comprueban la información de diversas fuentes, se ha obtenido un valor del sueldo medio en España para un desarrollador Android de 2400€al mes. Un mes tiene 22 días laborables de media (algunos 20 otros 24, dependiendo de si se cuentan sábados) cada uno de ellos con 8 horas de trabajo. Esto nos da una media de 176 horas/mes, siendo cada una de ellas pagada a 13,64€. De acuerdo con la planificación inicial, las horas totales serian 300, por lo que el habría que pagar al trabajador **4092€**. Este salario ya estaría exento de impuestos, habiendo sido estos calculados anteriormente para tener la cifra **1541€**, que serán destinados a Seguridad social del empleado.

Un QA-Tester tiene un salario mensual neto de 1589,8€[27], lo que sería 9,03€/hora. En el caso de un Product Owner, tiene un salario mensual neto de 2720€[26], lo que nos da 15,45€/hora. El salario mensual neto de un Ingeniero de Software (Arquitecto Software) es de 2738€[25], lo que nos da 15,55€/hora. Y por último un Scrum Master tiene un sueldo neto de 2709€[28], lo que nos da 15,39€/hora.

La dedicación esperada en horas de cada rol representado por el alumno será la siguiente:

- **Product Owner:** 30 horas (10)
- **QA-Tester:** 30 horas (10)
- **Ingeniero de Software:** 60 horas (20)
- **Desarrollador Android:** 180 horas (60)

Respecto al equipo que utilizará el desarrollador, será un *MacBook Pro 2021* [8]. Su precio es de 2249€ con una estimación de vida útil de 6 años (72 meses), lo que supone un precio de 31.23 €/mes. Al ser la duración estimada del proyecto 5 meses, el coste total del dispositivo serían **156.18€**. También se necesitará un dispositivo móvil para poder probar la app. En este caso, vamos a contar con un *Samsung Galaxy S22* [44] cuyo precio es de 849€. Estimando su vida útil en 4 años (48 meses), saldría un coste de 17.69€/mes. Para la duración del proyecto, el coste de este dispositivo sería de **88.44€**.

Respecto al coste de las herramientas, hay varias que permiten ser utilizadas de forma gratuita, mientras que por otras tendremos que pagar para poder utilizarlas. Comenzamos con *BalsamiQ Cloud* [10] cuyo coste es de 7.94€/mes, dando un coste de **39.7€**. También necesitaremos una licencia de *Astah Professional* [9], que supone un coste de 8.82€/mes, o lo que es lo mismo, **44.1€** para los meses de trabajo. Además, también necesitaremos *Microsoft 365* [38] con un coste de **52.5€** (10.5 €/mes).

Por último, como espacio de trabajo utilizaremos *ViaLab Coworking* [48], el cual tiene un precio de 160€ al mes, lo que resulta en **800€** para 5 meses.

Sumando todos los precios mencionados anteriormente, obtenemos un coste de **9920,52€**. A esto, le sumaremos un 25 %, para tener la capacidad de hacer frente a imprevistos económicos o retrasos, puesto que todos los costes están calculados en función del tiempo. Todo esto nos da un coste total de **12400,65€**. Estos cálculos pueden verse reflejados de forma más clara en la Tabla 2.16.

2.7.2. Presupuesto real

En el presupuesto real, no se aplicarán algunos de los costes mencionados en el apartado anterior. Estos son el salario, puesto que se trata de un estudiante realizando su Trabajo de fin de grado; y el espacio de co-working, ya que el estudiante realizará su trabajo desde su casa.

Por otro lado, al trabajar desde casa, tenemos que tener en cuenta los gastos de luz que produce el equipo que utiliza el estudiante. El consumo medio de un portátil es de 0.22 kWh [21], que en 300 horas serían 66kWh. Considerando que se trabaja entre las 16h y las 20h cada día, el precio medio del kWh en esa franja horaria es de 0.45278€/kWh [45], siendo el coste total de la luz de **29.88 €**.

El estudiante utilizará un portátil *Asus*, cuyo coste total es 1500€. Si se estima que su vida útil es de años (72 meses), el precio del equipo es de **104.16€** (20.83€/mes). Utilizará

Concepto	Precio Unitario	Cantidad	Total
Sueldo Desarrollador Android	13,64€/hora	180 horas	2455,2€
Sueldo Ingeniero de Software	15,55€/hora	60 horas	933€
Sueldo QA-Tester	9,03€/hora	30 horas	270,9€
Sueldo Product Owner	15,45€/hora	30 horas	463,5€
Sueldo Scrum Master	15,39€/hora	300 horas	4617€
Equipo Empleado	31,23€/mes	5 meses	156,18€
Smartphone de pruebas	17,69/mes	5 meses	88,44€
Espacio Coworking	160/mes	5 meses	800€
Licencia <i>BalsamiQ</i>	7,94/mes	5 meses	39,7€
Licencia <i>Astah Professional</i>	8,82/mes	5 meses	44,1€
Licencia <i>Microsoft 365</i>	10,5/mes	5 meses	52,5€
TOTAL			9920,52 €
+25 %			12400,65€

Tabla 2.16: Presupuesto Simulado

también un Xiaomi Redmi Note 10 Pro [37], cuyo precio es de 299€. Estimando su vida útil en 4 años (48 meses) nos queda un coste de **31.15€** (6.23€/mes).

Respecto a las licencias de los programas, en el caso de *Microsof 365* y *Astah Professional* las licencias son ofrecidas de forma gratuita por la universidad. Por otro lado, no se utilizará *BalsamiQ* sino *Figma*, siendo este último de uso gratuito.

Por tanto, el coste real total de proyecto, sumando todos los costes individuales, será de **165.19€**. Podemos verlo en la Tabla 2.17.

Concepto	Precio Unitario	Cantidad	Total
Electricidad	0.45278€/kWh	66kWh	29,88€
Equipo Estudiante	20,83€/mes	5 meses	104,16€
Smartphone de pruebas	6,23/mes	5 meses	31,15€
TOTAL			168.19

Tabla 2.17: Presupuesto Real

Capítulo 3

Tecnologías utilizadas

En este Capítulo se presenta un resumen de las tecnologías utilizadas tanto para la gestión del proyecto, como para el desarrollo.

3.1. GitLab

GitLab [32] es un repositorio de gestión de proyectos. Está dotado de interfaz gráfica y su principal objetivo es integrar múltiples funcionalidades para que solo sea necesaria una única aplicación durante todo el ciclo de vida del desarrollo software. Esto provoca una mejora en la eficiencia y la velocidad de trabajo. Para nosotros, servirá como sistema de control de versiones y como herramienta de gestión de proyectos.

3.1.1. Git

Como podemos deducir del nombre, *GitLab* está construido sobre *Git*. *Git*[22] es un sistema de control de versiones distribuido, de código abierto y gratis. Su propósito es llevar un registro de los cambios producidos en los ficheros del código fuente y la configuración del proyecto, principalmente software aunque también de otro tipo de ficheros.

3.1.2. GitLab Issue Board

GitLab Issue Board [24] es una herramienta usada para la gestión de proyecto software. Está constituida por tableros en los que encontramos diferentes columnas con tarjetas que representan *issues*. Una *issue* es una incidencia que tiene que llevarse a cabo en el proyecto.

Para facilitar el seguimiento de proyecto, se ha utilizado un tablero único dividido en 6 columnas distintas:

- **Product Backlog.** En esta columna se encontrarán todas las tarjetas que representan las distintas historias de usuario en las que se dividen las épicas del *Product Backlog*.
- **Sprint Backlog.** A esta columna se mueven las tarjetas de las historias de usuario de la columna de *Product Backlog* que se van a desarrollar en el actual sprint.
- **Doing.** En esta columna se encuentran las tarjetas de las *issues* que se están realizando actualmente, las cuales se moverán desde la columna de *Sprint Backlog*.
- **Done.** Aquí se encuentran todas las tarjetas que ya han sido finalizadas. Una vez que todas las tarjetas de una misma historia se encuentran en esta columna, pasarán automáticamente a la columna de *Review*.
- **Review.** En esta columna se encuentran todas las tarjetas pertenecientes a una historia de usuario que se ha terminado. Esperaremos a realizar la *Sprint Review* (Sección 2.1.3) para poder mover cualquier tarjeta de esta columna.
- **Closed.** Una vez damos por finalizado el sprint, todas las tarjetas asociadas a historias de usuario terminadas y validadas en la *Sprint Review* (Sección 2.1.3) pasarán a esta columna y de cerrarán.

Para las tarjetas, se ha decidido seguir la siguiente guía para poder identificar cada tarjeta de forma más eficiente. El título será de la forma “[*numeroHU*] *nombreHU*”. Además cada una de ellas tendrá asociada 4 tarjetas a mayores, para especificar las distintas fases de dicha historia:

- “[*númeroHU* - *AN*] *nombreHU*”. Tarea de análisis
- “[*númeroHU* - *DIS*] *nombreHU*”. Tarea de diseño
- “[*númeroHU* - *IMP*] *nombreHU*”. Tarea de implementación
- “[*númeroHU* - *TEST*] *nombreHU*”. Tarea de testing

También puede haber tareas en las que solo se realice documentación. El título de estas tareas será de la forma “[*DOC*]”.

Todas estas tareas se explicarán más adelante en el Capítulo 7.

En las Figuras 3.1 y 3.2 podemos ver el tablero utilizado durante el desarrollo del proyecto.

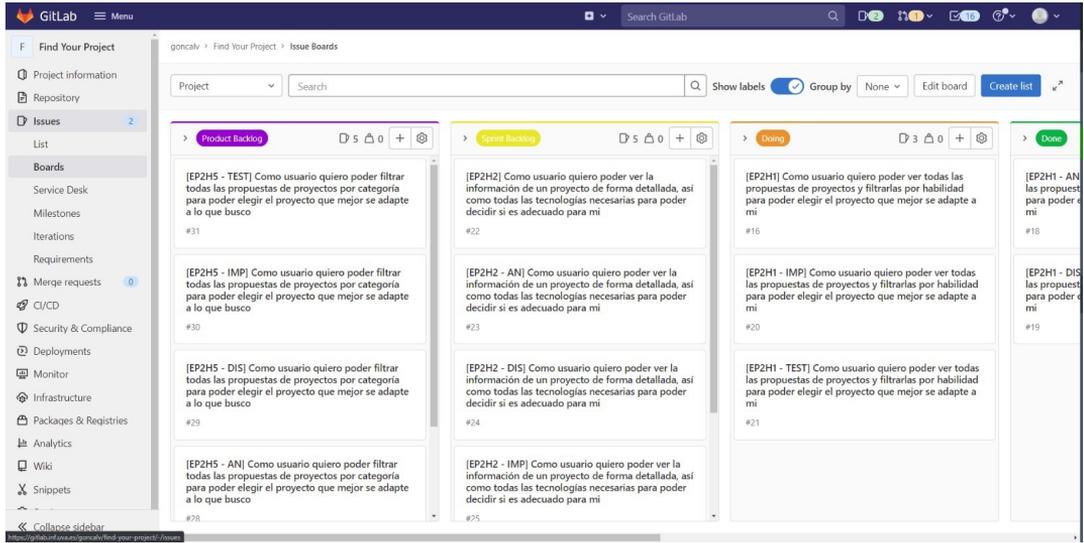


Figura 3.1: Tablero *GitLab Issue Board* (parte 1)

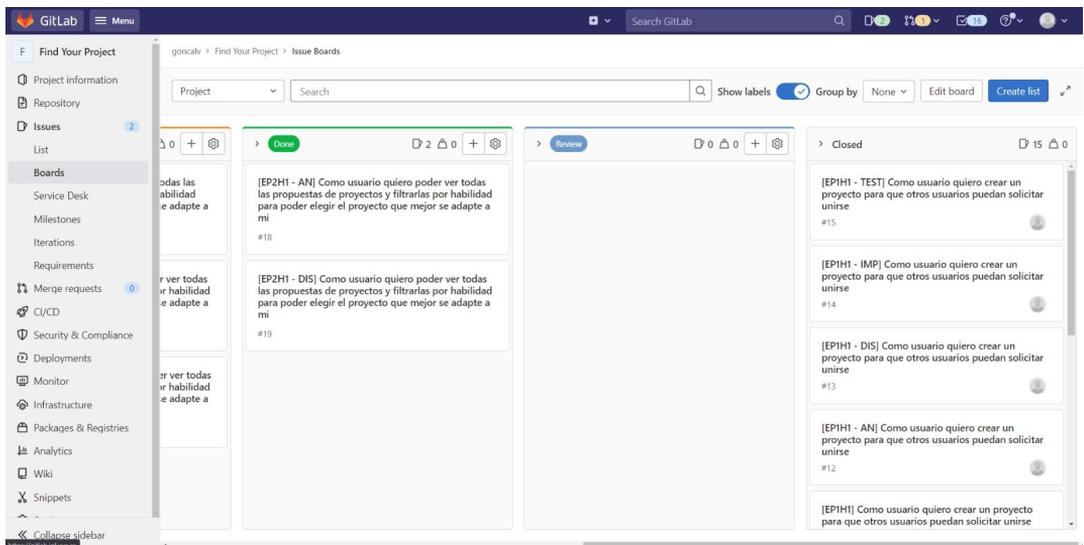


Figura 3.2: Tablero *GitLab Issue Board* (parte 2)

3.2. Android

Android [49] es un sistema operativo móvil propiedad de *Google*. Está basado en el kernel de Linux y está orientado principalmente a dispositivos móviles.

La API utilizada para este proyecto de Android es la **API 31**, ya que es la última estable en el momento del inicio del desarrollo del proyecto, pero el sistema operativo mínimo que debe tener el dispositivo para poder ejecutar nuestra app es **Android 9 (Pie)**, es decir, la API 28. Esto es debido a que algunas librerías solo pueden ser utilizadas con dispositivos que tengan una API mayor o igual a la 28.

Como lenguaje de programación se ha utilizado **Kotlin**, exactamente con el **JDK 11** de Java.

Como entorno de desarrollo se ha utilizado **Android Studio** [2] en su versión 2021.1, su última versión estable.

Además de todo esto, se han utilizado diferentes componentes y estilos. Los más importantes, serán detallados a continuación.

3.2.1. Material Design

Material Design [5] es un sistema adaptable de guías de estilo, componentes y herramientas que utiliza las mejores prácticas de diseño de interfaces de usuario (IU). En este proyecto se ha utilizado para seguir buenas prácticas a la hora de diseñar todas las interfaces de la aplicación.

3.2.2. Fragments

Tanto las *Activities* como los *Fragments* son los componentes principales de la lógica de una app Android. Una **Activity** [4] es una pantalla de la app. En general, una app Android está formada por varias pantallas, por lo que suele tener varias, aparte de la actividad principal con la que se arranca la app. Un **Fragment** [3] es una parte de interfaz de usuario de la aplicación. Tiene que estar siempre dentro de una actividad y, aunque tengan ciclos de vida distintos, el ciclo de vida de la actividad afecta al del fragmento. Además, una actividad puede contener varios fragmentos, y estos a su vez pueden ser utilizados en varias actividades, consiguiendo con esto modularizar la IU.

En este proyecto se va a utilizar *single-activity architecture* [11], recomendada por *Google*. Siguiendo esta arquitectura, la app tendrá una única actividad que servirá como contenedor para todos los fragmentos que mostrarán las distintas pantallas de la aplicación. Para ello, vamos a utilizar el componente *Navigation*, explicado a continuación.

3.2.3. Navigation

Navigation [6] permite implementar la navegación entre los diferentes fragmentos de la aplicación, integrando también la navegación con la barra superior, el menú lateral desplegable y el menú inferior de la app.

Este componente está formado por tres partes:

- **Gráfico de navegación.** Es un recurso XML que representa un grafo con todos los posibles caminos que el usuario puede tomar dentro de la app. Se muestra un ejemplo en la Figura 3.3
- ***NavHost*.** Contenedor vacío dentro de una actividad donde se mostrarán los fragmentos hacia los que se navegue.
- ***NavController*.** Objeto que gestiona la navegación dentro del *NavHost*

3.3. Firebase

Firebase [17] es una plataforma que integra varias herramientas, todas ellas pensadas para el desarrollo y despliegue de aplicaciones web y móvil. Esta plataforma es propiedad de *Google*, lo que hace que la integración con las aplicaciones Android sea perfecta. El objetivo de *Firebase* es ofrecer una plataforma con servicios de backend, análisis y monitorización sin necesidad de administrar servidores, y teniendo todo lo necesario en un mismo sitio.

A continuación, detallaremos los servicios que se han utilizado para el desarrollo de nuestra aplicación en las siguientes subsecciones.

3.3.1. Realtime Database

Realtime Database [19] es una base de datos NoSQL alojada en la nube. Los datos se guardan en forma de un gran árbol JSON y se sincronizan en tiempo real con los clientes. Además, no está pensada para consultas muy complejas y la latencia que tiene es muy baja, por lo que se adapta perfectamente al proyecto.

3.3.2. Authentication

Authentication [18] es el servicio de autenticación de *Firebase*. Tiene distintas funciones, pero las principales son la gestión de usuarios y sesiones. Permite autenticación mediante diferentes métodos y proveedores, pero en esta aplicación utilizaremos la autenticación mediante *GitHub*.

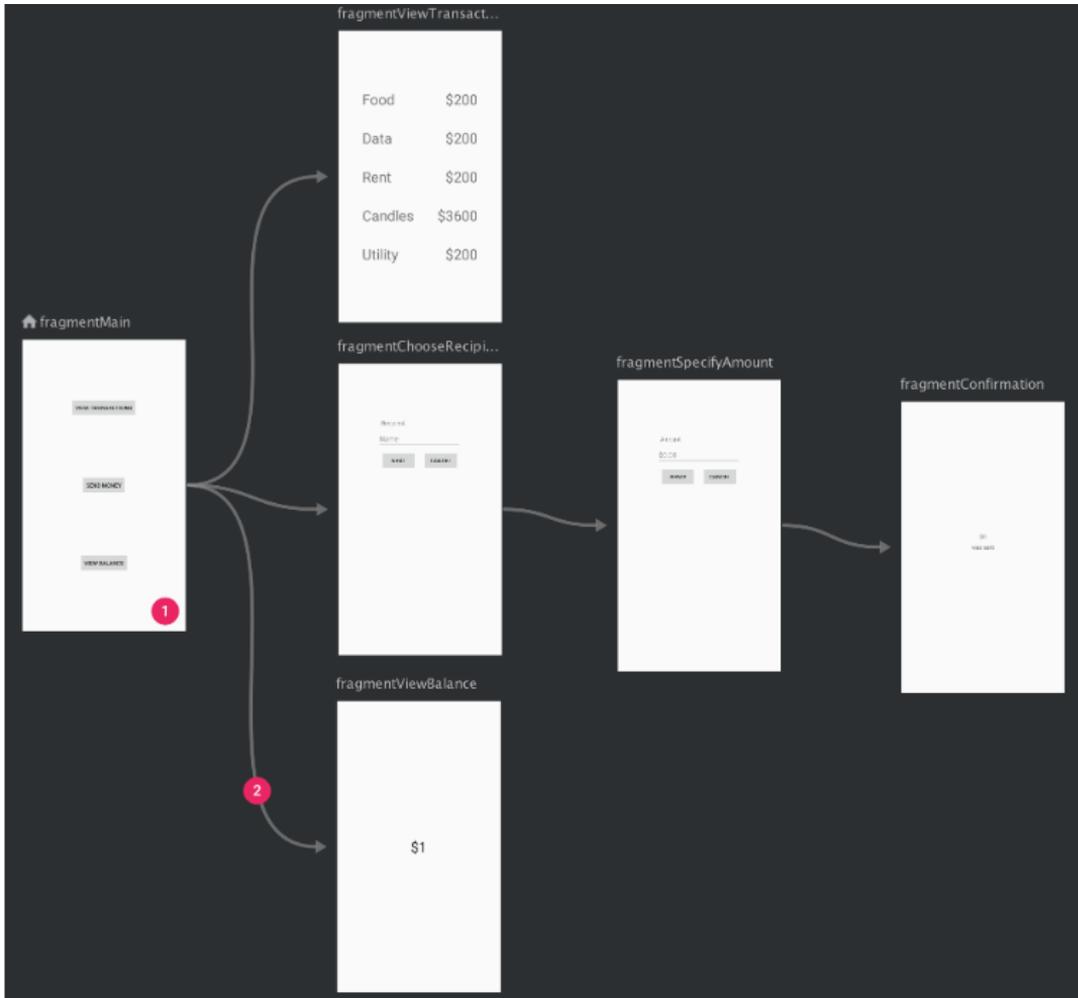


Figura 3.3: Ejemplo de Gráfico de navegación

3.4. Retrofit

Retrofit [43] es un cliente que permite realizar peticiones a APIs REST para Android. Las distintas peticiones que se pueden realizar son *GET*, *POST*, *PUT*, *PATCH*, *DELETE* y *HEAD*, y se pueden gestionar los distintos parámetros, encabezados y cuerpo de las mismas. Además traduce automáticamente los objetos devueltos al tipo de datos que se indique, como puede ser un POJO Kotlin en nuestro caso.

En este proyecto, lo utilizaremos para consumir la API de Github y poder hacer las llamadas necesarias a esta.

3.5. Herramientas para el análisis, diseño y documentación

3.5.1. Astah Professional

Astah [9] es una herramienta de modelado, utilizada para realizar todos los diagramas y modelos de las distintas actividades de análisis y diseño de la aplicación. En este proyecto hemos utilizado la versión ***Astah Professional 8.3.0***. La licencia ha sido cedida por la Escuela de Ingeniería Informática de la Universidad de Valladolid.

3.5.2. Figma

Figma [16] es una herramienta de prototipado y diseño vectorial. En este proyecto, ha sido utilizada para la creación de los bocetos de las vistas que posteriormente se crearán. Figma es gratis, por lo que no se ha tenido que utilizar ninguna licencia.

3.5.3. Overleaf

Overleaf [41] es un editor online de LaTeX colaborativo, en el que se ha desarrollado el documento actual. Almacena un historial del documento y permite edición de varios usuarios de forma simultánea.

3.6. Herramientas para la comunicación

3.6.1. Microsoft Teams

Microsoft Teams [39] es una plataforma desarrollada por *Microsoft* orientada al trabajo en equipo y la colaboración. Ofrece muchas características, como pueden ser canales de chat,

3.6. HERRAMIENTAS PARA LA COMUNICACIÓN

videoconferencia y compartición de archivos. Se ha utilizado principalmente para contactar con la tutora y tener allí un control de los archivos del proyecto.

Capítulo 4

Análisis

Desde el inicio del proyecto en el sprint 0 hasta el último sprint, se han ido llevando a cabo actividades de análisis, completando toda la información que mostraremos a continuación.

4.1. Requisitos funcionales como historias de usuario

Como se explicó en el Capítulo 2 en la Tabla 2.5, ya se ha realizado una división de las épicas en diferentes historias de usuario. Estas historias de usuario hacen referencia a los distintos requisitos funcionales del proyecto, por lo que no se volverán a mencionar en este capítulo.

4.2. Requisitos de información como historias de usuario

Aunque los requisitos de información forman parte de los requisitos funcionales, hemos decidido tratarlos por separado, para obtener el modelo de dominio inicial a partir de estos.

Podemos observarlos en la Tabla 4.1

4.3. Requisitos no funcionales como historias de usuario

Al igual que existen los requisitos funcionales, en todo proyecto también hay requisitos no funcionales. Para la obtención de dichos requisitos se ha seguido la guía FURPS [50].

En la Tabla 4.2 aparecen todos los requisitos no funcionales en forma de historia.

Historia
Como equipo de desarrollo quiero guardar los siguientes datos de un proyecto: nombre, descripción, organizador, categoría, fecha de creación y habilidades que se buscan, para manejar la información básica de la app
Como equipo de desarrollo quiero guardar los siguientes datos de un usuario: id de usuario, nombre de usuario, imagen de perfil, descripción del perfil, notificaciones activadas o no y habilidades, para manejar la información básica de la app
Como equipo de desarrollo quiero guardar los siguientes datos de una habilidad: nombre, lista de usuario pendientes y lista de usuarios asignados, para manejar la información básica de la app

Tabla 4.1: Requisitos de información

Número	Historia
RNF01	Como usuario quiero que la aplicación de gestión de proyectos se materialice en una app Android para tenerla siempre a mano en mi dispositivo móvil
RNF02	Como usuario quiero que la aplicación tenga un diseño atractivo y que permita al usuario realizar todas las acciones de manera intuitiva, para que todos los usuarios puedan utilizarla sin problemas
RNF03	Como usuario quiero recibir actualizaciones de los datos de la app en tiempo real y con una latencia menor a 1 segundo, para que todas las tareas se puedan realizar de forma fluida

Tabla 4.2: Requisitos no funcionales

4.4. Restricciones como historias de usuario

En este proyecto se han decidido incluir algunas restricciones basadas en los objetivos a cumplir. Podemos verlas en la Tabla 4.3

4.5. Modelo de dominio Inicial

A partir de los requisitos funcionales de información mencionados en la Tabla 4.1 hemos elaborado el modelo de dominio inicial, el cual está representado por el diagrama de clases de la Figura 4.1.

4.6. Modelo de proceso de Negocio

Como parte final del análisis del sistema, hemos realizado un diagrama de actividad que muestra el principal proceso de negocio que llevará a cabo nuestra aplicación. Puede verse

Historia
Como usuario quiero que la app sea gratuita para poder utilizarla libremente
Como equipo de desarrollo quiero distribuir la app a través de <i>Google Play</i> , para poder hacerla llegar todos los usuarios de forma simple, tal y como está especificado en los requisitos
Como usuario quiero que mis datos sean tratados acordes al RPGD y poder garantizar su privacidad

Tabla 4.3: Restricciones

en la Figura 4.2

Como podemos observar, en este proceso de negocio están implicados dos actores: el **organizador** de un proyecto y un **usuario** que pretende formar parte del proyecto. El proceso es comenzado por el organizador, quien crea el proyecto con los parámetros que desee (Nombre del proyecto, Descripción, Categoría en la que enmarcarlo y la lista de las habilidades necesarias). Una vez creado, el usuario podrá ver toda la información del proyecto y solicitar unirse a este, escogiendo la habilidad o habilidades que más encajen con su perfil y conocimientos. A continuación, el organizador podrá ver todas las solicitudes asociadas a cada habilidad, pudiendo realizar tres acciones para cada una de estas solicitudes: aceptar, rechazar o ignorar de momento la solicitud. Este proceso se repite hasta que todas las habilidades tienen al menos una solicitud aceptada, pudiéndose en ese momento crear el repositorio de trabajo y mandar las solicitudes de colaboración a los usuarios aceptados.

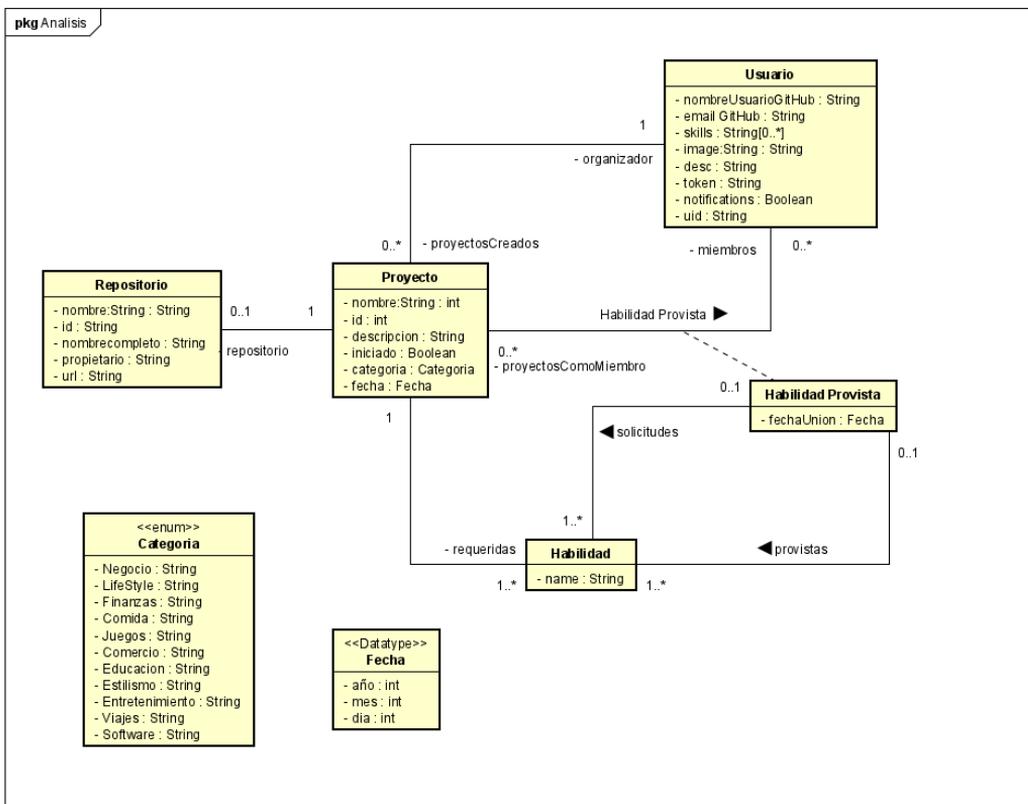


Figura 4.1: Modelo de Dominio Inicial

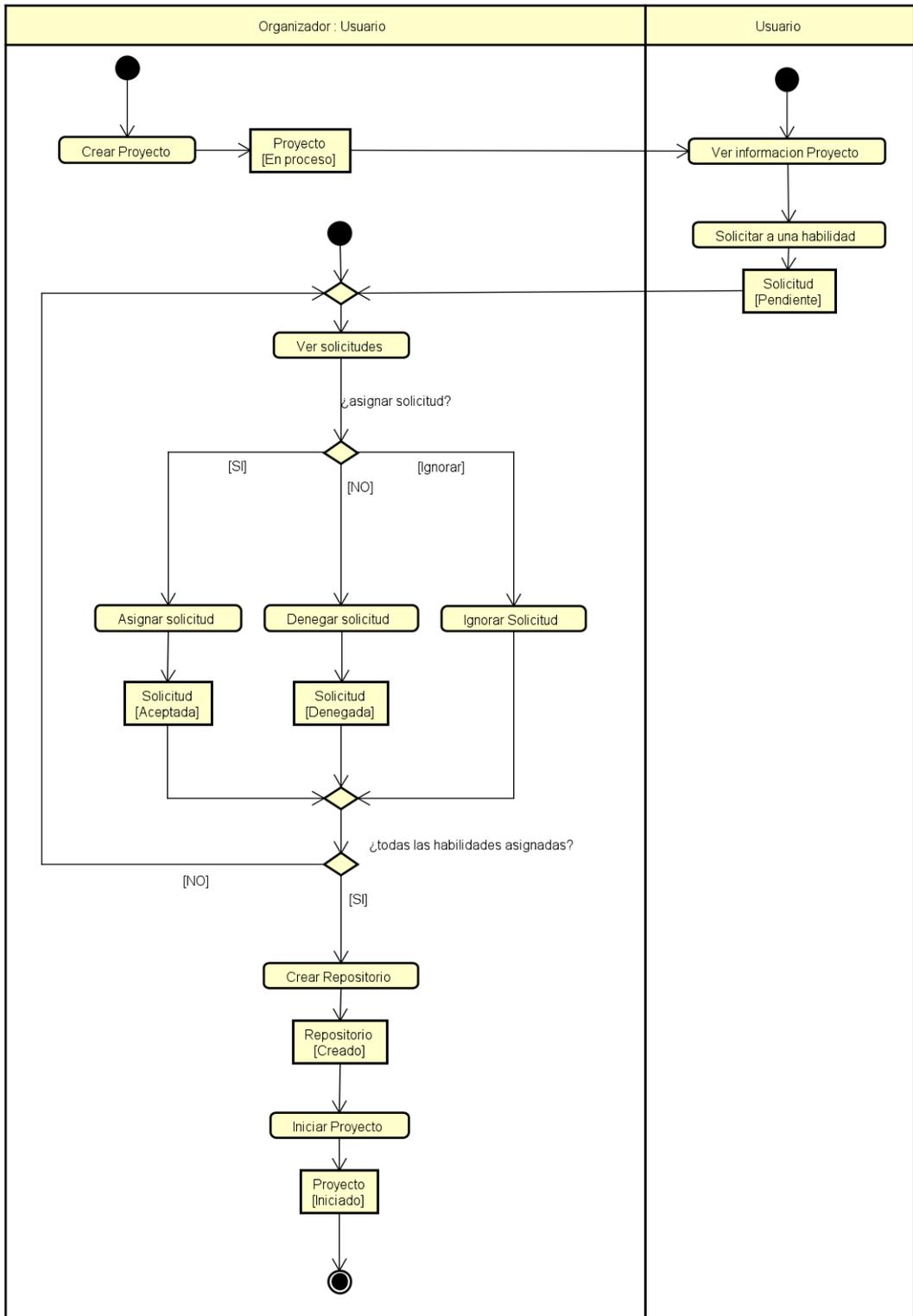


Figura 4.2: Modelo de Proceso de Negocio

Capítulo 5

Diseño

5.1. Decisiones de diseño

A continuación, se enumerarán las distintas decisiones de diseño que se han tomado durante el desarrollo de la app. Las razones de estas decisiones se mencionan en las Secciones 3.2 y 3.3.

- Se ha utilizado la **API 31** de Android.
- La interfaz de usuario seguirá las guías de diseño de *Material Design*.
- Se utilizará la arquitectura *single-activity* utilizando fragmentos, a la vez que el componente *Navigation* de Android. De esta manera conseguimos una navegación fluida con una buena experiencia de usuario.
- La base de datos utilizada será *Real Time Database* de *Firebase*.
- Para la gestión de usuarios, se ha utilizado el servicio *Authentication* de *Firebase*, utilizando Github como proveedor.

5.2. Arquitectura MVVM

Model-View-ViewModel [30] [51] o *Modelo-Vista-Vista Modelo* (MVVM) es un patrón arquitectónico de software orientado al principio de separación de responsabilidades. Su propósito es desacoplar lo máximo posible la lógica de usuario de la lógica de la aplicación. Este patrón deriva del patrón *Model-View-Controller* o Modelo Vista Controlador (MVC), muy conocido y utilizado.

Este patrón está formado por tres partes:

- **Modelo.** Representa la parte de la lógica de negocio de la aplicación, es decir, la información con la que trabajamos.
- **Vista.** Muestra la información de modelo por pantalla mediante la interfaz de usuario. También captura todos los eventos del usuario con la interfaz.
- **Vista Modelo.** Es el intermediario entre el Modelo y la vista. Contiene toda la lógica de la aplicación.

Una de las particularidades de este patrón es el uso de *data-binding*. Esto es un enlace entre la Vista y la Vista Modelo, de forma que cuando haya algún cambio en la Vista Modelo, se notifique a la Vista para que esta se actualice de forma automática. Todo esto se puede llevar a cabo gracias al patrón *Observador* que explicaremos en la Sección 5.3.1. A continuación, en la Figura 5.1 [30] podemos ver en un esquema lo explicado en este apartado .

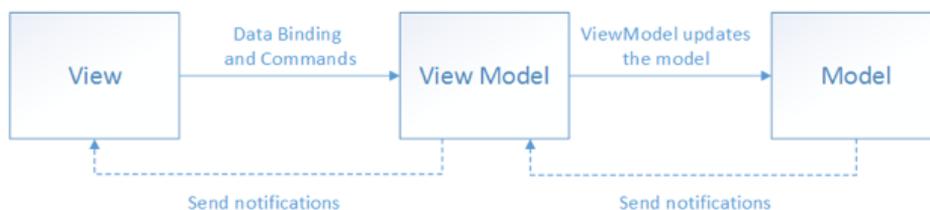


Figura 5.1: Relaciones entre partes en el patrón arquitectónico MVVM [30] (No es notación UML).

5.3. Patrones de diseño utilizados

A lo largo del desarrollo del proyecto, se han utilizado diferentes patrones de diseño para resolver los diferentes problemas que se han presentado. En esta sección explicaremos en detalle dichos patrones

5.3.1. Patrón Observador

El patrón **Observador** [40] es un patrón de diseño que permite observar los cambios producidos por un objeto. Su funcionamiento es el siguiente: cuando un objeto cambia, se le notifica al observador de dicho objeto que el estado del objeto que estaba observando ha cambiado. Una vez que el observador ha recibido la notificación, reacciona a ella actualizando la información necesaria.

Este patrón es el utilizado por la arquitectura *MVVM* y su *data-binding* para actualizar la información entre el Modelo y la Vista de manera automática.

En la Figura 5.2 podemos ver la estructura de este patrón.

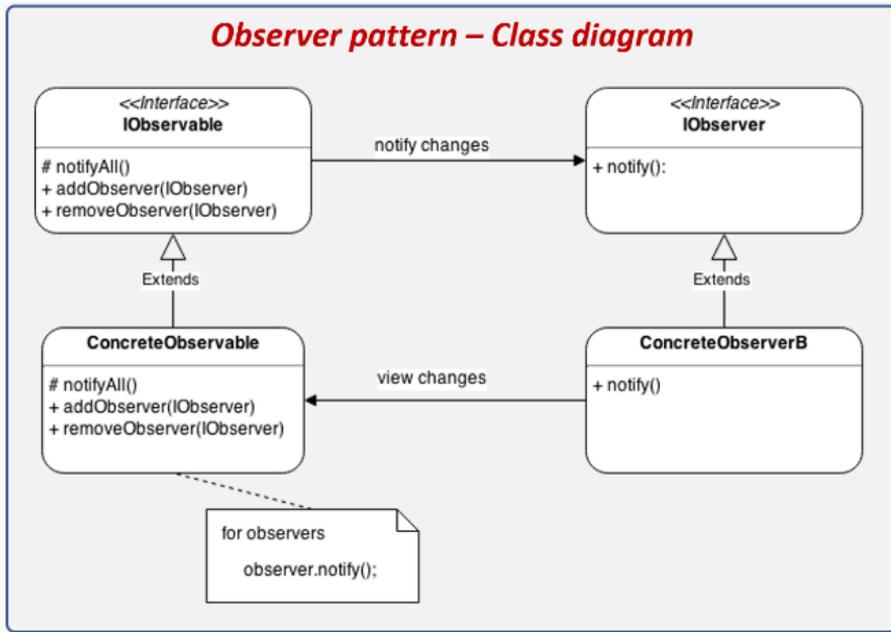


Figura 5.2: Estructura Patrón Observador [40]

5.3.2. Método Factoría

El **método Factoría** [52] es un patrón de diseño creacional. El objetivo de este patrón es definir una interfaz para crear un objeto dejando que sean las subclases las que decidan qué tipo concreto instanciar. En la Figura 5.3 podemos ver el esquema de este patrón.

En Android, todos los ViewModels se deben crear a través de la clase **ViewModelProvider**, mediante la cual solo se pueden crear ViewModels sin parámetros. Para solucionar esto, la solución que ofrece Android consiste en la creación de una subclase que herede de ViewModelProvider.Factory. En esta clase, se creará el correspondiente ViewModel con sus parámetros mediante un método factoría. Esta clase será la que se le pase al ViewModelProvider, encargándose Android de instanciar el ViewModel con los parámetros indicados.

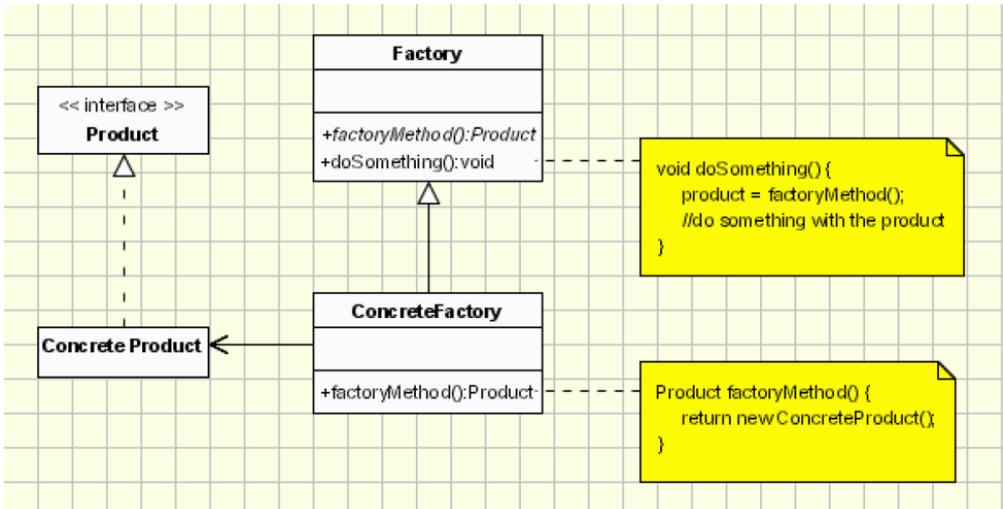


Figura 5.3: Estructura Método Factoría [52]

5.4. Diseño de la interfaz de usuario

5.4.1. Decisiones de diseño de la interfaz de usuario

En este apartado, comentaremos todas las decisiones de diseño que se han tomado a la hora de crear la interfaz de usuario:

- Se utilizarán los siguientes colores para el diseño de la aplicación:
 - Azul Claro: #90C4E9
 - Azul Oscuro: #8095CE
 - Violeta: #C58ADE
 - Morado: #8869A5
 - Añil: #AAB7E1
- Todos los botones y objetos tendrán bordes redondeados
- En caso de que un botón pueda estar o no activo, se seleccionará el color Violeta para el estado activo, y el Morado para el estado inactivo
- Respecto a las solicitudes a proyectos, se mostrará de distinta forma el botón situado a la derecha del nombre de la habilidad correspondiente en función del estado de la misma:

- Color **Violeta** - **Icono Hoja con un símbolo +** : Se puede aplicar a esa habilidad
 - Color **Violeta** - **Icono Interrogación:** Solicitud pendiente de aprobación
 - Color **Verde** - **Icono Tick:** Solicitud aceptada
 - Color **Violeta** - **Icono Lista:** Ver solicitudes (solo para los organizadores)
- El color primario de la aplicación será el **Azul Claro**, utilizándose siempre que se quiera designar alguna acción importante.
 - Todos los botones que conlleven una acción de borrar o eliminar, tendrán el fondo de color **Rojo**.

5.4.2. Mockups

En esta sección se incluirán los *mockups* o prototipos que muestran un boceto de las distintas pantallas de la aplicación.

Entre las Figuras 5.4 a 5.11 se muestran los *mockups* que se han realizado con la herramienta Figma (ver Sección 3.5.2).

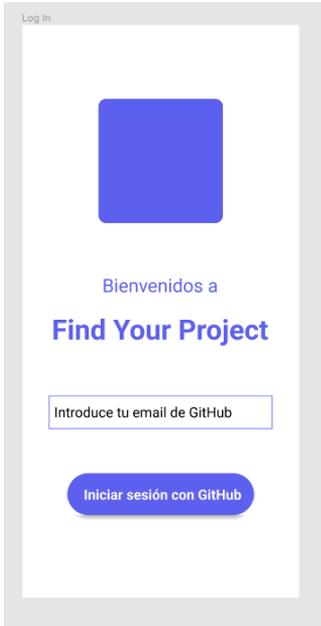


Figura 5.4: Vista LogIn



Figura 5.5: Vista Proyectos

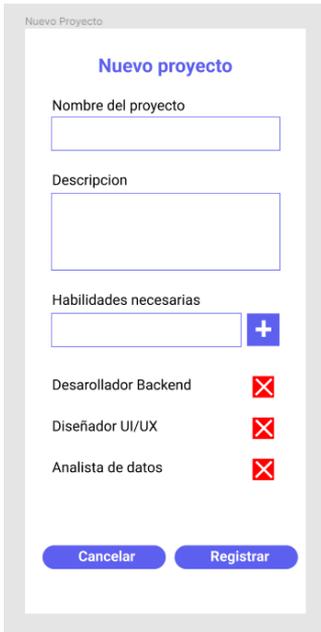


Figura 5.6: Vista Nuevo Proyecto

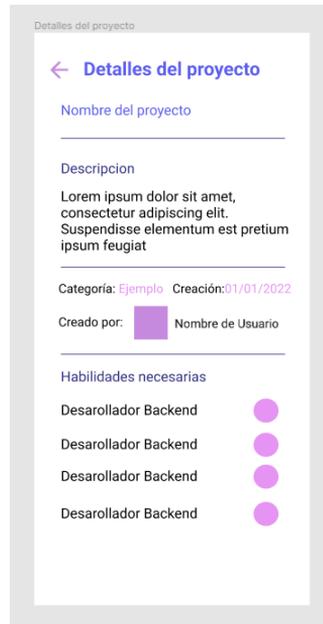


Figura 5.7: Vista Detalles del Proyecto

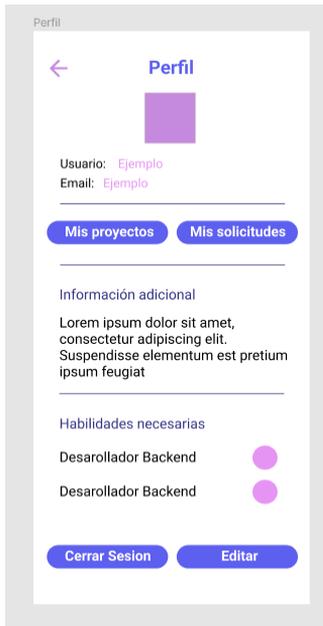


Figura 5.8: Vista Información Perfil

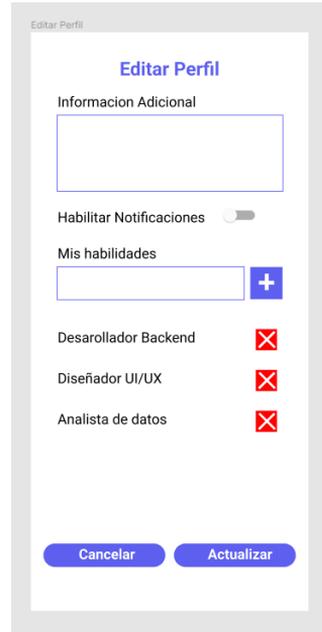


Figura 5.9: Vista Editar Perfil



Figura 5.10: Vista solicitudes

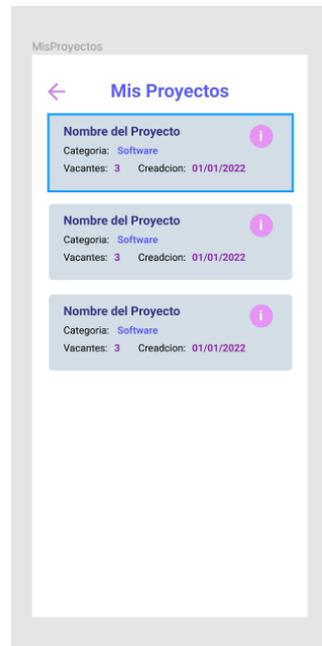


Figura 5.11: Vista Mis proyectos

5.5. Diseño Arquitectónico

La arquitectura general de la aplicación está basada en 4 paquetes, estando algunos de ellos formado por subpaquetes. Todos se encuentran incluidos dentro del paquete `com.goncalv.findyourproject`, y son los siguientes:

- **ui:** en este paquete encontramos las clases relacionadas con la interfaz de usuario y corresponde a la **Vista** del patrón **MVVM**. Está dividido en diferentes subpaquetes:
 - **activities:** aquí se encuentran las clases *Activity*. Al tratarse de una aplicación que sigue la arquitectura *single-activity*, solo existe una actividad principal. Aunque para reducir la complejidad y los problemas se decidió utilizar otra actividad solo para el login.
 - **fragments:** aquí podemos encontrar todas las clases *Fragment* o fragmentos de la aplicación.
 - **base:** y en este paquete, están las herramientas creadas para mejorar los componentes *RecyclerView* de forma programática. *RecyclerView* [7] es un tipo de componente de Android que permite mostrar de manera eficiente grandes conjuntos de datos de forma dinámica.
 - **adapters:** en este paquete se encuentran todos los adaptadores que necesita la clase *RecyclerView*. Estos adaptadores sirven para proveer al *RecyclerView* de una vista que represente cada elemento que se quiere mostrar.
- **viewmodels:** en este paquete se encuentran los *ViewModel* y las factorías asociadas a estos. Corresponde con el **Modelo de Vista** del patrón **MVVM**.
- **repositories:** en este paquete se encuentran las clases repositorio que han utilizado para operar con la base de datos. Se corresponde con la parte de **Modelo** del patrón **MVVM**.
- **entities:** se corresponde con las entidades del dominio en forma *POJO* o *Plain Old Java Object*.
- **api:** en este paquete se encuentran las clases necesarias para el acceso a la API de Github.
- **listeners:** en este paquete se encuentran todos los listeners y las clases tipo interfaz definidas en nuestra aplicación. Aquí se encuentran los *listeners* de los *adapters* mencionados anteriormente y también los *listeners* que se utilizaran para conectar las llamadas a la base de datos desde los repositorios con la interfaz.

A continuación, en la Figura 5.12 se puede observar el diagrama *Decomposition&Uses style*, el cual presenta la arquitectura general de la aplicación.

Siguiendo este diagrama, en las Figuras 5.13 a 5.18 podemos ver los diagramas *Decomposition&Uses style* de cada paquete de forma individual.

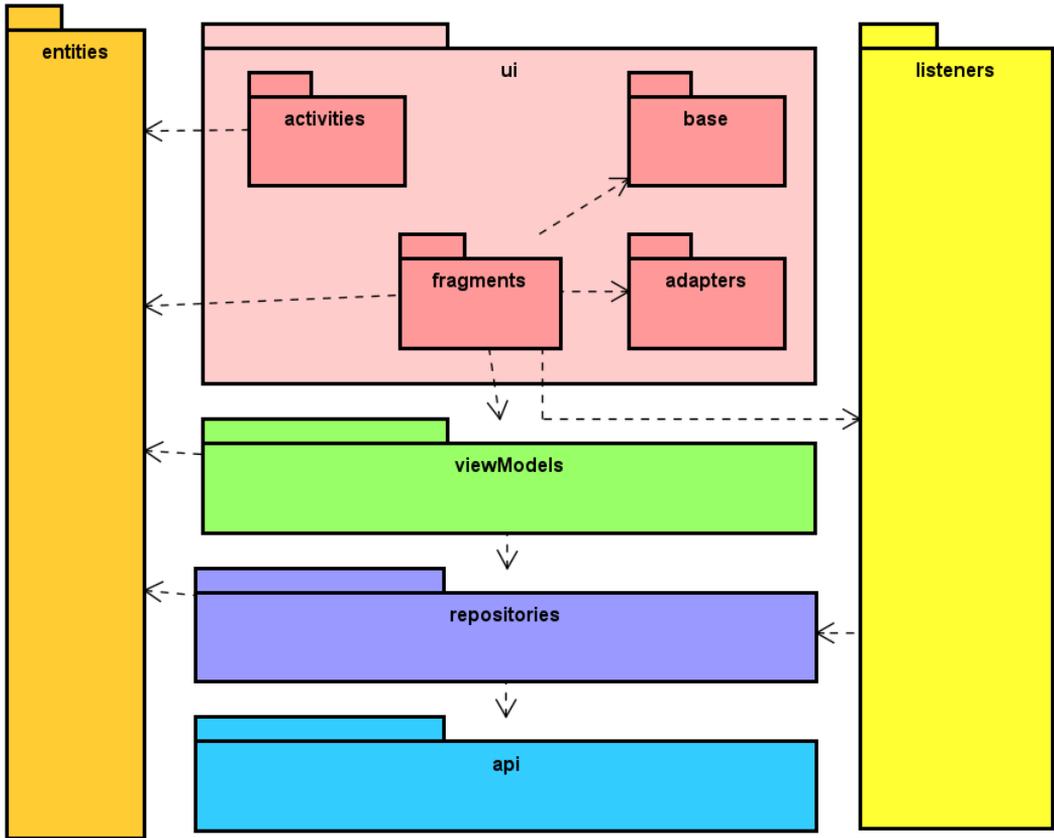


Figura 5.12: Arquitectura general de la aplicación

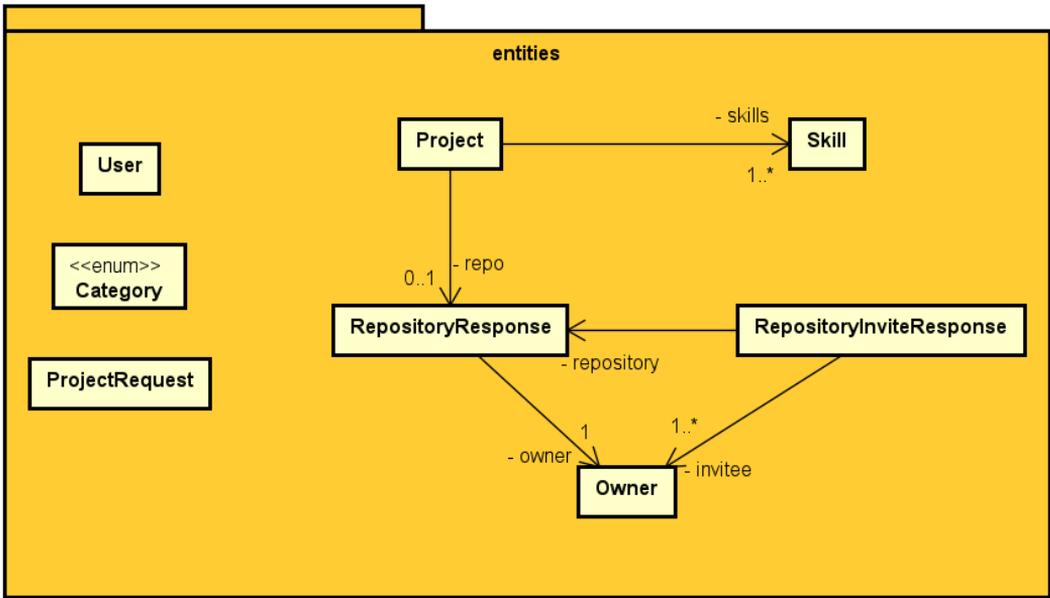


Figura 5.13: *Decomposition&Uses style* del paquete *entities*

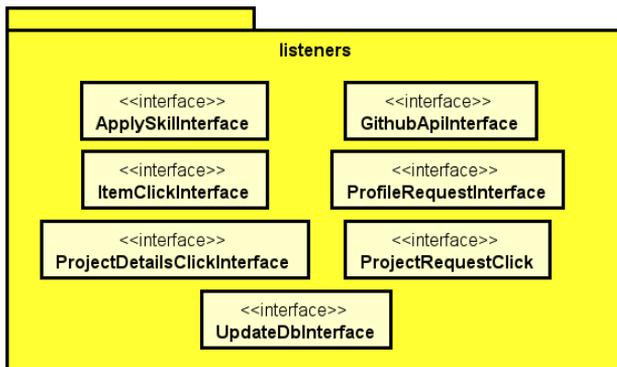


Figura 5.14: *Decomposition&Uses style* del paquete *interfaces*

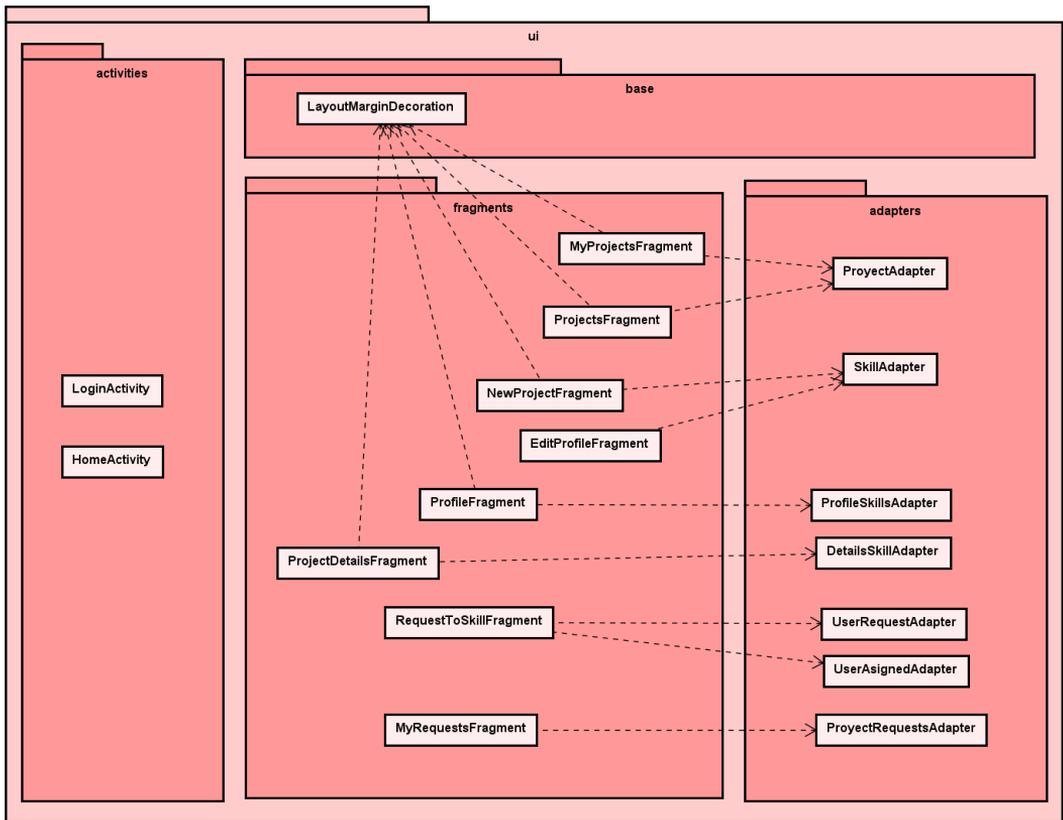


Figura 5.15: *Decomposition & Uses style* del paquete *ui*

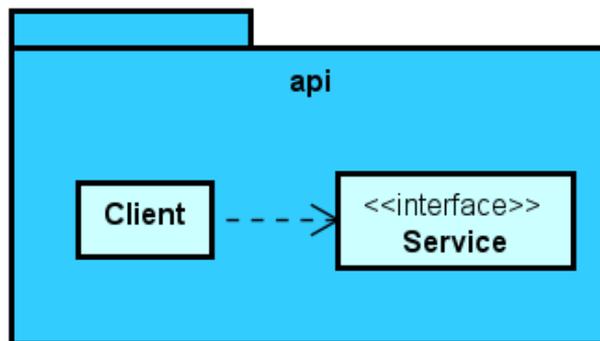


Figura 5.16: *Decomposition & Uses style* del paquete *api*

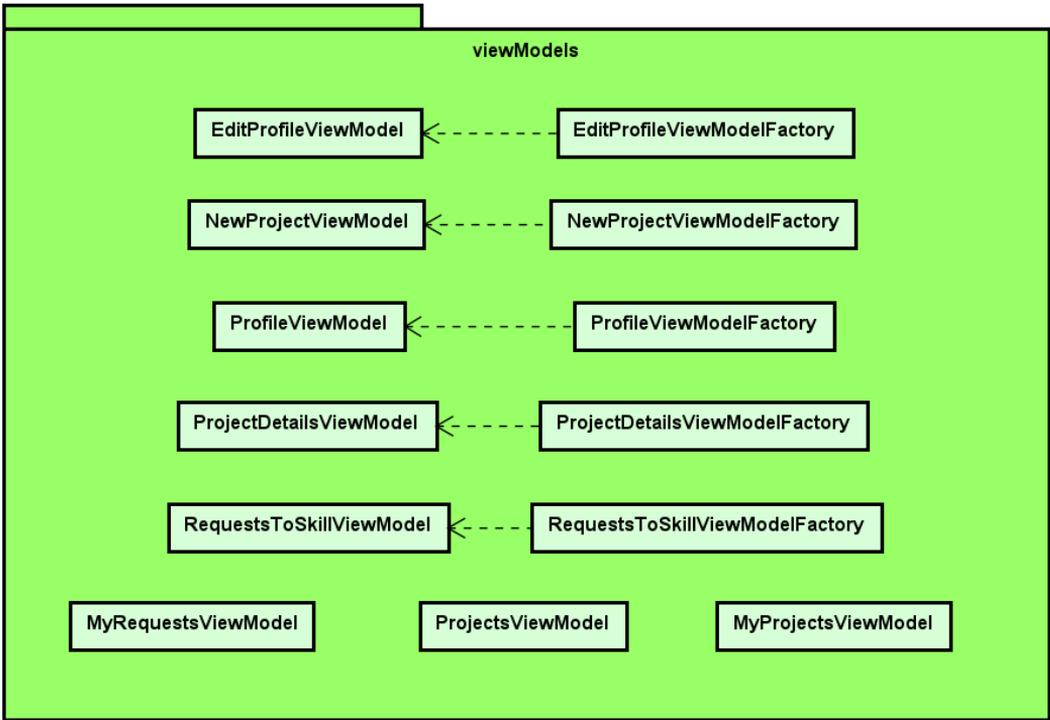


Figura 5.17: *Decomposition & Uses style* del paquete `viewModels`

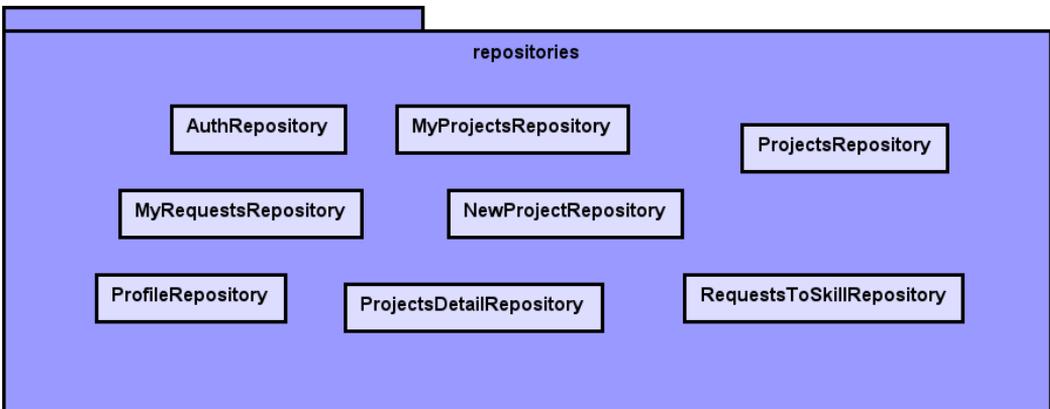


Figura 5.18: *Decomposition & Uses style* del paquete `repositories`

5.6. Diseño de la comunicación entre objetos

Para explicar el diseño de la comunicación entre objetos, se desarrollará y explicará el diagrama de secuencia de la historia de usuario **EP1H7 - Como usuario y organizador del proyecto, quiero poder eliminar una propuesta de proyecto, para poder desecharlo y notificar a los solicitantes la eliminación de dicho proyecto**. A parte de la secuencia principal, también se mostrarán y explicarán los diagramas asociados al patrón *Observador*.

El diagrama de la secuencia principal podemos verlo en la Figura 5.19. En el comienzo del diagrama, podemos ver como el actor (organizador del proyecto) realiza la acción de eliminar el proyecto en el que se encuentra. Siguiendo la arquitectura *MVVM*, la vista, en este caso *ProjectDetailsFragment* notifica al *ProjectDetailViewModel* la acción de eliminar el proyecto y este, a su vez, se lo notifica a *ProjectDetailsRepository*, que hace las veces de Modelo de la arquitectura. Una vez le llega la llamada *deleteProject()* al Modelo, este realiza la llamada a la base de datos para eliminar el proyecto. A esta llamada se le añaden dos *Listeners*, que se ejecutaran en función de si esta ha sido exitosa (*onSuccess()*) o ha fallado (*onFailure()*). En caso de que sea exitosa, se llama al método *onUpdateSucess()* de la interfaz *UpdateDbInterface*, el cual se ejecuta en *ProjectDetailsFragment()* y muestra un mensaje por pantalla de que el proyecto ha sido eliminado correctamente de la base de datos. En caso contrario, se llama al método *onUpdateFailure()* de la interfaz *UpdateDbInterface* que también se ejecuta en *ProjectDetailsFragment()* y muestra un mensaje de error por pantalla diciendo que ha habido un error al eliminar el proyecto de la base de datos.

Una vez explicado el mecanismo principal de la historia de usuario, en la Figura 5.20 vamos a explicar el funcionamiento del patrón Observador en esta historia. Como podemos ver, en el método *onCreateView()* del Fragment (método que se lanza cuando se crea por primera vez la vista), llamamos al método *getProject()* del ViewModel y lo ponemos a observar. Esto implica que cada vez que se actualice alguna información del proyecto, notificará a todos los observadores que ha habido cambios y mandará una actualización (*update()*) a cada uno de ellos. Cuando el Fragment detecta la actualización (*update()*), se ejecuta la secuencia que aparece en la Figura 5.21. Aquí, primero comprobamos si el proyecto es nulo, como sucede en este caso de uso una vez lo hemos eliminado de la base de datos. En caso de ser nulo, se cerraría la vista actual y navegaríamos a la vista principal, *ProjectsFragment*. En caso contrario, es decir, el proyecto existe y no es nulo, se muestra la información del proyecto por pantalla.

5.7. Diagrama de despliegue

El despliegue de la aplicación es muy sencillo. Solamente utilizamos una API externa, en este caso la de Github que está marcada en otro color, la cual no necesita ser desplegada por nosotros, pero que añadimos para señalar su uso. Al no depender de otros servicios backend externos a parte de los proporcionados por Firebase y la API mencionada, solo es necesario representar esos componentes.

A continuación, en la Figura 5.22 se puede observar el diagrama de despliegue de la aplicación.

5.8. Diseño de la Base de Datos

La base de datos utilizada, como ya hemos comentado antes, será *Realtime Database* de *Firebase*. Es una base de datos NoSQL y se almacena como objetos JSON. Básicamente, se trata de un gran árbol JSON. Esto produce algunos problemas a la hora de guardar relaciones a otros datos, puesto que pueden aparecer duplicación de datos y que se formen ramas hijas de forma infinita.

En este caso, hemos utilizado dos grandes Tablas para guardar la información: **proyectos** y **usuarios**. Puede verse en las Figuras 5.23 y 5.24. A continuación de estas figuras, vamos a explicar el porqué se ha decidido guardar la información de esta manera.

Como podemos observar no existe ninguna relación entre las tablas, al contrario de como suele ocurrir con las bases de datos relacionales, lo que dificultará un poco el trabajo.

5.8.1. Usuarios

Como hemos comentado anteriormente, **usuarios** es una tabla que almacena todos los usuarios de la aplicación. Cada id de usuario tiene asignada toda la información de ese “debajo”, al tratarse de un árbol JSON. Todos los campos son de tipo cadena de texto, menos las notificaciones que es booleano (verdadero o falso) y la lista de habilidades o *skills*, que es a su vez una lista de cadenas de texto.

5.8.2. Proyectos

Al igual que la tabla de usuarios, **proyectos** es una tabla que guarda todos los proyectos de la aplicación, asignando a cada id de proyecto toda la información del mismo.

Como no existe la posibilidad de usar tablas relacionadas, todos los campos asociados a algún usuario se han creado en forma de cadena de texto, guardando el uid del usuario en dichos campos, para poder acceder desde la otra tabla a esa información, como es el caso del

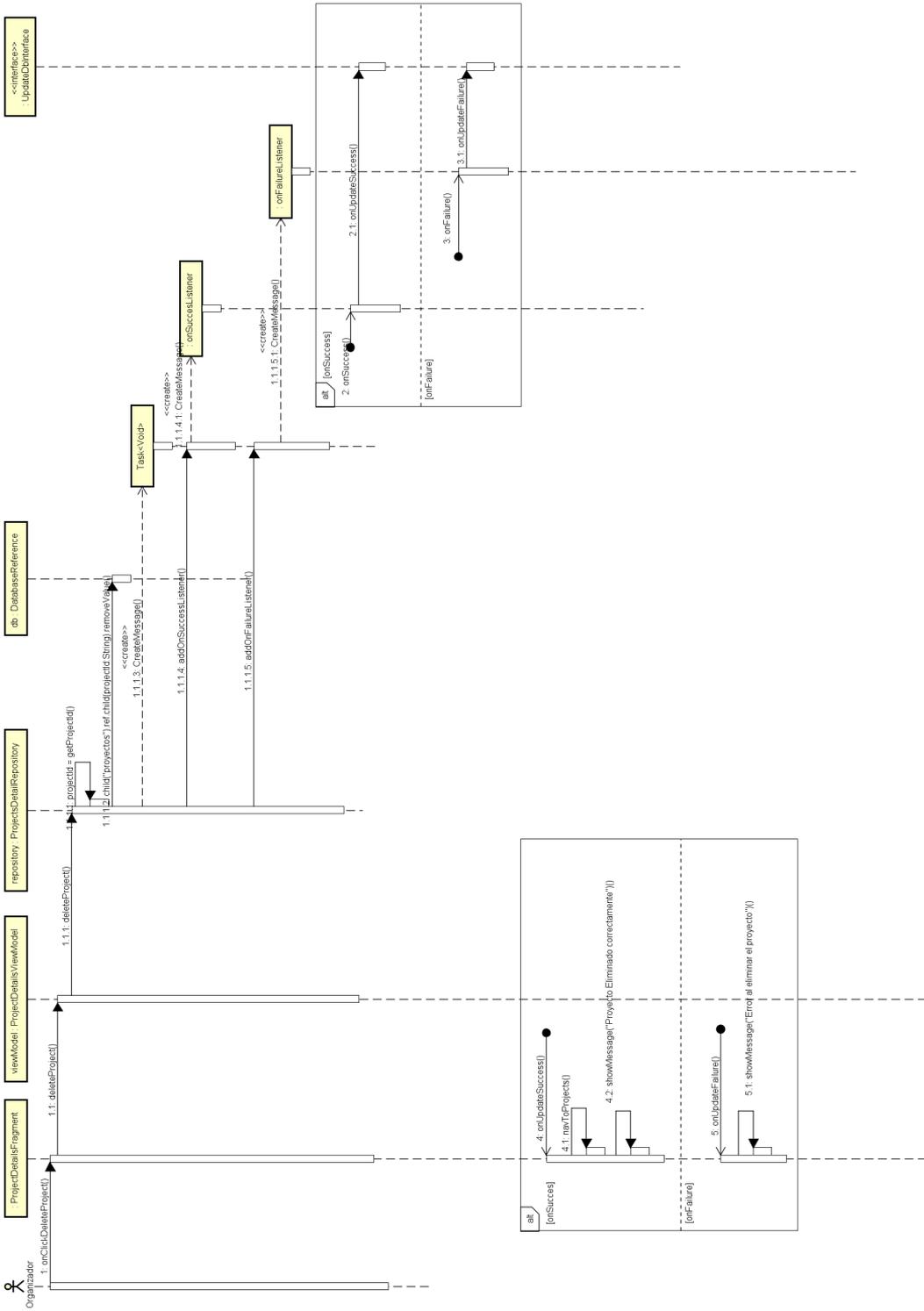


Figura 5.19: Diagrama de secuencia principal de la EP1H7

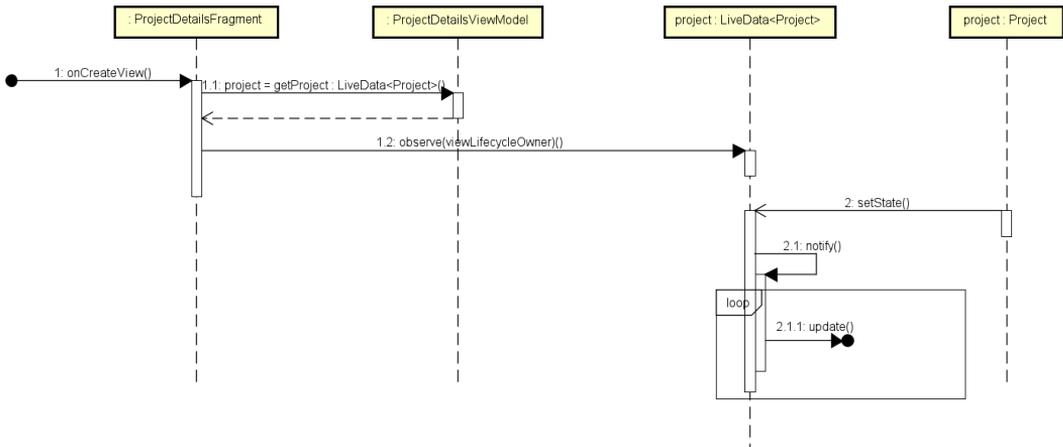


Figura 5.20: Diagrama de secuencia del patrón Observador

organizador. La parte más compleja de guardar han sido las habilidades o *skills*. Este campo es una lista de todas las habilidades necesarias para el proyecto. Cada una de las habilidades guarda su nombre y dos listas:

- **requests**, la cual es una lista con los ids de los usuarios que han solicitado para la habilidad
- **assignedTo**, la cual representa una lista con los ids de los usuarios que han sido aceptados y asignados a la habilidad correspondiente

Como tampoco existe ningún tipo específico para guardar las fechas, se ha decidido utilizar cadenas de texto para representar la fecha de creación de cada proyecto.

5.8.3. Traducción JSON a POJO por Firebase

Como he comentado anteriormente, la base de datos está formada por un gran árbol JSON. Todos los tipos de datos que se almacenan son los tipos disponibles de JSON: String, Long, Double, Boolean, Map<String, Object> y List<Object>. Por ello, la traducción del tipo Object con las clases Kotlin tipo POJO se realiza de forma prácticamente automática, teniendo que utilizar en estas solo los tipos de datos admitidos por JSON.

En el caso de nuestra base de datos, como podemos observar en las Figuras 5.23 y 5.24, solo utilizamos 3 tipos de datos: Strings o cadenas de texto; Mapas, para poder tener relacionados los IDs de usuarios y proyectos con su respectiva información; y Listas, que a su vez guardan atributos de tipo cadena de texto.

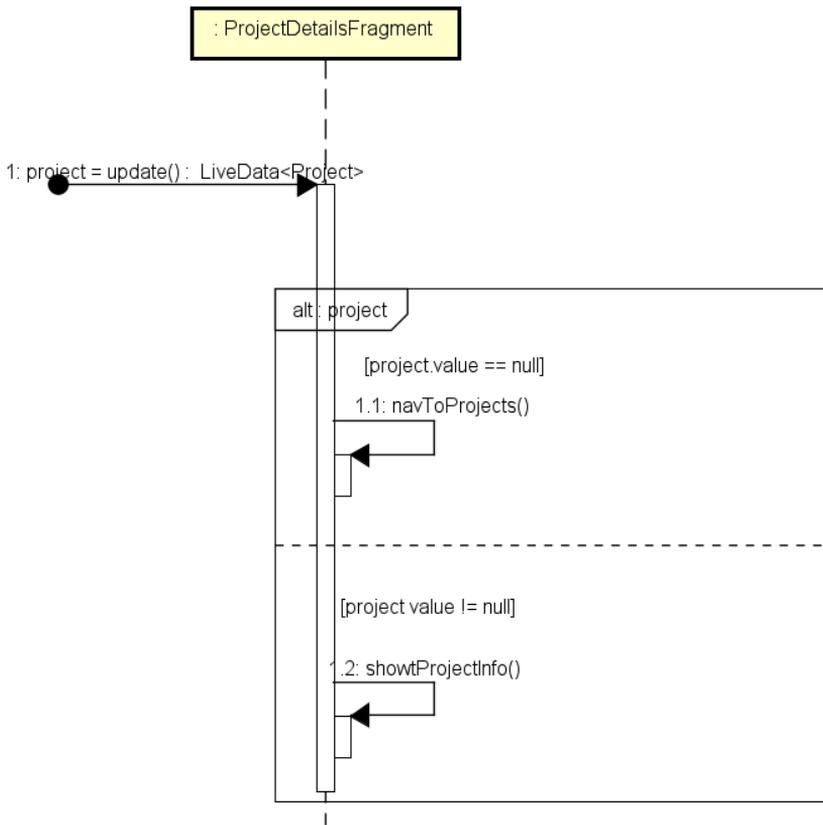


Figura 5.21: Diagrama de secuencia del patrón Observador (Continuación)

5.9. Retrofit

Como he comentado en el Capítulo 3.4, **Retrofit** [43][15] es lo que utilizaremos para realizar las llamadas a la API de Github. Aquí, voy a explicar cómo funciona.

Lo primero que creamos es un cliente, el cual será el que utilizemos en todas las peticiones que se vayan a realizar. Lo siguiente es crear una interfaz de Retrofit, la cual representa cada llamada HTTP que se realiza. Cada una de estas llamadas producirá su propio par de solicitud y respuesta HTTP. Se pueden realizar llamadas síncronas con *execute()* o asíncronas con *enqueue()*. En este proyecto, todas las llamadas son asíncronas.

Las llamadas *enqueue()* envían la petición al servidor de forma asíncrona, notificando el *Callback* (o devolución de llamada) de la respuesta en los diferentes casos (error al contactar con el servidor, en la creación de la petición o en el procesamiento de la petición). Este método *Callback* es ejecutado cuando llega una respuesta del servidor. En nuestro proyecto, diferenciamos cuando nos llega una respuesta o cuando llega un fallo. En el primer caso, se

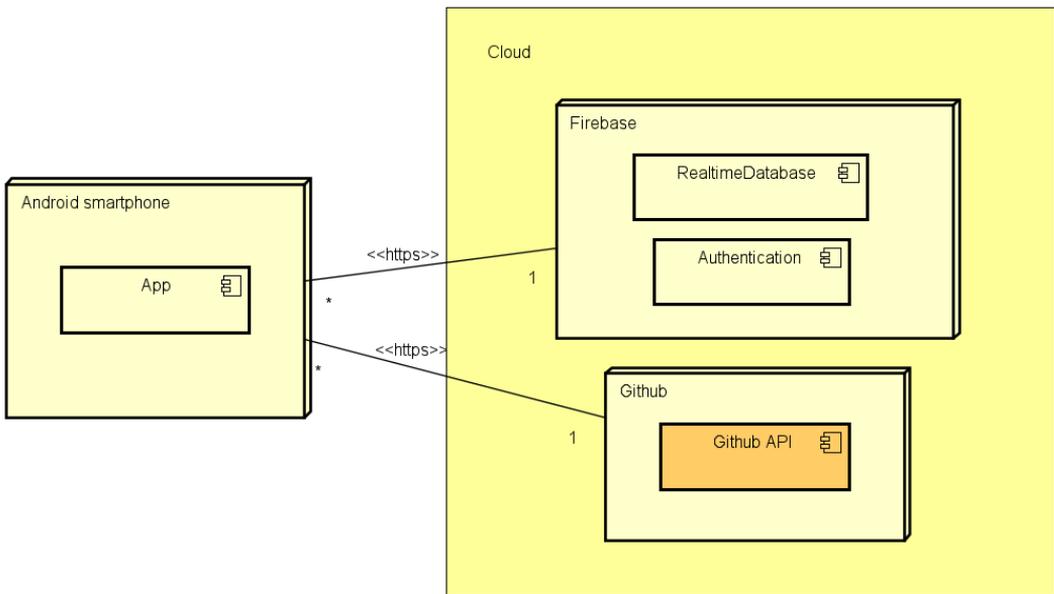


Figura 5.22: Diagrama de despliegue

comprueba que el código de respuesta sea el esperado, traduciendo a continuación el cuerpo de esta a un POJO Kotlin. En caso de que el código de respuesta no sea el esperado o sea el correspondiente a un error se notifica al usuario, permitiendo así que la aplicación siga corriendo sin dar lugar a errores.

```

"proyectos": {
  "id": {
    "category": "String",
    "creator": "uidUsuario",
    "date": "String",
    "description": "String",
    "initiated": true,
    "name": "String",
    "repo": {
      "full_name": "String",
      "html_url": "String",
      "id": 495107945,
      "name": "String",
      "node_id": "String",
      "owner": {
        "login": "username"
      }
    }
  },
  "skills": [
    {
      "requests": [
        "uidUsuario"
      ],
      "assignedTo": [
        "uidUsuario"
      ],
      "name": "String"
    }
  ],
  "vacants": 0
}

```

Figura 5.23: Estructura Lista de Proyectos en la BD

```

"usuarios": {
  "id": {
    "desc": "String",
    "email": "String",
    "image": "String",
    "notifications": false,
    "token": "String",
    "uid": "String",
    "username": "String",
    "skills": [
      "skill",
      "skill2"
    ]
  }
}

```

Figura 5.24: Estructura Lista de Usuarios en la BD

Capítulo 6

Implementación y pruebas

6.1. Implementación

6.1.1. Filtrado por habilidad

Una de las partes más importantes de la vista de Proyectos es la posibilidad de filtrar los diferentes proyectos por habilidades. Este filtrado se realiza por subcadena. Una subcadena[1] es una secuencia contigua de caracteres en una cadena. Por ejemplo, si tenemos la cadena *abc*, sus subcadenas serían: vacía, *a*, *ab*, *abc*, *b*, *bc* y *c*.

Lo que se hace es pedirle al usuario la habilidad por la que se desee filtrar. Una vez la tenemos, ejecutamos el filtrado por todos los proyectos existentes, buscando si existe alguna habilidad en el proyecto que contenga el texto a filtrar como subcadena. Si es así, se añade y se muestra por pantalla.

6.1.2. Problemas en la implementación

Durante las actividades de implementación de la aplicación me he ido encontrando con algunas dificultades. A continuación, voy a listar y describir las más importantes:

- **Desconocimiento del uso de *Navigation*.** Como antes de comenzar el proyecto nunca había trabajado con la arquitectura *single-activity* ni con *Android Navigation*, me supuso un esfuerzo acostumbrarme a utilizar esta tecnología, aunque según iba utilizándolo más, se volvía más fácil trabajar con ello.
- **Uso de la API de Github.** Nunca antes había tenido que consumir una API externa en una app Android, lo que ha supuesto un duro quebradero de cabeza. Además, una vez conocido su funcionamiento, he tenido que documentarme y aprender sobre la API

de Github y no ha sido tarea fácil por la gran cantidad de errores y tiempo que me ha consumido. Después de mucho esfuerzo, se ha conseguido utilizar como se esperaba. A continuación dejo un post que escribí en los foros de la comunidad de Github para preguntar sobre mi error: <https://github.community/t/error-trying-to-add-a-collaborator-via-api/251849> También escribí un ticket a Github explicando el fallo que estaba teniendo. Podemos ver tanto mi pregunta como su respuesta en las Figuras 6.1 y 6.2. Como resumen, las peticiones devolvían un código de error 404 que no estaba contemplado en la documentación de la API. Después de contactar con diversos usuarios y con el soporte de Github, se llegó a la conclusión de que el error estaba en el token de autorización con el que se estaban mandando las peticiones.

- **Uso de Realtime Database.** Estando siempre acostumbrado a usar bases de datos relacionales, enfrentarme a una base de datos no-relacional como es *Realtime Database* ha sido una tarea difícil. La parte más complicada ha sido tener que realizar consultas complejas teniendo que tener referencias de otra tabla, puesto que esta base de datos solo permite realizar consultas simples.
- **Uso de MVVM o observadores.** Era la primera vez que utilizaba la arquitectura MVVM por lo que me llevó un tiempo acostumbrarme a la organización de esta arquitectura y donde debía ir cada parte del código. Una vez dominado esto, tuve que enfrentarme a los observadores, que permiten a la aplicación reaccionar a los cambios de la base de datos en tiempo real. Esto no fue fácil, pero gracias a ello la aplicación reacciona muy rápido a los cambios en la IU.

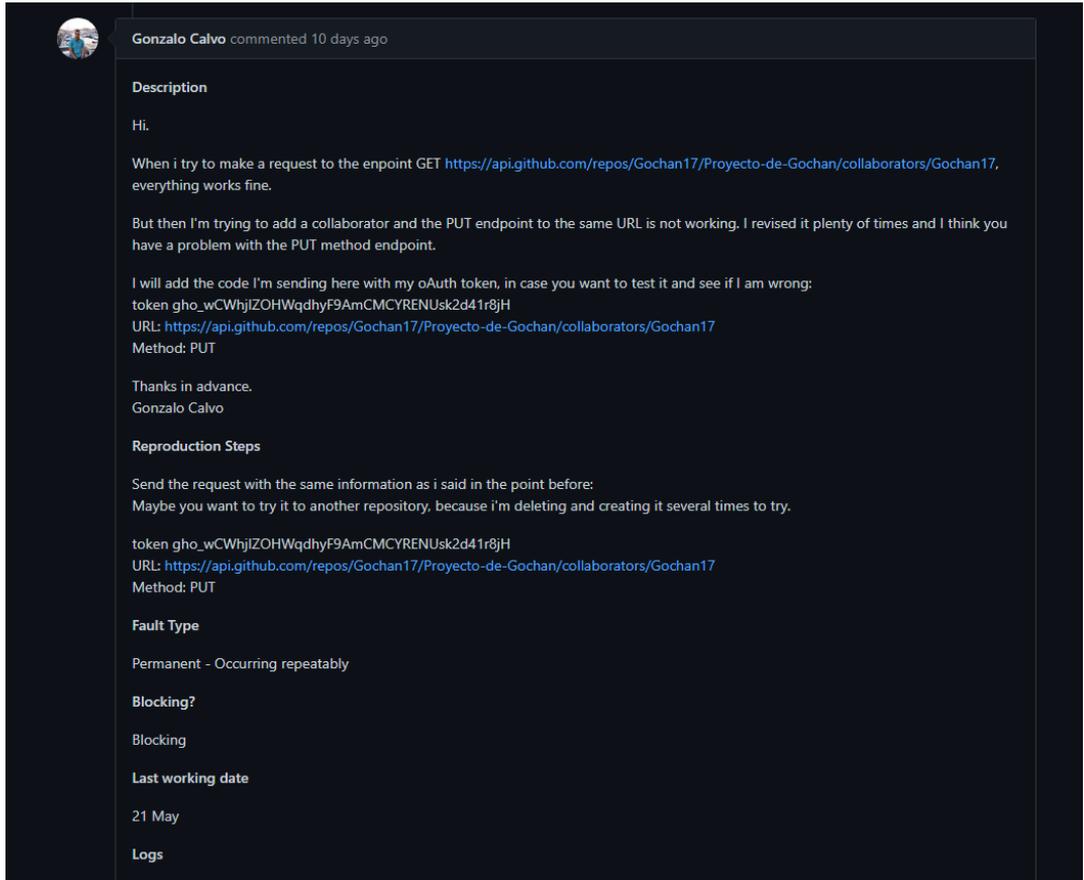


Figura 6.1: Ticket enviado a Github

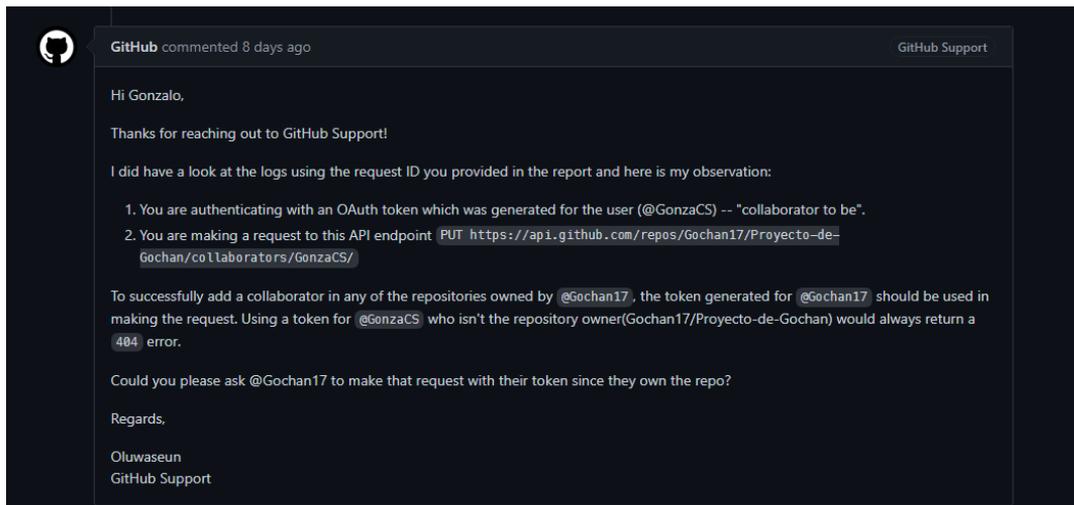


Figura 6.2: Ticket de respuesta de Github

6.2. Pruebas

En esta sección se recogen los casos de prueba realizados durante la implementación del proyecto.

6.2.1. Tipos de pruebas

Las pruebas realizadas en este proyecto han sido **Pruebas de aceptación de historias de usuario** [14]. Este tipo de pruebas se basan en verificar que el software cumple las expectativas desde el punto de vista del cliente y de los usuarios finales.

En este proyecto, se han utilizado este tipo de pruebas para comprobar que la funcionalidad implementada en cada historia era la correcta y seguía lo especificado en la planificación. Se ha realizado un seguimiento de las pruebas, que veremos vas adelante.

Además todas las pruebas de este proyecto se han realizado de forma **manual**, ya sea utilizando dispositivos físicos o virtuales (emulador de Android Studio).

6.2.2. Casos de prueba

En las Tablas 6.1 a 6.19 se muestran los casos de prueba junto al código que se le ha asignado.

CP-01	Registro
Descripción	Registrarse como nuevo usuario.
Escenario	Vista “LogIn”
Entrada	<ul style="list-style-type: none"> ■ Se introduce <i>gonzlocalvosanz@gmail.com</i> en el email de GitHub ■ Se acciona el botón de LogIn ■ Se introduce <i>xxxxxxxxxx</i> en la contraseña de GitHub ■ Se aceptan los permisos de GitHub
Resultado esperado	El usuario ha accedido a la aplicación con su nueva cuenta de usuario con email ”gonzlocalvosanz@gmail.com”
Comprobación	Accediendo a la pestaña de perfil aparece su información de GitHub (Nombre y email)

Tabla 6.1: CP-01. Registro

CP-02	Inicio de Sesión
Descripción	Inicio de sesión para un usuario registrado previamente.
Escenario	Vista “LogIn”
Entrada	<ul style="list-style-type: none"> ■ Se introduce <i>gonzlocalvosanz@gamil.com</i> en el email de GitHub ■ Se acciona el botón de LogIn ■ Se introduce <i>xxxxxxxxxx</i> en la contraseña de GitHub ■ Se aceptan los permisos de GitHub
Resultado esperado	El usuario ha accedido a la aplicación con su cuenta de usuario con email <i>gonzlocalvosanz@gmail.com</i>
Comprobación	Accediendo a la pestaña de perfil aparece su información de GitHub (Nombre y email)

Tabla 6.2: CP-02. Inicio de sesión

CP-03	Cerrar sesión
Descripción	Cerrar sesión en la aplicación.
Escenario	Vista "Profile"
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón de cerrar sesión
Resultado esperado	Se ha cerrado la sesión y se ha vuelto a la pantalla de "LogIn"
Comprobación	Se encuentra en la pantalla de LogIn)

Tabla 6.3: CP-03. Cerrar sesión

CP-04	Crear un proyecto
Descripción	Crear un proyecto para que otros usuarios puedan solicitar unirse
Escenario	Vista "NewProject"
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se introduce <i>Proyecto para desarrollo de nuevas tecnologías software</i> en el nombre del proyecto ■ Se introduce <i>En este proyecto vamos a tratar de conseguir desarrollar nuevas tecnologías y frameworks para el desarrollo software front y back</i> en la descripción del proyecto ■ Se selecciona la categoría <i>Software</i> ■ Se introduce <i>Desarrollador Nativo Android</i> como habilidad necesaria. ■ Se acciona el botón de Registrar
Resultado esperado	Se ha creado un proyecto con nombre <i>Proyecto para desarrollo de nuevas tecnologías software</i> y aparece en la vista de "Proyectos"
Comprobación	El nuevo proyecto con nombre <i>Proyecto para desarrollo de nuevas tecnologías software</i> se ha añadido a la Base de datos. (Posteriormente se podrá encontrar en la pestaña de proyectos)

Tabla 6.4: CP-04. Crear un proyecto

CP-05	Ver Filtrar Proyectos por categoría
Descripción	Filtrar todos los proyectos de la lista para que solo aparezcan los proyectos de la categoría determinada
Escenario	Vista “Home”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ▪ Se acciona el botón de filtro ▪ Se selecciona la categoría <i>Negocio</i>
Resultado esperado	Solo aparecen los proyectos cuya categoría es <i>Negocio</i>
Comprobación	Todos los proyectos que aparecen deben pertenecer a la categoría <i>Negocio</i>

Tabla 6.5: CP-05. Filtrar Proyectos por categoría

CP-06	Filtrar Proyectos por habilidad
Descripción	Filtrar todos los proyectos de la lista para que solo aparezcan los proyectos cuyas habilidades contengan las palabras buscadas
Escenario	Vista “Home”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ▪ Se acciona el botón de búsqueda ▪ Se escribe <i>Android</i> en el campo de texto
Resultado esperado	Solo aparecen los proyectos cuyas habilidades contienen la palabra “Android”
Comprobación	Todos los proyectos que aparecen deben tener una Habilidad necesaria que contenga la palabra <i>Android</i>

Tabla 6.6: CP-06. Filtrar Proyectos por habilidad

CP-07	Ver detalles de proyecto
Descripción	Ver la información detallada de un proyecto
Escenario	Vista “Home”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ▪ Se acciona el botón de detalle del proyecto con nombre <i>Proyecto para desarrollo de nuevas tecnologías software</i>
Resultado esperado	Se navega a la pantalla de Detalles del proyecto con nombre <i>Proyecto para desarrollo de nuevas tecnologías software</i>
Comprobación	Aparece en pantalla la información de un proyecto cuyo nombre es <i>Proyecto para desarrollo de nuevas tecnologías software</i>

Tabla 6.7: CP-07. Ver detalles de proyecto

CP-08	Ver información de perfil
Descripción	Ver la información de perfil del usuario
Escenario	Vista “Profile”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ▪ Se acciona el botón de <i>Perfil</i> situado en la barra de Navegación
Resultado esperado	Se navega a la pantalla de Perfil y se muestra la información del usuario
Comprobación	Aparece en pantalla la información del usuario (Usuario e email de GitHub e información adicional de la aplicación)

Tabla 6.8: CP-08. Ver información de perfil

CP-09	Editar Perfil
Descripción	Editar la información de perfil del usuario
Escenario	Vista “Profile”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón de <i>Editar</i>, el cual esta situado en la parte inferior derecha y está marcado con un lápiz ■ Se introduce <i>Estudiante de Ingeniería Informática. Desarrollador Android en Practicas. Ponente en Cursos de Kotlin y Jetpack Compose</i> en el campo Información adicional. ■ Se habilitan las notificaciones ■ Se añaden <i>Desarrollo Android, Kotlin, Java y Desarrollo Mobile</i> como Mis Habilidades ■ Se acciona el botón que dice <i>Actualizar</i>
Resultado esperado	Se navega a la pantalla de Perfil y se muestra la información del usuario actualizada.
Comprobación	Aparece en pantalla la información del usuario actualizada con la Información Adicional nueva(<i>Estudiante de Ingeniería Informática. Desarrollador Android en Practicas. Ponente en Cursos de Kotlin y Jetpack Compose</i>), y la lista de Mis Habilidades actualizada mostrando <i>Desarrollo Android, Kotlin, Java y Desarrollo Mobile</i>

Tabla 6.9: CP-09. Editar Perfil

CP-11	Solicitar unirse a proyecto
Descripción	Solicitar unirse a un proyecto en la habilidad escogida
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón a la derecha de la habilidad <i>Skill1</i> del proyecto con nombre <i>Proyecto de Prueba</i> ■ Se selecciona la opción <i>Si</i> en la ventana de confirmación
Resultado esperado	Se ha mandado nuestra solicitud a la habilidad <i>Skill1</i> del proyecto con nombre <i>Proyecto de Prueba</i>
Comprobación	En la base de datos, accediendo al proyecto con nombre <i>Proyecto de Prueba</i> , en el apartado <i>Skills</i> accedemos a la skill con nombre <i>Skill1</i> . En la lista <i>requests</i> debe aparecer nuestro uid de usuario

Tabla 6.10: CP-10. Solicitar unirse a proyecto

CP-11	Editar Proyecto
Descripción	Editar la información y habilidades de un proyecto siendo el administrador
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón azul con el icono de un pincel situado en la parte inferior derecha. El nombre del proyecto es <i>Proyecto de Prueba</i> ■ Se elimina la habilidad con nombre <i>Skill2</i> ■ Se acciona el botón que dice <i>Actualizar</i> para guardar la nueva información del proyecto
Resultado esperado	Se ha actualizado la información del proyecto.
Comprobación	Se ha navegado a la vista de detalles del proyecto y ahora ya no aparece la habilidad con nombre <i>Skill2</i>

Tabla 6.11: CP-11. Editar Proyecto

CP-12	Eliminar Proyecto
Descripción	Eliminar un proyecto ya creado
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón rojo con el icono de una papelera situado en la parte inferior izquierda de la pantalla del proyecto con nombre <i>Proyecto de Prueba</i> ■ Se selecciona <i>Si</i> en la ventana de confirmación mostrada
Resultado esperado	Se ha eliminado el proyecto con nombre <i>Proyecto de prueba</i>
Comprobación	Ya no aparece el proyecto con nombre <i>Proyecto de prueba</i> entre el resto de proyectos

Tabla 6.12: CP-12. Eliminar proyecto

CP-13	Ver información de perfil de un solicitante
Descripción	Ver la información de perfil de un solicitante a una de las habilidades del proyecto
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón morado a la derecha de la habilidad <i>Skill1</i>. Nos encontramos en el proyecto con nombre <i>Proyecto de Prueba</i> ■ Se acciona el botón con el icono de información del perfil al lado del usuario con nombre <i>Gochan17</i>.
Resultado esperado	Se muestra la pantalla con la información de perfil de <i>Gochan17</i>
Comprobación	Se ha navegado a la pantalla de Perfil, y la información mostrada es la del usuario con nombre <i>Gochan17</i>

Tabla 6.13: CP-13. Ver información de perfil de un solicitante

CP-14	Aceptar solicitante
Descripción	Aceptar un solicitante a una de las habilidades de un proyecto
Escenario	Vista "Project Details"
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón morado a la derecha de la habilidad <i>Skill1</i>. Nos encontramos en el proyecto con nombre <i>Proyecto de Prueba</i> ■ Se mantiene presionado el cuadro que representa el perfil de <i>Gochan17</i> ■ Se acciona el botón que dice <i>Aceptar</i> que se muestra en la parte inferior derecha.
Resultado esperado	Se acepta el usuario con nombre <i>Gochan17</i> para la habilidad con nombre <i>Skill1</i>
Comprobación	Ha desaparecido el usuario con nombre <i>Gochan17</i> de la lista de solicitantes y aparece en la vista de aceptados.

Tabla 6.14: CP-14. Aceptar solicitantes

CP-15	Ver mis solicitudes
Descripción	Ver mis solicitudes a proyectos
Escenario	Vista "Profile"
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón que dice <i>Mis solicitudes</i>
Resultado esperado	Se muestran todas las solicitudes realizadas por el usuario
Comprobación	Se ha navegado a la vista que muestra las solicitudes del perfil

Tabla 6.15: CP-15. Ver mis solicitudes

CP-16	Cancelar solicitud
Descripción	Cancelar solicitud de unión a un proyecto
Escenario	Vista “My Requests”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se mantiene presionado unos segundo sobre la solicitud a eliminar ■ Se acciona el botón que dice <i>Eliminar</i> que aparece debajo de la solicitud
Resultado esperado	Se ha eliminado la propuesta a la habilidad <i>Skill1</i> del proyecto con nombre <i>Proyecto de Prueba</i>
Comprobación	Ya no aparece la solicitud en la vista “Mis Solicitudes”

Tabla 6.16: CP-16. Ver mis solicitudes

CP-17	Ver mis proyectos
Descripción	Ver mis proyectos creados e iniciados
Escenario	Vista “Profile”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el boton que dice <i>Mis Proyectos</i>
Resultado esperado	Se muestran los proyectos creados e iniciados por el usuario
Comprobación	Todos los proyectos que se muestran tienen como organizador a el usuario

Tabla 6.17: CP-17. Ver mis proyectos

CP-18	Crear Repositorio
Descripción	Crear Repositorio con los miembros del equipo inicial
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón que dice <i>Crear Repositorio</i>
Resultado esperado	Se ha creado el repositorio y se han mandado las solicitudes de colaboradores a los miembros del equipo inicial
Comprobación	Se ha creado el nuevo repositorio en Github en el perfil del usuario y en colaboradores aparecen enviadas las solicitudes al resto del equipo.

Tabla 6.18: CP-18. Crear Repositorio

CP-19	Iniciar Proyecto
Descripción	Iniciar Proyecto y Añadir información repositorio para el resto de usuarios
Escenario	Vista “Project Details”
Precondición	Estar autenticado en la aplicación
Entrada	<ul style="list-style-type: none"> ■ Se acciona el botón que dice <i>Iniciar</i>
Resultado esperado	Se ha iniciado el proyecto y se muestra la información del repositorio
Comprobación	Aparece la información del repositorio en la vista donde se encuentra el usuario

Tabla 6.19: CP-19. Iniciar Proyecto

6.2.3. Resultados de las pruebas

En esta sección, se recogen los resultados de las pruebas realizadas, al igual que los fallos encontrados y sus soluciones. Todo esto se resume en la Tabla 6.20.

6.3. Copyright

Como creador, quiero que el código de este proyecto sea privado. Por ello, el código se alojará en un repositorio privado al que solo tendrá acceso el creador del proyecto. Además, no se le aplicará ninguna licencia. Por tanto, se aplica automáticamente la ley de Copyright [42].

Caso de Prueba	Resultado	Solución
CP-01	OK	-
CP-02	OK	-
CP-03	OK	-
CP-04	OK	-
CP-05	OK	-
CP-06	OK	-
CP-07	OK	-
CP-08	OK	-
CP-09	OK	-
CP-10	OK	-
CP-11	Fallo: se duplican las habilidades después de editar un proyecto	Solución: Se limpia la lista de habilidades antes de editar un proyecto
CP-12	OK	-
CP-13	Fallo: se muestran las opciones de usuario (editar y cerrar sesión) cuando se visita otro perfil	Solución: se han deshabilitado dichos campos cuando el perfil que se muestra no es el del usuario autenticado
CP-14	OK	-
CP-15	OK	-
CP-16	OK	-
CP-17	OK	-
CP-18	Fallo: no se envían las invitaciones a los colaboradores	Solución: Si da error en las invitaciones, se solicita que se renueve el token de usuario
CP-19	OK	-

Tabla 6.20: Resultados y soluciones de los casos de prueba

Solamente se permitirá ver el código a los miembro del tribunal, y únicamente durante el período de evaluación del trabajo de fin de grado.

Capítulo 7

Seguimiento del proyecto

7.1. Introducción

En este Capítulo se presentará el seguimiento del proyecto realizado sprint a sprint. Cada sprint está formado por un número indeterminado de historias de usuario. La información de cada sprint, se organizará en forma de tabla con distintas columnas de datos: **Historia de usuario**, que hace referencia a las historias realizadas en el sprint; **Tareas**, que listará las actividades a realizar para dicha historia; **Tiempo estimado**, el cual hace referencia al tiempo estimado en el evento *Sprint Planning* (Sección 2.1.3) en función de los puntos de historia asignados; **Tiempo empleado**, que registrará el tiempo real dedicado a dicha historia; y por último **Estado**, que mostrará el estado en el que se encuentra la historia entre los siguientes: No comenzado, En progreso y Finalizado

Cada historia de usuario se ha dividido en Tareas:

- **Análisis - AN.** En esta tarea se comprobará si es necesario modificar el modelo de dominio inicial para el correcto desarrollo de la historia.
- **Diseño - DIS.** En esta tarea se añadirán y completarán los diagramas existentes con los nuevos detalles que se introduzcan en el desarrollo de la historia.
- **Implementación - IMP.** En esta tarea codificaremos el diseño creado, añadiendo a la aplicación un aumento de funcionalidad, siendo esta la descrita en la historia de usuario.
- **Testing - TEST.** En esta tarea se probará la funcionalidad implementada en la historia.

Se utilizarán las etiquetas que aparecen a la derecha del nombre de las tareas para referirse a estas más adelante. Además de estas tareas, como se ha comentado anteriormente pueden aparecer tareas de Documentación, para la que se utilizará la etiqueta **DOC**.

7.2. Seguimiento de los sprints realizados

7.2.1. Sprint 0 (01/02/22 - 17/02/22)

Como hemos comentado anteriormente, este sprint es distinto al resto. Su duración es distinta a la del resto de sprints, y ha sido de un total de 16 horas.

En este sprint, se han realizado las siguientes tareas:

- **Elaboración del *Product Backlog* inicial.** A partir de la idea inicial del proyecto, se creó la primera versión del *Product Backlog*, escribiendo todos los objetivos de la aplicación como épicas, y dividiendo estas a su vez en historias de usuario.
- **Calendarización inicial.** Se creó el calendario con las fechas de inicio y fin de cada sprint, así como eventos importantes de (eventos Scrum, vacaciones, límite de entregas, etc). Este calendario se puede ver en la Tabla 2.1.
- **Plan de riesgos.** Se realizó el plan de riesgos mostrado en la Sección 2.6.
- **Plan de presupuestos.** Para terminar con la planificación inicial, se creó el plan de presupuestos mostrado en la Sección 2.7
- **Formación y búsqueda de información inicial.** Para poder comenzar con el desarrollo de forma más rápida, fluida y cuanto antes, se ha dedicado tiempo a la búsqueda de información sobre la arquitectura *MVVM* (Model - View - ViewModel) [51], al igual que sobre la forma de navegación que se utilizará en la aplicación, *Jetpack Navigation* [6].

La duración total de este sprint ha sido de **20 horas**.

7.2.2. Sprint 1 (17/02/22 - 03/03/22)

En este primer sprint se da por comenzado el desarrollo del proyecto y sus historias de usuario.

En la Tabla 7.1, se pueden ver las tareas que se han llevado a cabo en el sprint 1.

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP3H1	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	12 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP3H4	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	5 horas	2 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP1H1	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	25 horas	18 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
Tiempos Totales		40 horas	32 horas	Todas las tareas finalizadas correctamente

Tabla 7.1: Tareas Sprint 1

En total, el tiempo empleado en el sprint fue de **32 horas**.

7.2.3. Sprint 2 (03/03/22 - 17/03/22)

En la Tabla 7.2, se pueden ver las tareas que se han llevado a cabo en el sprint 2.

7.2. SEGUIMIENTO DE LOS SPRINTS REALIZADOS

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP2H1	<ul style="list-style-type: none"> ■ AN ■ DIS ■ IMP ■ TEST 	15 horas	18 horas	<ul style="list-style-type: none"> ■ AN - Finalizado ■ DIS - Finalizado ■ IMP - Finalizado ■ TEST - Finalizado
EP2H2	<ul style="list-style-type: none"> ■ AN ■ DIS ■ IMP ■ TEST 	10 horas	12 horas	<ul style="list-style-type: none"> ■ AN - Finalizado ■ DIS - Finalizado ■ IMP - Finalizado ■ TEST - Finalizado
EP2H5	<ul style="list-style-type: none"> ■ AN ■ DIS ■ IMP ■ TEST 	10 horas	3 horas	<ul style="list-style-type: none"> ■ AN - Finalizado ■ DIS - Finalizado ■ IMP - Finalizado ■ TEST - Finalizado
Tiempos Totales		35 horas	33 horas	Todas las tareas finalizadas correctamente

Tabla 7.2: Tareas Sprint 2

En total, el tiempo empleado en el sprint fue de **33 horas**.

En la *Retrospective Meeting*, se ha recopilado la siguiente información de cara a futuros sprints:

- Las interacciones con la base de datos *Firabase Realtime Database* han supuesto una mayor carga de trabajo de la esperada.
- La dificultad del cambio de fondo en las interacciones entre los distintos botones de filtrado supusieron el cambio de utilizar Botones de Acción Flotantes a Botones con Imagen normales, modificando en estos el atributo *background*.
- La historia EP2H5 se realizó en un tiempo mucho menor del estimado ya que se realizó un buen trabajo implementando la historia EP2H1, pudiendo reutilizar gran parte del trabajo de esta.

7.2.4. Sprint 3 (17/03/22 - 31/03/22)

En la Tabla 7.3, se pueden ver las tareas que se han llevado a cabo en el sprint 3.

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP3H2	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	8 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP3H5	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	20 horas	20 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
Tiempos Totales		30 horas	28 horas	Todas las tareas finalizadas correctamente

Tabla 7.3: Tareas Sprint 3

Además de todo esto, también se avanzó una cantidad de **3 horas** en la memoria.

En total, el tiempo empleado en el sprint fue de **31 horas**.

7.2.5. Sprint 4 (31/03/22 - 14/03/22)

En la Tabla 7.4, se pueden ver las tareas que se han llevado a cabo en el sprint 4.

7.2. SEGUIMIENTO DE LOS SPRINTS REALIZADOS

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP2H3	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	15 horas	20 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP1H6	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	10 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado (menos notificaciones) ▪ TEST - Finalizado
EP1H7	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	8 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado (menos notificaciones) ▪ TEST - Finalizado
Tiempos Totales		35 horas	38 horas	Todas las tareas finalizadas correctamente menos las tareas referentes a las notificaciones

Tabla 7.4: Tareas Sprint 4

En total, el tiempo empleado en el sprint fue de **38 horas**.

La historia **EP2H3** supuso más tiempo del esperado para llevarla a cabo. Hubo que hacer una refactorización en la forma en la que estaba estructurada la base de datos, lo que hizo que casi todos los datos de proyectos que había guardados fueran inútiles y encadenaran gran cantidad de errores. El problema quedó solucionado y se pudo continuar con el desarrollo.

En la *Retrospective Meeting*, se ha recopilado la siguiente información de cara a futuros sprints:

- El trabajo realizado en la historia **EP2H3** con la base de datos, supondrá un ahorro

de tiempo en futuros sprints.

7.2.6. Sprint 5 (21/04/22 - 05/05/22)

En la Tabla 7.5, se pueden ver las tareas que se han llevado a cabo en el sprint 5.

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP1H3	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	15 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP1H2	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	8 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP2H4	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	15 horas	12 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
Tiempos Totales		35 horas	35 horas	Todas las tareas finalizadas correctamente

Tabla 7.5: Tareas Sprint 5

En total, el tiempo empleado en el sprint fue de **35 horas**.

En la *Retrospective Meeting*, se ha recopilado la siguiente información de cara a futuros sprints:

- El sprint se ha realizado de forma satisfactoria y lo implementado en el podrá ser utilizado en en próximos sprints.

7.2.7. Sprint 6 (05/04/22 - 19/05/22)

En la Tabla 7.6, se pueden ver las tareas que se han llevado a cabo en el sprint 6.

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP2H6	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	5 horas	5 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP3H3	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	10 horas	12 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
DOC	<ul style="list-style-type: none"> ▪ DOC 	20 horas	18 horas	<ul style="list-style-type: none"> ▪ DOC - Finalizado
Tiempos Totales		35 horas	35 horas	Todas las tareas finalizadas correctamente

Tabla 7.6: Tareas Sprint 6

En total, el tiempo empleado en el sprint fue de **35 horas**.

7.2.8. Sprint 7 (19/05/22 - 02/06/22)

En la Tabla 7.7, se pueden ver las tareas que se han llevado a cabo en el sprint 7.

Historia de usuario	Tareas	Tiempo estimado	Tiempo Empleado	Estado
EP1H4	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	15 horas	25 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
EP1H5	<ul style="list-style-type: none"> ▪ AN ▪ DIS ▪ IMP ▪ TEST 	5 horas	2 horas	<ul style="list-style-type: none"> ▪ AN - Finalizado ▪ DIS - Finalizado ▪ IMP - Finalizado ▪ TEST - Finalizado
DOC	<ul style="list-style-type: none"> ▪ DOC 	15 horas	53 horas	<ul style="list-style-type: none"> ▪ DOC - Finalizado
Tiempos Totales		35 horas	80 horas	Todas las tareas finalizadas correctamente

Tabla 7.7: Tareas Sprint 7

En este sprint, se ha decidido trabajar mucho más en la memoria para adelantar la fecha de entrega del proyecto, por eso las horas de trabajo en este sprint son mayores de lo que deberían.

En total, el tiempo empleado en el sprint fue de **80 horas**.

7.3. Resumen de la ejecución del proyecto

Una vez terminado el proyecto, vamos a observar y comparar distintos aspectos de lo descrito en la planificación con los resultados reales del proyecto.

Por motivos de tiempo, no se ha podido implementar la historia de usuario **EP3H6** - Como usuario quiero poder recibir notificaciones cuando haya algún cambio sobre un proyecto o solicitud, para poder conocer su estado sin necesidad de acceder a la aplicación. Esta historia se añadirá en en la Sección 8.1.

7.3.1. Calendarización

La calendarización inicial, que podemos observar en la Tabla 2.1, ha sido respetada durante casi todo el proyecto, excepto en los sprints finales. Por razones de mejorar la fecha de entrega, solamente se han realizado 7 sprints.

El último sprint del proyecto (sprint 7), a pesar de haber mantenido su duración de 2 semanas, se han trabajado bastante más horas de las planificadas. Esto ha sido para poder terminar el proyecto en una fecha mejor para su entrega.

Tampoco se ha utilizado ninguno de los sprints adicionales ya que no ha sido necesario por el buen desarrollo del proyecto.

En definitiva, el proyecto se ha desarrollado desde el **1 de febrero** hasta el **1 de junio** de 2022, teniendo una duración aproximada de **4 meses**.

7.4. Tiempo empleado

La realización del TFG en el Grado de Ingeniería Informática, según la guía docente (explicado en la Sección 2.3) tiene asignados 12 ECTS. Esto equivale aproximadamente a 300 horas de trabajo del estudiante. Este proyecto se planificó para ser terminado empleando 300 horas, pudiendo aumentarse este tiempo a 370 horas utilizando los dos sprint extra.

Sumando los **Tiempo Totales** de cada Sprint y el tiempo empleado en el sprint 0, el cómputo total del tiempo empleado en la ejecución de este proyecto ha sido de **304 horas**.

7.5. Costes

Teniendo en cuenta la duración aproximada de 4 meses para el proyecto, y 304 horas dedicadas, se recalcula el coste final y se contrasta con los presupuestos reales y simulados planificados en la Sección 2.7.

7.5.1. Coste simulado final

Los elementos que tenderemos son los mismos que en el presupuesto simulado (Sección 2.16, pero variará con la cantidad de tiempo y la duración que se han dedicado realmente al proyecto. El cálculo del coste simulado podemos verlo en la Tabla 7.8.

La dedicación real en horas de cada rol representado por el alumno es la siguiente:

- **Product Owner:** 25 horas

- QA-Tester: 25 horas
- Ingeniero de Software: 50 horas
- Desarrollador Android: 204 horas

Concepto	Precio Unitario	Cantidad	Total
Sueldo Desarrollador Android	13,64€/hora	204 horas	2782,56€
Sueldo Ingeniero de Software	15,55€/hora	50 horas	777,5€
Sueldo QA-Tester	9,03€/hora	25 horas	225,75€
Sueldo Product Owner	15,45€/hora	25 horas	386,25€
Sueldo Scrum Master	15,39€/hora	304 horas	4678,56€
Equipo Empleado	31,23€/mes	4 meses	124,92€
Smartphone de pruebas	17,69/mes	4 meses	70,76€
Espacio Coworking	160/mes	4 meses	640€
Licencia <i>BalsamiQ</i>	7,94/mes	4 meses	31,76€
Licencia <i>Astah Professional</i>	8,82/mes	4 meses	35,28€
Licencia <i>Microsoft 365</i>	10,5/mes	4 meses	42€
TOTAL			9795,34 €
+25 %			12244,18€

Tabla 7.8: Coste Simulado

Como podemos observar, el coste simulado (**12244,18**) es casi idéntico al presupuesto inicial (**12400,65**), aunque está muy ligeramente por debajo. con esto podemos concluir que la estimación inicial fue la correcta y se hubiera podido cubrir el coste simulado al completo.

7.5.2. Coste real final

En este caso, los elementos también son los mismos que los explicados en el presupuesto real en la Sección 2.17, pero al igual que el coste simulado, variará con la cantidad de tiempo y la duración que se han dedicado realmente al proyecto. El cálculo del coste real podemos verlo en la Tabla 7.9.

Como ya comentamos en el presupuesto real, el consumo medio de un portátil es de 0.22 kWh [21], que en 304 horas serían 66.88kWh

El total estimado en el presupuesto real fue de 168,19€, el cual si lo comparamos con el coste total de **138,52€** es mayor. Esto indica que estimación inicial fue correcta y se hubiera podido cubrir el coste real.

Concepto	Precio Unitario	Cantidad	Total
Electricidad	0.45278€/kWh	66.88kWh	30,28€
Equipo Estudiante	20,83€/mes	4 meses	83,32€
Smartphone de pruebas	6,23€/mes	4 meses	24,92€
TOTAL			138,52€

Tabla 7.9: Coste Real

Capítulo 8

Conclusiones

Una vez finalizado este proyecto, me gustaría comentar mis impresiones sobre él y sobre los meses dedicados a él. Han sido unos meses de trabajo muy duro para sacar adelante este proyecto, puesto que tenía que compaginarlo con las practicas de empresa y posteriormente el trabajo. Ha sido muy satisfactorio ir viendo como mi aplicación, **Find Your Project**, iba tomando forma y como una idea que tuve en momentos difíciles, ha llegado a ser una aplicación totalmente funcional y disponible para todos los usuario en Google Play Store. Además, durante la realización de este proyecto, se han intentado aplicar de la manera más satisfactoria posible todos los conocimientos aprendidos durante el estudio de la mención de **Ingeniería del Software** En cuanto a los objetivos establecidos en la Sección 1.4:

- Se ha conseguido desarrollar una app que sirva como nexo de unión para desarrolladores de habla hispana con el inicio de sus proyectos.
- Se ha finalizado de forma exitosa un proyecto de Desarrollo de Software real desde cero.
- La aplicación **Find Your Project** se ha publicado de forma satisfactoria en la tienda virtual de aplicaciones **Google Play Store**.

8.1. Líneas de trabajo futuras

Entre las posibles líneas de trabajo futuras destacan:

- **Uso de notificaciones push.** Como no se ha podido realizar la historia de usuario relacionada con las notificaciones push, se ha decidido añadirla como línea de trabajo futura. Creemos que es una funcionalidad muy interesante para el usuario, puesto que se le notificará cuando suceda algún cambio en alguno de sus proyectos o solicitudes.

- **Apartado Redes sociales en el perfil.** Una de las funcionalidades que me gustaría añadir en un futuro es la posibilidad de añadir al perfil del usuario en la aplicación la información sobre sus redes sociales, para que así los organizadores de proyectos puedan contactar con los solicitantes de una manera más casual.
- **Internacionalización.** Establecer el inglés como idioma principal de la aplicación, siendo posible cambiarlo al español si se desea, permitiendo así que pueda ser utilizada por todo aquel que conozca alguno de estos lenguajes, que cubren una gran parte de la población. Si fuera necesario, también se podría adaptar a otros lenguajes, en función de la demanda de los usuarios.

Bibliografía

- [1] Adobe Flash Player. Búsqueda de subcadenas y patrones en cadenas. https://help.adobe.com/es_ES/as3/dev/WS5b3ccc516d4fbf351e63e3d118a9b90204-7f00.html#:~:text=Las%20subcadenas%20son%20caracteres%20secuenciales,buscar%20subcadenas%20de%20una%20cadena. Accedido el 30/05/2022.
- [2] Android Developers. Android studio. <https://developer.android.com/studio?hl=es>. Accedido el 25/04/2022.
- [3] Android Developers. Fragments. <https://developer.android.com/guide/fragments>. Accedido el 25/04/2022.
- [4] Android Developers. Introduction to activities. <https://developer.android.com/guide/components/activities/intro-activities>. Accedido el 25/04/2022.
- [5] Android Developers. Material design. <https://material.io/develop/android>. Accedido el 25/04/2022.
- [6] Android Developers. Navigation. https://developer.android.com/guide/navigation?gclid=Cj0KCQiA64GRBhCZARIsAHOLriIXx1_eecnAPpIkvL03HU_hJPya63Ykt-Rd77TP8wPqtPsNTjshJskaAsY2EALw_wcB&gclidsrc=aw.ds. Accedido el 02/03/2022.
- [7] Android Developers. Recyclerview. <https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView>. Accedido el 08/06/2022.
- [8] Apple. Comprar un macbook pro de 14 pulgadas. <https://www.apple.com/es/shop/buy-mac/macbook-pro/14-pulgadas>. Accedido el 22/02/2022.
- [9] Astah Professional. Astah professional. <https://astah.net/products/astah-professional/>. Accedido el 22/02/2022.
- [10] BalsamiQ. Balsamiq cloud online store pricing. <https://balsamiq.com>. Accedido el 22/02/2022.
- [11] Ben Kitpitak. Reasons to use android single-activity architecture with navigation component. <https://oozou.com/blog/reasons-to-use-android-single-activity-architecture-with-navigation-component-36>. Accedido el 25/04/2022.
- [12] Bob Hughes Y Mike Cotterell. ¿qué es scrum?

- [13] Claire Drumond. Software project management, 5^a edición, mcgraw-hill higher education.
- [14] Digité. Pruebas de aceptación: el qué y el por qué. <https://www.digite.com/es/agile/pruebas-de-aceptacion/>. Accedido el 30/05/2022.
- [15] Ejemplo de Android. Kotlin android retrofit – introducción y ejemplo. <https://camposha.info/android-examples/es/android-retrofit/>. Accedido el 30/05/2022.
- [16] Figma. Figma. <https://www.figma.com/?fuid=>. Accedido el 25/04/2022.
- [17] Firebase. Firebase. <https://firebase.google.com/>. Accedido el 25/04/2022.
- [18] Firebase. Firebase authentication. <https://firebase.google.com/docs/auth>. Accedido el 25/04/2022.
- [19] Firebase. Firebase realtime database. <https://firebase.google.com/docs/database>. Accedido el 25/04/2022.
- [20] Fiverr. Plataforma de servicios freelance. https://es.fiverr.com/?source=top_nav. Accedido el 19/05/2022.
- [21] GanaEnergia. ¿qué consumo nos supone utilizar el ordenador? <https://ganaenergia.com/blog/que-consumo-nos-supone-utilizar-el-ordenador/>. Accedido el 02/03/2022.
- [22] git. About. <https://git-scm.com/about>. Accedido el 25/04/2022.
- [23] Github. Github. <https://github.com/>. Accedido el 16/05/2022.
- [24] GitLab. Issue boards. https://docs.gitlab.com/ee/user/project/issue_board.html. Accedido el 25/04/2022.
- [25] Glassdoor. Sueldos para el puesto de arquitecto de software en españa. https://www.glassdoor.es/Sueldos/arquitecto-de-software-sueldo-SRCH_K00,22.htm?clickSource=searchBtn. Accedido el 08/06/2022.
- [26] Glassdoor. Sueldos para el puesto de product owner en españa. https://www.glassdoor.es/Sueldos/product-owner-sueldo-SRCH_K00,13.htm?clickSource=searchBtn. Accedido el 08/06/2022.
- [27] Glassdoor. Sueldos para el puesto de qa tester en españa. https://www.glassdoor.es/Sueldos/qa-tester-sueldo-SRCH_K00,9.htm. Accedido el 08/06/2022.
- [28] Glassdoor. Sueldos para el puesto de scrum master en españa. https://www.glassdoor.es/Sueldos/scrum-master-sueldo-SRCH_K00,12.htm?clickSource=searchBtn. Accedido el 08/06/2022.
- [29] GNU. ¿qué es el software libre? <https://www.gnu.org/philosophy/free-sw.es.html>. Accedido el 16/05/2022.

- [30] Hongbeom. Create android app with mvvm pattern simply using android architecture component. <https://medium.com/hongbeomi-dev/create-android-app-with-mvvm-pattern-simply-using-android-architecture-component-529d983eaabe>. Accedido el 30/03/2022.
- [31] Indeed. Salario de un android developer en españa. <https://es.indeed.com/career/android-developer/salaries>. Accedido el 22/02/2022.
- [32] inLab FIB. Descubriendo gitlab. <https://inlab.fib.upc.edu/es/blog/descubriendo-gitlab>. Accedido el 25/04/2022.
- [33] Juan Carlos . ¿cuál es el sueldo de un programador android? <https://www.tokioschool.com/noticias/cual-es-el-sueldo-de-un-programador-android/>. Accedido el 22/02/2022.
- [34] Ken Schwaber Y Jeff Sutherland. Scrum guides.
- [35] Malt. Malt. <https://www.malt.es/>. Accedido el 19/05/2022.
- [36] Max Rehkopf. Epics ágiles: definición, ejemplos y plantillas. <https://www.atlassian.com/es/agile/project-management/epics>. Accedido el 08/06/2022.
- [37] MI. Redmi note 10 pro. <https://www.mi.com/es/product/redmi-note-10-pro/>. Accedido el 02/03/2022.
- [38] Microsoft. Comprar todos los planes de microsoft 365. <https://www.microsoft.com/es-es/microsoft-365/>. Accedido el 22/02/2022.
- [39] Microsoft. Microsoft teams. <https://www.microsoft.com/es-es/microsoft-teams/group-chat-software>. Accedido el 25/04/2022.
- [40] Oscar Blancarte. Observer. <https://reactiveprogramming.io/blog/es/patrones-de-diseno/observer>. Accedido el 30/03/2022.
- [41] Overleaf. Overleaf. <https://es.overleaf.com/>. Accedido el 25/04/2022.
- [42] Q2E Computing. Licencias de software y copyright. <https://www.qe2computing.com/sistemas/gnu-linux/licencias-software-copyright/>. Accedido el 31/05/2022.
- [43] Retrofit. Retrofit. <https://square.github.io/retrofit/>. Accedido el 30/05/2022.
- [44] Samsung. Precio samsung galaxy s22. <https://www.samsung.com/es/smartphones/galaxy-s22/buy/>. Accedido el 22/02/2022.
- [45] Tarifaluzhora. Precio de la tarifa de luz por hora. <https://tarifaluzhora.es/>. Accedido el 02/03/2022.
- [46] Twago. Twago. <https://www.twago.com/>. Accedido el 19/05/2022.
- [47] Universidad de Valladolid. Guia docente del trabajo de fin de grado (mención ingeniera de software).

- [48] VIA LAB Coworking. Tarifas coworking valladolid. <https://www.vialabcoworking.com/precios-alquiler-oficinas-compartidas-en-valladolid/>. Accedido el 22/02/2022.
- [49] Wikipedia. Android. <https://es.wikipedia.org/wiki/Android>. Accedido el 25/04/2022.
- [50] Wikipedia. Furps. <https://en.wikipedia.org/wiki/FURPS>. Accedido el 17/05/2022.
- [51] Wikipedia. Mvvm. <https://en.wikipedia.org/wiki/Model-view-viewmodel>. Accedido el 02/03/2022.
- [52] Yania Crespo González Carvajal. Apuntes del tema 5 de la asignatura diseño de software. Accedido el 30/03/2022.

Apéndice A

Manuales

A.1. Manual de instalación

Para instalar la aplicación, se necesita un **smartphone o tablet Android** cuya versión sea mayor o igual que **Android 9.0 Pie**.

El proceso de instalación es muy simple. Solo debe acceder al enlace del Google Play Store que se encuentra en el Apéndice B e instalar la aplicación o buscar por el nombre “Find your project”.

Será imprescindible disponer de una cuenta de Github para el uso de la aplicación.

A.2. Manual de mantenimiento

A continuación, describiremos todo lo necesario para el mantenimiento del proyecto, tanto para mi como para otro desarrollador que se incorpore.

Lo primero de todo, será descargar el editor de código **Android Studio**. En este proyecto se ha utilizado su versión **2021.1**.

Para acceder al repositorio con el código fuente de la aplicación, será necesario contactar con el creador del proyecto para que este le otorgue permisos de desarrollador, para poder editar y subir código a este.

Una vez hecho esto, será necesario obtener acceso al proyecto de Firebase. Para ello, se requerirá contactar con el creador de este proyecto, puesto que no es seguro dejar aquí las credenciales.

Una vez tenemos acceso al proyecto de Firebase y a la base de datos, será necesario crear un proyecto paralelo con las mismas características y una base de datos de prueba diferente a

la del sistema en producción, ya que el proyecto existente es el utilizado en producción y para probar y desarrollar nuevas funcionalidades, será mejor crear un nuevo proyecto solamente para **desarrollo**.

También serán necesarias las claves y las firmas de la aplicación para poder subir actualizaciones de esta a *Google Play Store*.

Si tiene cualquier duda sobre el proyecto, no dude en contactar con el creador en el correo **gonzocalvosanz@gmail.com**.

A.3. Manual de usuario

Una vez que accedemos a la app por primera vez, nos pide iniciar sesión con nuestra cuenta de Github. Después de esto, accederemos a la vista principal de la aplicación, la vista de **Proyectos** (Figura A.1). A partir de aquí, vamos a distinguir dos tipos de usuarios: **Usuarios que quieran crear un proyecto** (Anexo A.3.1) y **Usuarios que quieran unirse a un proyecto** (Anexo A.3.2).

Todas las figuras que se mencionarán a continuación se encuentran entre las Figuras A.1 y A.16 y estarán al final de este apartado.

A.3.1. Usuarios que quieran crear un proyecto

En este apartado, explicaremos como crear y manejar un proyecto.

Lo primero que tendremos que hacer, será hacer click en el botón para crear un proyecto, el cual se encuentra en la parte inferior derecha de la vista *Proyectos* (Figura A.2). Una vez hecho esto, navegamos a la Vista *Crear Proyecto* (Figura A.3, donde debemos rellenar toda la información relativa a nuestro proyecto. A continuación, accionaremos el botón que dice *Registrar*, creando el proyecto y navegando a la pantalla principal.

Para poder acceder a los proyectos que hemos creado, podemos seleccionarlo en la vista principal, o podemos navegar con la barra inferior a la pantalla de *Perfil* (Figura A.5) y accionar el botón *Mis Proyectos*. Esto nos llevará a la vista *Mis Proyectos* (Figura A.8 pudiendo navegar también a *Mis proyectos iniciados* (Figura A.9. Haciendo click en el botón al lado de cualquier proyecto, navegaremos a la vista *Detalles del proyecto*. Como organizadores del proyecto, podremos editar (click en el botón con el lápiz y navegar a la vista *Editar Proyecto*, Figura A.6), eliminar el proyecto(click en el botón rojo con la papelera) o podremos acceder a las solicitudes de las habilidades. Haciendo click sobre el botón morado a la derecha de cada habilidad, navegaremos a la vista *Solicitudes* (Figura A.10. En esta vista, podremos navegar hacia la vista *Perfil* de los solicitantes pulsando sobre el botón con una *I* de información. También podremos aceptar o rechazar solicitudes, manteniendo pulsado encima de estas y accionando uno de los dos botones.

Una vez que todas las habilidades de tu proyecto tienen al menos un usuario aceptado, te aparecerá el botón de *Crear Repositorio* (Figura A.11). Accionando este botón, se creará el repositorio en tu cuenta de Github y se mandarán las invitaciones a todos los miembros aceptados. Una vez hecho esto, aparecerá el botón *Iniciar* (Figura A.12). Pulsando este botón, se iniciará el proyecto y ningún usuario podrá mandar ninguna solicitud. Además, aparecerá el repositorio de trabajo (Figura A.13).

A.3.2. Usuarios que quieran unirse a un proyecto

En este apartado explicaremos como solicitar unirse a un proyecto y manejar tus solicitudes.

Lo primero que tenemos que hacer es seleccionar el proyecto que queremos y navegar a sus detalles, haciendo click en el botón morado al lado de este (Figura A.2). Una vez dentro, elegiremos la habilidad o habilidades que queramos solicitar, y haremos click en el botón al lado de cada una de ellas. Se nos abrirá una ventana emergente pidiéndonos confirmación (Figura A.14), la cual aceptaremos. Después de realizar esto, veremos que nos aparece un símbolo de interrogación en el botón de la habilidad solicitada (Figura A.15). Eso significa que la solicitud está pendiente de aprobación. Una vez que el creador del proyecto acepte la solicitud, el fondo del icono cambiará a color verde y aparecerá un Tick (Figura A.16).

Para manejar nuestras solicitudes, solo tendremos que navegar a la vista *Perfil* (Figura A.5), y accionar el botón *Mis Solicitudes*. De esta forma, navegaremos a la vista *Mis Solicitudes* (Figura A.7), pudiendo navegar entre las que tenemos pendientes y aceptadas. En caso de querer borrar alguna de ellas, solo será necesario mantener pulsado encima de la solicitud que queramos eliminar y accionar el botón *Eliminar*.



Figura A.1: Vista LogIn



Figura A.2: Vista Proyectos

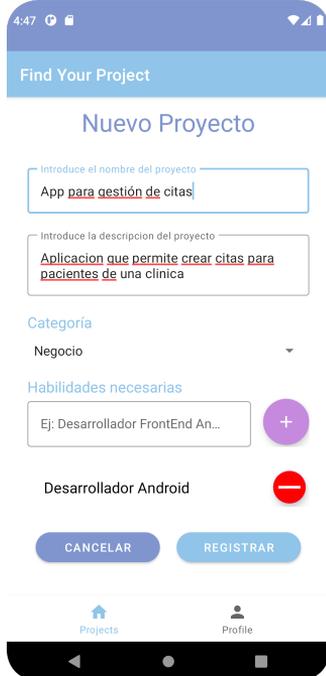


Figura A.3: Vista Nuevo Proyecto



Figura A.4: Vista Detalles del Proyecto

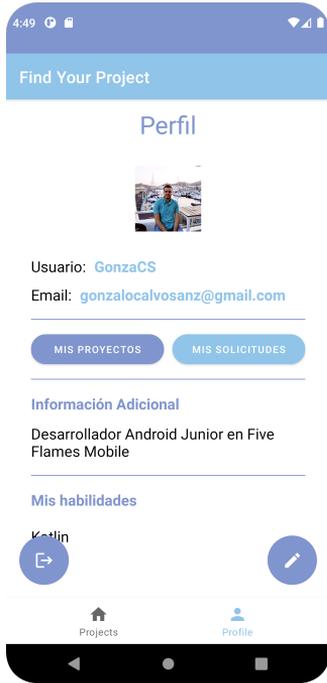


Figura A.5: Vista Información Perfil

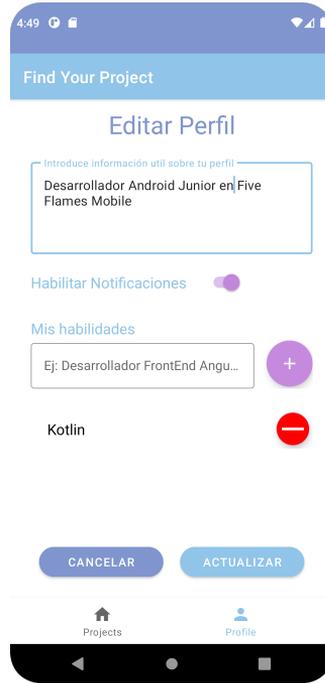


Figura A.6: Vista Editar Perfil



Figura A.7: Vista mis solicitudes



Figura A.8: Vista Mis proyectos



Figura A.9: Vista proyectos iniciados

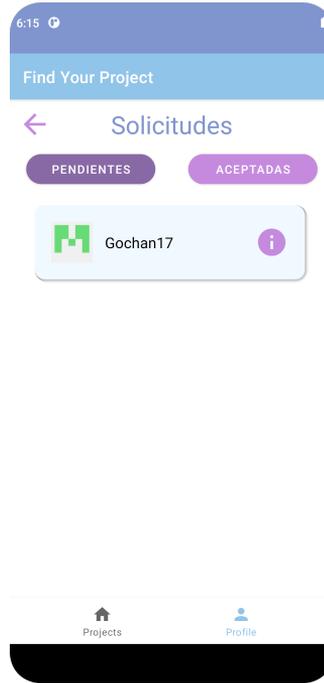


Figura A.10: Vista solicitudes



Figura A.11: Vista Detalles Proyecto completo



Figura A.12: Vista Detalles Repositorio Creado

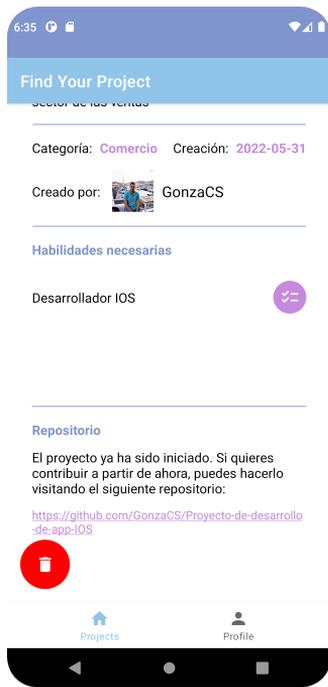


Figura A.13: Vista Detalles Iniciado

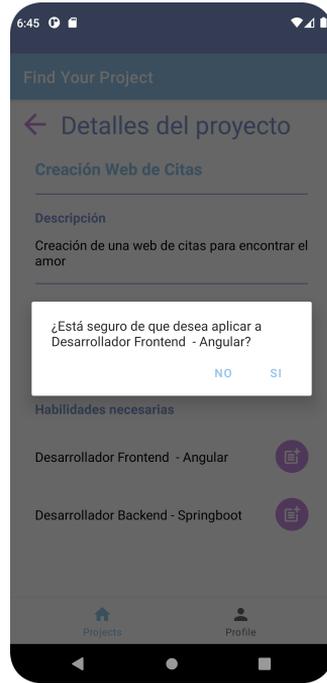


Figura A.14: Vista Enviar Solicitud

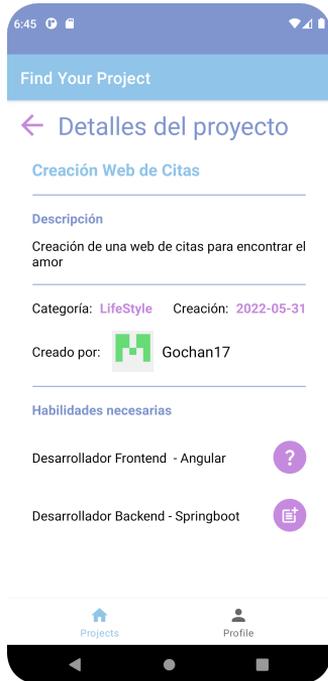


Figura A.15: Vista Detalles Pendiente

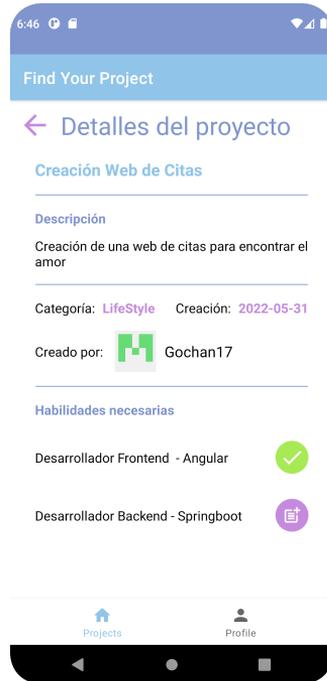


Figura A.16: Vista Detalles Aceptado

Apéndice B

Resumen de enlaces adicionales

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

- Repositorio del código:
<https://gitlab.inf.uva.es/goncalv/find-your-project>.
Solo se proporcionará acceso a los miembros del tribunal durante el periodo de evaluación del trabajo de fin de grado.
- Enlace a Find Your Project en Google Play Store: <https://play.google.com/store/apps/details?id=com.goncalv.findyourproject&gl=ES>