



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

**Grado en Ingeniería Informática
(Mención en Ingeniería del Software)**

**Aplicación multiplataforma para el
Desarrollo Social del Medio Rural con
Flutter y Firebase**

Autor:

D. José María Herrero Calle

Tutor:

D. Joaquín Adiego Rodríguez

Agradecimientos

Este proyecto no se hubiera realizado sin el seguimiento que he tenido durante mi carrera de tantos profesores que me han brindado apoyo a lo largo de los años, y como no dar las gracias, a mi tutor Joaquín, su guía y conocimientos han resultado fundamentales para poder desarrollar el proyecto.

Por último, agradecer también el apoyo y comprensión de mi familia pues han sido muchas horas dedicadas a este trabajo en vez de a ellos, por fin se materializan todos los sacrificios que hemos realizado para poder llegar a la meta.

Mis campeonas y referencia en esta carrera son sin duda mi madre, mi hermana, mi mujer y mi hija, gracias por esperar, animar y aguantar, sin vosotras no hubiera sido posible.

Resumen

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación multiplataforma enfocada al sector del campo, se trata de crear una herramienta para facilitar la interconexión de los agentes que operan en él, gracias al COVID se han puesto de manifiesto las carencias que este sector sufre debido al deterioro de su red social natural a causa de la despoblación.

Así pues, la aplicación consiste en un medio de comunicación que permite al usuario registrarse, compartir contenidos, comentar y sobre todo expandir su red social para con el resto de agentes, pudiendo seguirles, chatear y enviar emails.

El stack tecnológico está alineado con la metodología ágil utilizada, Scrum, y por ello se han utilizado el framework Flutter y la base de datos Firebase.

Abstract

The target of this Final Degree Project is to develop a multi-platform app focused in the contry-side, is about create a tool to made easy the interconnection of the agents that has her activity in it, thanks to covid is be manifest the narrow circumstances that this sector suffers due about the deterioration of her nature social network caused by the depopulation.

So that the application consists in a social media that allow user to register, share contents, comments and above all expand their social network to the rest of agents, being able to follow them, chat and send emails.

The technological stack is in line with the agile methodology used Scrum, therefore the Flutter's framework and Firebase database were used.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Índice general	IX
Índice de figuras	XII
Índice de Tablas	XIII
1. Introducción	1
1.1. Contexto	1
1.2. Motivos	3
1.3. Mercado Internacional	3
1.4. Mercado Nacional	5
1.5. Objetivos dentro del TFG	6
1.6. Objetivos fuera del TFG	6
2. Fases	7
2.1. Fase de definición de producto	7
2.2. Fase de análisis tecnológico	7
2.3. Fase de Formación	7
2.4. Fase de Implementación.....	7
2.5. Fase de testeo y chequeo del producto	8
3. Medios utilizados	9
3.1. Formación	9
3.2. Software	9
3.3. Hardware.....	9
3.4. Equipos de Testeo	10
4. Stack tecnológico	11
4.1. Firebase	11
4.2. Nativas frente a híbridas	12
4.3. Ionic, React y Flutter.....	13
4.3.1. Ionic	13
4.3.2. React.....	13
4.3.3. Flutter	14

4.4.	Dart.....	15
4.5.	Paquetes y librerías utilizados.....	15
5.	Metodología.....	19
5.1.	Scrum.....	19
5.1.1.	Equipo Scrum.....	19
5.1.2.	Artefactos.....	21
5.1.3.	Eventos.....	22
5.1.4.	Estructuración del trabajo.....	23
6.	Planificación.....	25
6.1.	Calendario de Sprints.....	26
6.2.	Planificación con Trello.....	27
6.3.	Sprint 1.....	28
6.4.	Sprint 2.....	29
6.5.	Sprint 3.....	30
6.6.	Sprint 4.....	32
6.7.	Sprint 5.....	34
6.8.	Sprint 6.....	36
6.9.	Sprint 7.....	38
6.10.	Análisis.....	39
6.11.	Presupuesto.....	40
6.11.1.	Costes de personal.....	40
6.11.2.	Costes indirectos.....	40
6.11.3.	Margen Neto / Operativo.....	40
6.11.4.	Presupuesto final.....	41
7.	Documentación Técnica.....	42
7.1.	Requisitos del sistema.....	42
7.1.1	Requisitos funcionales.....	42
7.1.2.	Requisitos de Información.....	44
7.2.	Casos de uso.....	45
7.2.1.	Diagrama de casos de uso.....	45
7.2.2.	Caso de Uso: Login.....	46
7.2.3.	Caso de Uso: Registro.....	47
7.2.4.	Caso de Uso: Crear Publicación.....	49
7.2.6.	Caso de Uso: Ver Perfil.....	50
7.3.	Modelo de dominio.....	51
7.4.	Diagramas de secuencia.....	52
7.4.1.	Diagrama de secuencia del Caso de uso: Login.....	52
7.4.2.	Diagrama de secuencia del Caso de uso: Registro.....	53

7.4.3.	Diagrama de secuencia del Caso de uso: Ver Perfil	54
7.4.4.	Diagrama de secuencia del Caso de uso: Crear Publicación	55
7.5.	Diagrama de paquetes	56
7.6.	Diagrama de despliegue	57
7.7.	Diseño	57
7.7.1.	Arquitectura	57
8.	Pruebas	59
8.1.	Testeo con usuarios	59
8.2.	Resultados	60
8.2.1.	Fallos de comprensión	60
8.1.2.	Fallos a nivel de UI	61
8.1.3.	Fallos de actualización	62
8.2.	Conclusión	62
9.	Conclusiones	63
9.1.	Conclusiones y mejoras futuras	63
9.2.	Líneas para seguir en el futuro	63
A.	Manual de usuario	64
A.1.	Android	64
A.2.	iOS	64
	Bibliografía	66

Índice de figuras

1.1	Densidad de población por provincias	2
1.2	Pirámide poblacional Castilla y León.....	2
1.3	Empresas AgTech similares y financiación recaudada.....	5
4.1	Comparativa renderizado Flutter y React Native.....	14
5.1	Diagrama de estructuración del trabajo en Scrum.....	24
6.1	Estado de tablero, Sprint 1.....	28
6.2	Estado de tablero, Sprint 2.....	29
6.3	Estado de tablero, Sprint 3.....	30
6.4	Estado de tablero, Sprint 4.....	32
6.5	Estado de tablero, Sprint 5.....	34
6.6	Estado de tablero, Sprint 6.....	36
6.7	Estado de tablero, Sprint 7.....	38
7.1	Diagrama de casos de uso.....	45
7.2	Modelo de dominio.....	51
7.3	Diagrama de secuencia del caso de uso Login.....	52
7.4	Diagrama de secuencia del caso de uso Registro.....	53
7.5	Diagrama de secuencia del caso de uso Ver Perfil.....	54
7.6	Diagrama de secuencia del caso de uso Crear Publicación.....	55
7.7	Diagrama de paquetes.....	56
7.8	Diagrama de despliegue.....	57
7.9	Ejemplo de StreamBuilder.....	58
8.1	Issue- Notificación sin imagen.....	61

Índice de Tablas

6.1	Calendario de Sprints.....	26
6.2	Horas estimadas frente a las realizadas, Sprint 1.....	28
6.3	Horas estimadas frente a las realizadas, Sprint 2.....	30
6.4	Horas estimadas frente a las realizadas, Sprint 3.....	31
6.5	Horas estimadas frente a las realizadas, Sprint 4.....	33
6.6	Horas estimadas frente a las realizadas, Sprint 5.....	35
6.7	Horas estimadas frente a las realizadas, Sprint 6.....	37
6.8	Horas estimadas frente a las realizadas, Sprint 7.....	38
6.9	Desglose del presupuesto del proyecto.....	41
6.10	Desglose estimación costes Firebase, Año 1º.....	41

Capítulo 1

1. Introducción

En este proyecto se propone la creación de una Red Social entorno al campo como temática común. Su finalidad es que aquellas personas que viven desarrollando esta actividad encuentren un canal de comunicación adaptado a sus necesidades.

Para comprender mejor el motivo de este TFG, vamos a profundizar en algunos detalles:

- El **contexto** que envuelve actualmente al campo.
- **Motivos** para desarrollar este MVP.
- **Mercado internacional** de este mercado.
- **Mercado nacional**.
- Los **objetivos** dentro del TFG.
- **Objetivos futuros**.

1.1. Contexto

En los últimos años la densidad de población en España ha disminuido de forma alarmante, esto queda reflejado en el informe titulado “La despoblación de la España interior” realizado por Funcas [1] en el que se citan las 23 provincias que tienen territorios susceptibles de ser considerados áreas despobladas, las nueve provincias de Castilla y León se encuentran entre ellas.

(No incluye capitales de provincia ni ciudades de más de 50.000 habitantes)

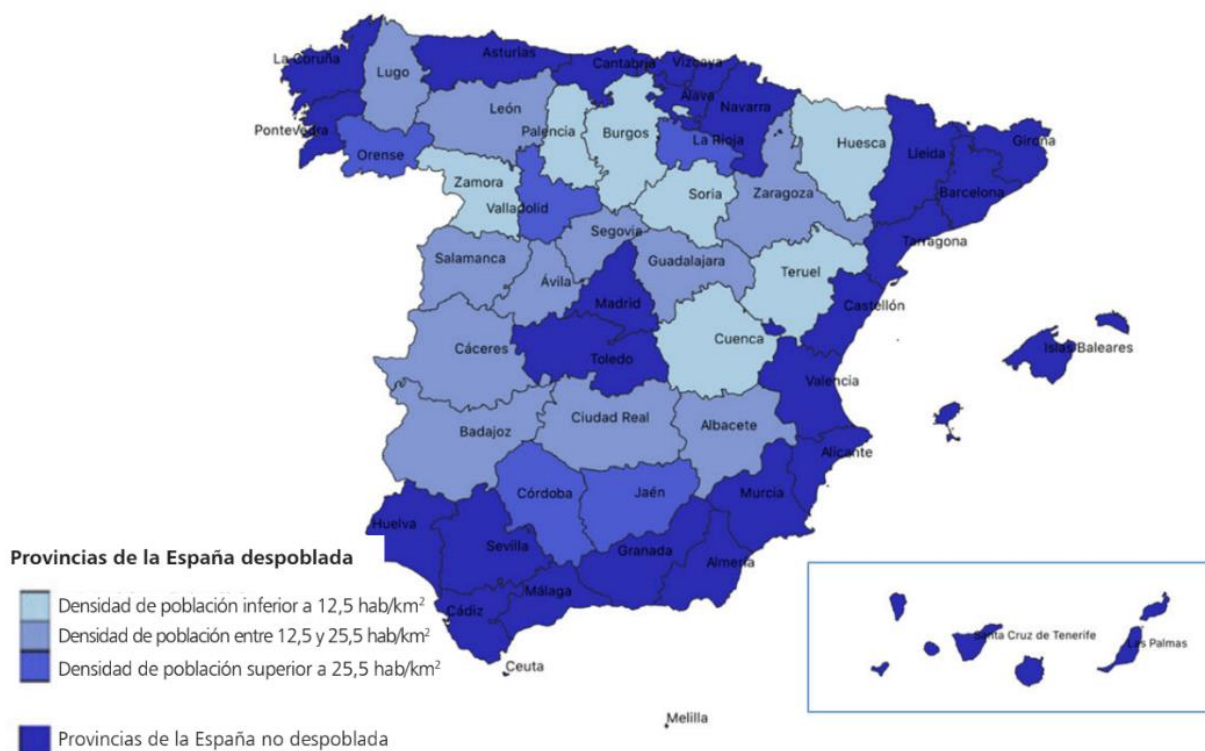


Figura 1.1: Densidad de población por provincias, 2019

Podemos imaginar cómo pueden haberse deteriorado las relaciones por proximidad en este ámbito, y el futuro que nos espera, pues la pirámide poblacional muestra claramente como no hay relevo generacional [2].

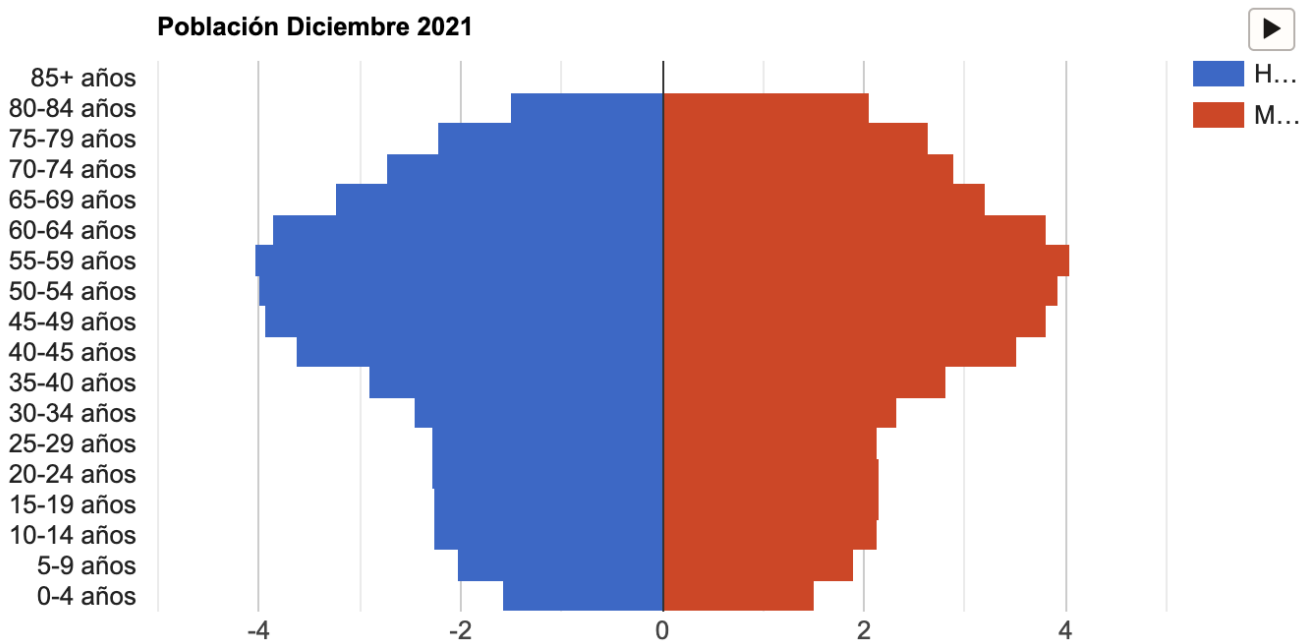


Figura 1.2: Pirámide poblacional Castilla y León

Por lo tanto, el problema se va a ir agravando con los años, este proyecto pretende aliviar las dificultades que mucha gente del campo tiene a consecuencia de este deterioro poblacional para acceder a información sobre su entorno, complicaciones a la hora de conocer los servicios que les rodean o también para consumirlos en proximidad.

Y es que hoy en día la mejor ayuda que se puede dar es a través de la digitalización ya que son zonas de baja densidad poblacional, esto es de sobra conocido por la Unión Europea que no es ajena a la problemática y sitúa como objetivos fundamentales de los fondos FEDER, la innovación y digitalización junto a una Europa Verde, además de otros como la conectividad, temas sociales y acercar Europa a los ciudadanos. [3]

Diversos informes y estudios determinan que los jóvenes y las mujeres son los principales precursores del cambio de esta dinámica negativa y la aplicación nace con la vocación de ser su principal herramienta en el día a día, por ello se tratará de hacer una promoción especial de su actividad y todo aquello que tenga que ver con el eje 5 del Plan de Recuperación del gobierno [4].

1.2. Motivos

La motivación para este TFG viene de una serie de circunstancias que se dieron cuando inicié la carrera, en aquel entonces se produjo la muerte de mi padre, el cual era quien realizaba las labores del campo. En ese momento tengo que compaginar dichas labores con mis estudios en la universidad, de manera que durante muchos años dado que no tenía el mismo conocimiento ni conocía muchos de los servicios y personas que me rodeaban, me veía forzado a tener que elegir entre hacer ciertas tareas del campo o los estudios, lógicamente consiguiendo peores resultados en lo que quedaba para otro momento.

Esta situación fue mejorando, pero esto no significa que esas carencias no sigan existiendo y no se estén agravando en el campo por todo lo que hemos indicado anteriormente.

Para mí este TFG es en realidad un MVP que nace con la idea de ayudar a la comunidad a la que aún pertenezco, la gente que trabaja en campo tiene arraigo por todo lo que le rodea, es este sentimiento de pertenencia el que fomenta que a pesar de que mi situación personal no es la misma que entonces, no pueda mirar para otro lado a la hora de aportar mi granito de arena para mejorar la situación. No será por no intentarlo.

1.3. Mercado Internacional

Ya hay empresas en otros países realizando cosas parecidas a lo que se quiere hacer:

FBN (Estados Unidos, Canada, Australia): es una red de agricultores que opera en varios países, los agricultores disponen de un marketplace para el intercambio de insumos, el principal valor que añaden son el análisis sobre tendencias de los mismos insumos que venden, otro atractivo es un foro en el que los agricultores comparten preguntas y experiencia entre ellos.

AGROFY (Sudamérica, España): marketplace de productos agrarios su valor distintivo es la alta cantidad de suministros que tienen disponible, puedes encontrar de todo y financiarlo a través de múltiples opciones. Sin embargo, no disponen de componente social.

GANAZ (Estados Unidos): es muy similar a lo que se quiere hacer porque es un gestor de mano de obra para la agricultura y empresas agroalimentarias, principalmente opera en web y no tiene ningún componente social.

AGVEND (Estados Unidos): marketplace agrario en la que cada marca dispone de su propia intranet y app de venta, dispone una solución ad hoc para cada una de ellas, pero no tienen componente social.

YAGRO (Inglaterra): marketplace y analítica de datos agro, está enfocado a grandes explotaciones y facilita un mercado de cotizaciones sobre la adquisición de insumos. Desarrolla su actividad ofreciendo tres tipos productos:

- Analíticas: para controlar la productividad de cada explotación.
- Marketplace: En el que comparar precios sobre insumos, maquinaria, etc...
- Plataforma de seguros: Los titulares de explotación reciben una revisión gratuita de su póliza sobre varios proveedores de seguros para conseguir mejores condiciones.

No tiene componente social.

HAYLOAD (Australia) gestor de empleados y maquinaria agrícola, es una especie de Adecco del campo, su punto fuerte es ser especialistas en resolver esa necesidad en el sector, no tiene red social.

A nivel internacional estas empresas están ejerciendo un claro liderato impulsadas por la inversión recibida. Fondos de inversión como Black Rock o grandes empresas con una trayectoria de décadas en el sector como es el caso de Monsanto recientemente adquirida por Bayern, están tomando participación en ellas a sabiendas de que el mercado esta evolucionado y hay que posicionarse en estas nuevas formas de hacer negocio.



Figura 1.3: Empresas AgTech similares y financiación recaudada

Aunque nuestra área geográfica de influencia inicial va a ser España y Portugal, es normal que nuestra red se extienda internacionalmente a otros países.

1.4. Mercado Nacional

En España tenemos a múltiples empresas trabajando dentro de los diferentes segmentos de mercado como pueden ser fitosanitarios, ERP, o AgTech, pero en la actualidad tiene sentido analizar realmente los que por decirlo así, tienen una vocación más digital.

AGROTERRA (España): se inició en 2002 y que en 2018 facturó 155.000 euros dando el salto en 2020 a Estados Unidos, es el marketplace de referencia en el mercado agrario español, trata la mayoría de los segmentos, carece de aplicación y sus planes de expansión están enfocados a Estados Unidos, tiene acuerdos de colaboración con COAG y grandes marcas de suministro, como componente social tiene un foro. La diferencia con la plataforma desarrollada en el TFG sería que no tiene app desarrollada ni es una red social.

AGROANUNCIOS (España): Portal de anuncios dedicado al mundo agropecuario, con un interfaz pobre en diseño se ha convertido en una referencia a la hora de buscar este tipo de productos. No tiene app desarrollada y su modelo de negocio se basa en la publicidad. Que puedan ser referencia, demuestra que en este aspecto hay hueco para nuevos competidores como nosotros.

AGRONOMIE (Francia): Recientemente ha abierto su portal en España es uno de los grandes comercializadores de insumos agrícolas en Francia, destaca por tener opciones orientadas al cliente como negociación de seguros, cliente Premium con ventajas exclusivas, etc... pero no tiene componente social.

VISUAL (España): Es la referencia en cuanto a software de gestión agrícola en España, dispone de app y está contratada por empresas y grandes explotaciones agropecuarias. Es el ERP del campo en España junto a Hispatec y gracias a su ecosistema y capacidad es claramente un posible competidor a largo plazo. No es un canal de comunicación como Cosechalia.

AGRANDO (Alemania): Agrando tiene como objetivo conectar a varias partes interesadas para ello cada empresa puede generar su propia página web de insumos dentro del marketplace y operar a través de su plataforma. No tiene componente social.

SEMBRALIA (España): Web dedicada a la compraventa de suministros muy especialmente en fitosanitarios porque este es el origen de la matriz. Solo aporta eso, no hay app, ni interacción entre usuarios, etc...

AGROTRADER (España): APP que facilita muchas de las cosas que en Cosechalia se quiere introducir, puedes acceder a los anuncios que hay en tu entorno sobre maquinaria, etc... no tiene componente social.

Esta es la más parecida a lo que la plataforma pretende hacer, solo le faltaría la red social.

OSIGRIS (España): APP y web que es un cuaderno de campo, ese es su punto fuerte, facilitan una serie de descuentos por el uso de la app.

Cosechalia se diferencia en que aportamos un componente social.

1.5. Objetivos dentro del TFG

El principal objetivo de este proyecto es desarrollar una aplicación para los entornos Android y iOS aprovechando la tecnología de Flutter, este framework nos permite prácticamente hacer un solo desarrollo para sacar los dos productos, esta aplicación será una red social del campo que tratará de aumentar y mejorar las relaciones entre las personas que viven en él. Esto significa mejorar su acceso a aquello que les rodea y sobre todo relacionarse con el resto de las personas que les son próximas, para ello como objetivos fundamentales se busca facilitar la posibilidad de poder publicar contenidos, así como poder establecer conversaciones a través del servicio de mensajería implementado.

Tras analizar la situación del mercado nacional y exterior, se ve claramente como no hay un agente dominante en el mercado que trabaje las relaciones sociales, por ello y no solo porque él campo lo demanda, es por lo que se va a desarrollar como MVP esta red social.

1.6. Objetivos fuera del TFG

El objetivo final del MVP no es otro que proveer de datos para la toma de decisiones futura, de este modo tras acabar el TFG se introducirán una serie de herramientas como Google Analytics para depurar los flujos de usuario, así como conocer las funcionalidades de mayor interés. Todo esto para definir y probar nuevas funcionalidades relacionadas antes de abrir el producto a nivel nacional.

Capítulo 2

2. Fases

Para conseguir los objetivos y desarrollar el MVP, se define seguir las siguientes fases:

2.1. Fase de definición de producto

- Análisis de soluciones existentes en el mercado, para identificar una propuesta de valor que haga este proyecto la referencia para los agentes del campo.
- Clasificación de los agentes que desarrollan actividad en él y definición de nuestro buyer persona.
- Contraste de la hipótesis de que el producto tiene demanda en el mercado.
- Diseño de producto práctico y funcional para esta etapa del proyecto: MVP.
- Definición de la lógica de negocio y el modelo de almacenamiento (base de datos) para ofrecer las funcionalidades teniendo en cuenta una futura migración.

2.2. Fase de análisis tecnológico

- Definición del stack tecnológico.

2.3. Fase de Formación

- Realización de un curso online de Flutter en que se explican los principales conceptos de Dart y Flutter y a través de ejemplos prácticos para adquirir los conocimientos necesarios en el desarrollo del MVP y posterior publicación en tiendas.

2.4. Fase de Implementación

- Programación en Flutter de la plataforma, así como búsqueda de recursos de imágenes requeridos a nivel de UI. Cuando se ha elegido la metodología de Scrum, uno de los fundamentos para hacerlo es el dinamismo que nos aporta su uso, especialmente en esta fase, ya que es un producto vivo y que esta lógicamente abierto a los cambios.

2.5. Fase de testeo y chequeo del producto

- A través de su instalación en distintos móviles Android comprobar su correcto funcionamiento. A raíz de trabajar durante 3 años como Quality Assurance he podido concluir que puede haber problemas con el layout, plugins, librerías entre los diferentes modelos de móviles. De modo que confío en la utilización de testers para realizar el testeo de la aplicación, así como para recibir feedback al respecto. La mayoría de los fallos se producen por la diferencia de tamaño entre pequeñas y grandes pantallas, así como incompatibilidades con algunas de las versiones de sistema operativo más utilizadas.

- Por ello se ha definido un grupo de testers con perfiles dispares para la evaluación del producto y que disponen de dispositivos de distintas marcas cada uno:

- Ingenieros en Informática.
- Ingenieros en Industrias Agroalimentarias.
- Agricultores.
- Usuarios de redes sociales.

Capítulo 3

3. Medios utilizados

3.1. Formación

Para poder realizar el proyecto, se ha realizado el siguiente curso de Flutter:

- Flutter: Tu guía completa de desarrollo para iOS y Android [5]

Así como consultado en innumerables ocasiones la buena y extensa documentación de Flutter [6].

3.2. Software

- Flutter (Channel stable, 2.10.2)
- Android toolchain - develop for Android devices (Android SDK versión 32.1.0-rc1)
- Xcode - develop for iOS and macOS (Xcode 13.3.1)
- Android Studio (version 2021.1)
- VS Code (version 1.66.2)
- Sistema operativo (macOS 12.3.1 21E258 darwin-x64, locale es-ES)

3.3. Hardware

- MacBook Pro (13-inch, 2017)
 - Procesador: 2,3 Ghz Intel Core i5 de doble núcleo.
 - Memoria RAM: 8GB 2133Mhz LPDDR3
 - Gráficos: Intel Iris Plus Graphics 640 1536MB
 - Disco duro: Sabrent Rocket Nano/ NVMe PCIe M.2 2242 SSD Se actualizan los 128 GB por ser insuficientes para el proyecto.

3.4. Equipos de Testeo

- Lenovo ZUK PRO 2.0 - Android 7.1.1
- Realme GT Master Edition – Android 11 “RealmeUI 2.0”
- Samsung Galaxy A12. – Android 11
- Moto G30 – Android 11
- Vivo Y21S – Android 11 “FunTouch OS 11.1”
- Xiaomi Mi 10T Pro – Android 12 “MIUI V13”

Capítulo 4

4. Stack tecnológico

4.1. Firebase

Para base de datos se ha tenido en cuenta la elección de Firebase [7] por que la idea es crear una red social en la que la interacción entre los agentes y de estos con la plataforma sea recurrente.

El sistema de notificaciones de Firebase no solo es gratuito, sino que es fácil de implementar y nos permite anunciar eventos a los usuarios para que estos sientan desde el primer momento, que están en una plataforma viva.

Al ser una base de datos que sincroniza los datos en tiempo real permite implementar el chat y notificaciones de manera que el usuario perciba una conversación fluida y sin esperas.

La agilidad a la hora de tener el servicio levantado elimina parte del tiempo dedicado a la configuración, contratación e iniciación de un servidor tradicional.

Por otro lado, Firebase tiene algunos pros más y sus contras [8]:

Pros:

- Sistema basado en documentos, que es fácil de diseñar y de entender a nivel de base de datos.
- La integración y mantenimiento con las diversas plataformas es muy sencilla.
- Forma parte del entorno Google de manera que hay un soporte inmejorable sobre actualizaciones, integraciones y mantenimiento de servidores.
- Hoy en día los tiempos de latencia se han reducido considerablemente al ubicar servidores en Europa, de este modo también han eliminado los posibles problemas que pudieran aparecer en cuanto a la protección de datos se refiere al tener que ajustarse a la normativa europea que lo regula.
- Consta de una versión gratuita que resulta ideal para el desarrollo del MVP y que no pone en compromiso el presupuesto inicial, los tiempos de desarrollo, ni el coste de mantenimiento de la infraestructura.
- Resulta clave la facilidad de implementación de un chat con esta tecnología, no solo por el innumerable número de ejemplos que podemos encontrar en internet, sino también porque la configuración del entorno es muy sencilla.

Contras:

- Su escalabilidad es posible, pero proyectando los costes nos indica que hay un momento en el que a no ser que se negocien las condiciones, deberíamos hacer una migración a un sistema propio basado en una tecnología similar, con esta idea, las apuestas a futuro serían Appwrite o Supabase
- Dependemos del proveedor, no tenemos acceso al código fuente para conocer sus entresijos.
- Otro detalle importante es que, aunque las consultas son rápidas, las transacciones de datos pueden ser muy grandes si no hacemos una gestión adecuada. Esto es algo que se tiene en cuenta mejorar en el futuro limitando el número de resultados solicitado y con la introducción de thumbnails que también disminuirán el tamaño de las consultas. Si aun así se mantiene este tipo de problemas, Firebase tiene una herramienta adecuada para el análisis de lo que ocurre: Firebase Performance Monitoring que nos permite obtener métricas sobre ello [9].
- El uso de Firebase no es barato y es complicado hacer una predicción de costes, no obstante, en la sección de presupuesto, se ha hecho el ejercicio con respecto a las estimaciones de uso que se prevén en el primer año de vida del MVP Sacando la conclusión de que es la mejor opción posible en relación costo beneficio.

4.2. Nativas frente a híbridas

Desde agosto de 2018 he desarrollado labores de QA en una empresa de software propietario, testando y analizando las aplicaciones móviles, así he podido valorar las siguientes conclusiones que ayudan a entender por qué utilizar un desarrollo híbrido:

Se puede tener un solo equipo para realizar el desarrollo y mantenimiento de la aplicación, esto redunda en una serie de ventajas:

- Estructura de costes muy inferior a tener desarrolladores de tecnología nativa, la mayor diferencia reside en la aportación de cotizaciones a la seguridad social que disminuye prácticamente a la mitad.
- Desviaciones y posteriores alineaciones entre las plataformas sobre las especificaciones originales del producto, en este caso no cabe lugar a ello, pues normalmente se tiene el mismo código origen, esto podemos tenerlo en cuenta como un sobre coste de tiempo y dinero para el proyecto y en lo que refiere entre plataformas no existirá.
- Encaja como un guante en las metodologías ágiles: Por lo dicho anteriormente disminuye mucho el tiempo dedicado reuniones para realizar cambios, etc...

No todo van a ser ventajas, por contra:

- El salario de los desarrolladores en aplicaciones híbridas complejas es mayor.
- Y en aplicaciones de gran complejidad es recomendable tener un control perfecto del código y subprocesos que pueden darse en ellas, de este modo lo ideal es utilizar desarrollos nativos para tener control completo.
- Lo anterior está alineado con la limitación en cuanto a rendimiento que normalmente presentan las aplicaciones híbridas, estas presentan un mayor consumo de recursos y por lo tanto suelen tener mayor consumo de batería y ser más lentas.

4.3. Ionic, React y Flutter

4.3.1. Ionic

Pros:

- Un solo desarrollo permite sacar el producto en las plataformas más utilizadas WEB, ANDROID, iOS.
- El sistema de componentes acelera el desarrollo.

Contras:

- Comunidad es bastante numerosa pero no tan grande como la de REACT.
- En mi experiencia como desarrollador con esta tecnología, tuve que sufrir un cambio disruptivo entre las versiones IONIC 3 e IONIC 4 en el que a través de un script se debería poder hacer la migración de una versión a la superior. Nunca fue posible.
- Hay un delay importante a la hora de recibir actualizaciones sobre paquetes para la integración de servicios externos, esto deja en clara desventaja a esta plataforma a la hora de tener el producto lo más actualizado posible.

4.3.2. React

React es una tecnología contrastada y sobre todo apoyada por una gran comunidad, por lo que tiene muchísimo código en forma de librerías en JavaScript y que se pueden adjuntar al proyecto disminuyendo mucho el tiempo de desarrollo. Podríamos apostar por ella, porque es una de las tecnologías más demandadas, sin embargo, se espera un cambio en Google que hace que Flutter parezca una opción de mayor futuro y es que Google está trabajando en un sistema operativo del que se espera que Dart sea el lenguaje oficial [10]. Si esto se cumple un desarrollador de Flutter podría trabajar fácilmente sobre el código nativo generado y al revés un desarrollador de Android podría convertirse con facilidad en desarrollador de Flutter.

4.3.3. Flutter

Flutter es el kit de software en código abierto Dart para el desarrollo de aplicaciones híbridas desarrollado por Google. Es uno de los frameworks que consigue generar código nativo, de manera que el rendimiento de este código es muy alto y gracias a ello el UI es muy fluido. El poder acceder a este código nativo, nos permite introducir mejoras a ese nivel. Además de esto, Flutter también copia la idea de React de estructurar el UI en base a componentes, en este caso podemos estructurar el UI con los Widgets que nos proporciona, tenemos dos librerías de Widgets desarrolladas para cada versión de sistema operativo Cupertino para iOS y Material para Android, con lo que podemos reproducir sus estilos de diseño para que la experiencia de usuario sea la mejor posible.

Con las herramientas adecuadas, Flutter puede ser la opción ideal para hacer desarrollos ágiles. Una de estas herramientas, es el plugin que existe para adobe [11] este nos permite migrar [12] directamente a código el UI diseñado, disminuyendo aún más las desviaciones y tiempos de desarrollo, lógicamente no se ha aplicado en el proyecto por falta de tiempo, pero es algo que aporta muchos beneficios a las empresas desarrolladoras de aplicaciones.

Tampoco hay que perder de vista, que cuando desarrollamos en Flutter estamos dentro del entorno Google, de manera que las actualizaciones sobre integraciones e información sobre ciertos bugs producidos en Android deberían llegarnos antes que al resto de soluciones híbridas, por lo tanto Flutter nos permite reducir la vida de este tipo de fallos en el mercado.

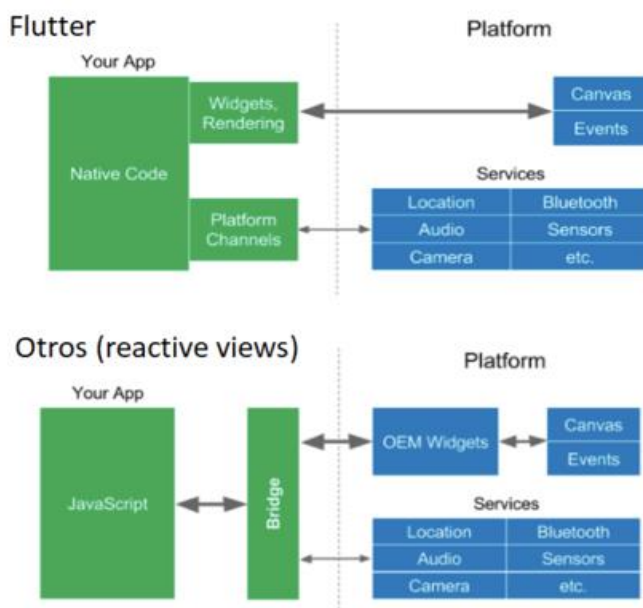


Figura 4.1: Comparativa Renderizado Flutter y React Native

Cuando uno se pone a trabajar con Flutter se da cuenta que, a nivel de UI, nos permite personalizar prácticamente todo, esto es posible por que eleva los widgets y el renderizador desde la plataforma a la aplicación, de manera que, aunque estamos trabajando con componentes, estos se convierten en algo personalizable y extensible. Solo requiere de un canvas en el que renderizar los widgets para que aparezcan en la pantalla del dispositivo, y tener acceso a los eventos y servicios [13].

Por lo que se ha comentado, en términos generales un programador experto puede conseguir con Flutter aplicaciones más agradables, más rápidas, extensibles y personalizables, que con el resto de las soluciones híbridas.

4.4. Dart

El futuro de Dart es prometedor, como se ha comentado anteriormente Google pretende presentar un sistema operativo en este idioma, pero no hay que ir muy lejos para tener un ejemplo o avance de lo que viene. Recientemente, en mayo de 2021 Google presentó Fuchsia OS un sistema operativo para dispositivos conectados que se dice, puede sustituir a Android, sea o no este el relevo, lo que sí es cierto es que su código se encuentra programado entre otros lenguajes en Dart. “C++, Web, Rust, Go, Flutter y Dart.”, dando muestras del peso que tiene en el entorno Google [14].

Y es que Dart ofrece ventajas importantes:

- Animaciones y transiciones que se ejecutan en 60 fps mucho más suaves.
- Realizar object allocation y garbage sin bloqueos, de modo que a nivel visual disminuye la aparición de el jank y el sutter.
- Como se puede ver en el esquema anterior, al generar código nativo, no tiene que crear un puente entre dominios, eso hace que no haya cuellos de botella en el rendimiento.
- Dart no necesita un lenguaje de diseño declarativo separado como JSX o XML, así tiene un interfaz visual integrado para Flutter además la característica de Hot Reload da la opción al compilador de recargar y continuar ejecutando código sin perder el estado de la aplicación, de manera que hace que el diseño sea algo fácil y rápido de implementar.

4.5. Paquetes y librerías utilizados

Existe un espacio web oficial que actúa como repositorio para los paquetes que podemos incluir en los proyectos realizados en Dart o Flutter. En PUB.DEV [15] encontraremos una descripción de la funcionalidad, así como ejemplos de uso y la valoración que tiene por parte de la comunidad.

Al igual que sucede en javascript con npm, la comunidad de flutter no hace más que aumentar el número de paquetes y librerías disponibles facilitando así el desarrollo agregando funcionalidades modulares.

En este TFG se ha agregado un gran número de librerías, cada una con un propósito diferente:

- **flutter_lints:** contiene un conjunto recomendado de lints para aplicaciones, paquetes y complementos de Flutter para fomentar las buenas prácticas de codificación. Verifica estáticamente el código.
- **cloud_firestore:** permite la comunicación con Firebase y establecer el modelo de datos utilizado.

- **firebase_auth:** nos da acceso por autenticación a Firebase y todas las funcionalidades derivadas como recuperar contraseña y gestionar las sesiones.
- **google_sign_in:** en este caso nos permite la autenticación directa con el sistema de Google.
- **uuid:** nos facilita la creación de un identificador inequívoco por usuario dentro de nuestro sistema, entre las versiones que podemos aplicar, elegimos V4 por que genera un uuid aleatorio que no permite relacionar directamente la información con la fuente que la ha generado.
- **firebase_analytics:** en esta etapa del MVP no está implementado pero en el futuro permite hacer un seguimiento adecuado de los flujos de usuario, definir embudos de conversión y así dar prioridad a ciertas funcionalidades en detrimento de otras. Fundamental para la toma de decisiones futuras.
- **firebase_storage:** nos permite comunicarnos con Firebase y de esta manera poder acceder al almacenamiento, el cual, es fácilmente escalable.
- **image_picker:** nos permite acceder a las imágenes del móvil para subirlas a la red social.
- **image:** nos permite manipular y cargar imágenes de tipo: PNG, JPEG, GIF, BMP, WebP, TIFF, TGA, PSD, PVR, y OpenEXR
- **path_provider:** para encontrar las ubicaciones más comunes del sistema de archivos el directorio temporal y el de documentos.
- **firebase_messaging:** es clave para el desarrollo de la aplicación porque nos permite realizar la organización de mensajes y notificaciones push en Android y iOS sin necesidad de gastar dinero por este servicio.
- **flutter_svg:** permite trabajar con SVG para integrarlos en el proyecto cambiando características como el color.
- **flutter_svg_provider:** podemos introducir un svg como imagen de un widget de tipo imagen, esto no sería posible sin esta librería.
- **cached_network_image:** muestra imágenes de internet y las mantiene en cache.
- **timeago:** nos permite convertir timestamps pasados para mostrar el tiempo que ha transcurrido desde entonces, solo en inglés y español.
- **material_design_icons_flutter:** librería que contiene todos los iconos de diseño disponibles.
- **firebase_core:** es fundamental para poder utilizar los servicios que ofrece Firebase en la app desarrollada, pero, además, si quisiéramos utilizar algunos servicios del proyecto en otras app secundarias nos lo permitiría hacer.
- **fluttertoast:** para la creación de mensajes emergentes de tipo toast en Android y iOS.

- **photo_view:** permite acceder a las fotos compartidas en el chat y así hacer zoom en ellas, etc.
- **google_fonts:** paquete para utilizar las fuentes de fonts.google.com.
- **provider:** es un wrapper que se coloca envolviendo InheritedWidgets para poder compartir propiedades en toda la aplicación.
- **story_view:** directamente nos da la posibilidad de visualización de Stories.
- **flutter_staggered_grid_view:** genera la posibilidad de mostrar contenidos en cuadrículas de distintos diseños.
- **share:** con esta funcionalidad comparte contenidos fuera de la aplicación.
- **url_launcher:** permite abrir URL de tipo web, teléfono, SMS, e email.
- **flutter_linkify:** convierte las URL e emails en elementos clickables.
- **ionicons:** galería de iconos de Ionic.
- **link_preview_generator:** para previsualizar el contenido de los links.
- **another_flushboard:** customiza los mensajes de notificación al usuario.
- **pagination_firestore:** permite simplificar la paginación de con datos obtenidos del Firestore.
- **flutter_reaction_button:** su integración nos permite añadir el botón para agregar reacciones en las publicaciones.
- **adaptive_theme:** se van a poder adaptar los colores de la aplicación para cambiar dinámicamente entre un tema light o dark.
- **string_validator:** facilita la validación de strings así como la conversión a distintos formatos.
- **json_annotation:** proporciona funciones para poder trabajar fácilmente con el formato de datos Json.
- **badges:** así obtenemos la funcionalidad que permite generar avisos en los iconos correspondientes principalmente indicando que el usuario tiene notificaciones pendientes de consultar.
- **iconly:** permite utilizar esta librería de iconos.
- **webview_flutter:** da la opción de utilizar widgets de tipo webview.
- **flutter_mobx:** facilita la introducción de observer widgets que se actualicen cuando se produzcan cambios.
- **flutter_localizations:** da la opción de la internacionalización de los strings de la app.
- **background_app_bar:** con esta librería podemos introducir una imagen en el fondo de la appBar.

- **country_code_picker:** añade la funcionalidad de poder seleccionar el país desde un desplegable dando como opciones el prefijo, bandera y/o nombre del país.
- **geocator:** proporciona la localización en Android o iOS. “En desarrollo de funcionalidades”
- **geocoding:** permite realizar el reverse geocoding en relación con la localización o la localización en función a la dirección. En el futuro este será un servicio realmente recurrente en la app, con lo que no se ha implementado por querer utilizar un proveedor gratuito para estos servicios como puede ser [HERE](#).

Capítulo 5

5. Metodología

Para realizar la planificación se han descartado los modelos estructurados debido a que el software está en continua evolución, la tecnología o requisito que aplicabas ayer mañana puede quedarse obsoleto. Estos últimos años he estado aplicando metodologías ágiles para el desarrollo del software, y he podido comprobar como los requisitos van cambiando incluso cuando se desarrolla software propietario por eso la confianza en elegir una metodología ágil en concreto Scrum.

5.1. Scrum

Entre los principios de Scrum se encuentra el trabajo colaborativo, iterativo e incremental para poder adaptarnos a los cambios que pueden aparecer a lo largo del desarrollo del proyecto. En este caso la elección era clara debido a la falta de experiencia sobre Flutter y las posibilidades de cambio sobre los requisitos del proyecto.

Para la planificación también tenemos en cuenta que a diferencia de los modelos estructurados en scrum podemos hacer coincidir fases del proyecto o cambiar su orden lógico.

Al aplicar Scrum, utilizamos en una serie de elementos base [16]:

5.1.1. Equipo Scrum

El Scrum Team debe componerse por perfiles multifuncionales, cada uno siendo responsable de sus tareas. Todos los componentes deben ser capaces de trabajar por objetivos, pues siempre deben controlar los plazos de tiempo y la meta a la que se debe llegar. Todos estos puntos son clave para el éxito y otorga independencia al equipo con respecto a agentes externos ya sean de la misma empresa o no.

Dentro de un Scrum Team, encontramos los siguientes roles principales:

5.1.1.1.Product Owner

Es el propietario del producto y su objetivo es maximizar la propuesta de valor. Para eso debe ser capaz de colocarse del lado del cliente.

En sus manos está la gestión del Product Backlog que no es otra cosa que el conjunto de tareas que consiguen dar valor al resultado.

5.1.1.2.Scrum Master

Es un coordinador, por eso es el que tiene la misión de aplicar Scrum y trasladar al equipo la metodología para que el desarrollo del producto se concrete en tiempo y forma.

De esta manera su cometido principal es el de facilitador que elimina los impedimentos que le van surgiendo al Scrum Team a medida que avanza el proyecto. Otra parte relevante de su trabajo es hacer que todo el equipo entienda lo que es el estado de Done y sean capaces de aplicar Scrum para llegar hasta él.

También es responsabilidad del Scrum Master mantener la motivación y el buen entendimiento entre los componentes del equipo.

El Scrum Master sería como la mano derecha del Product Owner a la hora de hacer entender al equipo el objetivo, el alcance y el dominio del producto, así como enseñar al propio Product Owner como trabajar con el Product Backlog para maximizar el valor del producto.

- Es muy importante tener claro que el Product Owner no es el líder del proyecto:
- Las decisiones las toma el equipo.
- No define los tiempos de ejecución, solo hace el seguimiento.
- No organiza ni indica qué tareas realizar ni en qué orden, eso lo hace el Product Owner.

Lo que debe ser es una referencia que en todo momento facilite la aplicación de Scrum para llevar a cabo el proyecto que se tiene entre manos.

5.1.1.3 Scrum Development Team

El equipo de desarrollo se compone de profesionales de manera que forman un equipo auto-organizado y multifuncional. Esto significa que toman la decisión de definir las tareas y quien las realiza, su cometido es el desarrollo del producto basándose en la metodología Scrum.

5.1.1.4 Skateholders

No podemos considerarlos como un rol propio en la metodología Scrum pero cumplen una tarea muy importante ser un agente independiente al trabajo del Scrum Team que enriquece con feedback el producto final. Por eso se suelen agregar al Sprint Review, canalizar sus deseos y necesidades sobre el producto al Product Backlog dan la posibilidad de aumentar la propuesta de valor.

5.1.2. Artefactos

El propósito de los artefactos es canalizar la información de manera que todos los miembros del equipo scrum entiendan de la misma manera el artefacto, como artefactos tenemos:

5.1.2.1.Product Backlog

Mantiene una pila ordenada con lo que debe componer el producto, de esta manera es donde residen los requisitos del proyecto.

Se puede decir que producto y pila son dependientes el uno del otro, mientras el producto se desarrolle la pila existirá y mientras la pila exista el producto tendrá mejoras pendientes.

Los atributos que definen el Product Backlog son:

- Descripción
- Orden
- Estimación
- Valor

5.1.2.2.Spring Backlog

Es el conjunto de elementos del Product Backlog que se toman para formar el Sprint.

5.1.2.3.Increment

El Sprint Goal puede aumentar o disminuir a partir del Sprint Backlog a medida que el Sprint avanza. Cuando el Sprint concluye, todos esos elementos seleccionados del Sprint Backlog concurren en un Incremento de producto terminado, este es el Incremento del que hablamos.

5.1.3. Eventos

Su finalidad es crear regularidad y disminuir el número de reuniones improvisadas.

5.1.3.1.Sprint

Su cometido es el de contener al resto de eventos, se establece una duración máxima de 4 semanas. A diferencia del resto de eventos la duración del Sprint no se modifica a lo largo del proyecto.

Su objetivo es un incremento de producto terminado, utilizable y desplegable. El alcance es modificable siempre que el Product Owner y el equipo de desarrollo lo vean necesario.

5.1.3.2. Sprint Planning

La reunión de planificación define el alcance del sprint y viene a durar unas 6 - 8 horas, todo el equipo participa para definir el siguiente Incremento de valor.

Durante el Sprint Planning, el Product Owner define el objetivo y elementos que compondrán el Product Backlog.

El Scrum Master otorgará las tareas a cada equipo y guiará a cada uno con el fin de que cada tarea tenga un tiempo definido y que vaya en línea con la estimación temporal final que tiene el Sprint.

En este evento el Scrum Team, selecciona unas tareas del Product Backlog para componer el Sprint Backlog, conseguir desarrollar estas tareas es lo que se conoce como Sprint Goal.

5.1.3.3.Daily Scrum Meeting

Estas reuniones diarias son uno de los elementos que hacen que Scrum sea tan dinámico y adaptativo. Su propósito es que el equipo de desarrollo comparta sus avances y en relación con cómo se está ejecutando el proyecto se planté la siguiente jornada de trabajo.

El resultado es la cohesión y coherencia de lo que se hace, el equipo explica al Scrum Master y Product Owner cómo van a completar el Sprint Object.

Estas reuniones se realizan siempre a la misma hora y tienen una duración de 15 minutos.

5.1.3.4.Sprint Review

En esta reunión el equipo de desarrollo presenta al resto el resultado del Sprint su duración está relacionada con la duración del Sprint y viene a ser una hora por semana dedicada.

La reunión discurre con la participación del Scrum Team y a veces son invitados los Stakeholders, si así se lo propone el Product Owner.

El Product Owner indica que objetivos se consiguieron y cuales no, analiza el Product Backlog y propone si es así una nueva fecha de fin del Proyecto.

El equipo indica que se ha hecho bien, que no, y los problemas y carencias se han tenido en el Sprint.

Finalmente se suele modificar el Product Backlog con nuevas tareas por la aparición o modificación de requisitos.

5.1.3.5.Sprint Retrospective

Trata en detalle el trabajo realizado en el Sprint, de manera que el Scrum Team puede proponer mejoras. El Scrum Master tiene el objetivo de fomentar este proceso canalizando al equipo para encontrar soluciones y gestionar las relaciones interpersonales, con ello se pretende lograr trabajar de manera más eficaz el próximo Sprint.

Debería durar 3 horas para un Sprint de 4 semanas y disminuir 45 minutos por cada semana que no dure el sprint.

5.1.4. Estructuración del trabajo

Para estructurar el trabajo tenemos una serie de conceptos que nos ayudan a la hora de conceptualizarlo y dimensionarlo [17]:

5.1.4.1.Stories

Historias de usuario, de antemano hay que aclarar que no son requisitos del sistema, son la explicación desde el punto de vista del usuario de una funcionalidad que le aporta valor, lógicamente debe concluir con una implementación software desarrollada por el equipo de desarrollo. Estas se pueden descomponer en tareas y subtareas.

5.1.4.2.Epics

Las Épicas son una composición de historias, estas historias, aunque realizadas de manera independiente, cuando se completan, conducen a formar parte de un trabajo de mayor que llamaremos Épica.

5.1.4.3.Initiatives

A su vez las Iniciativas están compuestas por Épicas. La Iniciativa se sitúa a nivel estructural por encima de ellas ya que representan una complejidad mayor que no se puede plasmar de otro modo, estamos hablando de objetivos de mayores dimensiones y extensos.

A medida que aumenta la complejidad, el tiempo necesario es mayor, cuando hablamos de historias, nos referimos a semanas, para Épicas meses y cuando nos referimos a Iniciativas consideraremos varios trimestres o incluso el año.

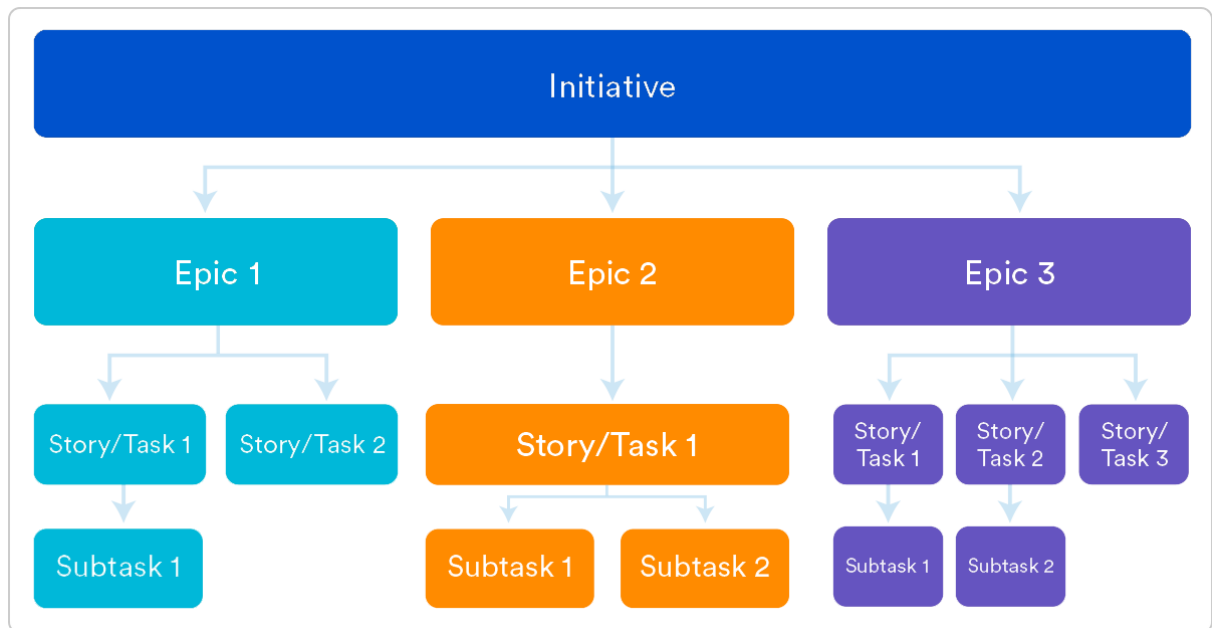


Figura 5.1: Diagrama de estructuración del trabajo en Scrum

Capítulo 6

6. Planificación

A la hora de aplicar Scrum tenemos una serie de limitaciones, el equipo y recursos son los justos de manera que se han aplicado una serie de cambios por necesidad.

Para el desarrollo mínimamente correcto del proyecto se ha tenido en cuenta que el Product Owner será el tutor del proyecto, así como que las figuras de Scrum Master y el Scrum Development Team recaerán en mí.

El tiempo dedicado al producto que se pretende presentar al terminar el proyecto, debe tener una estimación inicial de 300 horas, que coincide con la carga en horas que debe cumplir un TFG. Con independencia de las desviaciones que se puedan producir a lo largo del proyecto la duración mínima debe ser esta.

Como primera toma de contacto, se ha tenido una reunión para dar a conocer la idea que se tiene en mente y realizar un análisis de necesidades para poder completar una versión entregable del mismo, de esta manera se concretaron las fases que hemos visto en el **Capítulo 2**.

- Fase de definición de producto
- Fase de análisis tecnológico
- Fase de Formación
- Fase de Implementación
- Fase de testeo y chequeo del producto

Inicialmente se considera esta distribución de Sprints, en los que se tienen en cuenta Fases que no son estrictamente implementación, en el trabajo he podido ver como Scrum es capaz de introducir en la planificación pruebas de concepto, documentación, búsqueda de alternativas, elaboración de presupuestos, formaciones, etc, todo para llevar a buen puerto el producto:

- Sprint 1: Fases de análisis y definición
- Sprint 2: Fase de Formación
- Sprints 3 – 6: Fase de Implementación
- Sprint 7: Fase de Testeo y chequeo del producto a la que se añade el cierre y presentación documental del TFG.

6.1. Calendario de Sprints



Tabla 6.1: Calendario de Sprints

De manera que tenemos 3 semanas para el desarrollo de cada Sprint, otorgando a cada uno 43 horas.

Otra de las adaptaciones que se han tenido que realizar es substituir las Daily y trasladar el objetivo de estas reuniones diarias a una sola reunión que se unifica con el Sprint Review y el Sprint Retrospective, en esta reunión se hace una demo del Incremento en el Sprint, se valora el resultado de este, se analiza la visión del producto por eso se formulan estas preguntas:

- ¿Qué ha ido bien?
- ¿Qué ha ido mal?
- ¿Qué se puede mejorar?
- ¿Qué se puede agregar?
- ¿Qué cosas pueden impedir un progreso adecuado en él futuro?
- ¿Qué aporta más valor al cliente?
- ¿Falta algo en la planificación?
- ¿Falta algo en él producto?

Tras esta reunión también nos hemos visto obligados a realizar el Sprint Planning, para concretar el Sprint Backlog y el Sprint Goal siguientes.

6.2. Planificación con Trello

A la hora de elegir un software para hacer el seguimiento de Scrum me hubiera gustado utilizar lo que utilizaba en el trabajo: Confluence a nivel documental y Jira a la hora de realizar los tableros y crear las historias, pero dado que estas son de pago, se ha decidido utilizar Trello que es una herramienta gratuita del mismo Grupo de Software Atlassian.

Trello nos permite combinar una serie de herramientas [18] que nos para realizar la planificación de modo gratuito:

- **Trello Boosted Boards:** Super potente para el trabajo que tenemos entre manos pues nos da un Contador de tarjetas, Clasificación de tarjetas por labels, títulos a las tarjetas, establecer niveles de prioridad...
- **Trello Cards Links:** Permite agregar links a las tarjetas.
- **Trello Pegatinas:** Permite personalizar con imágenes las tarjetas.
- **Trello Butler:** Permite meter reglas automáticas por ejemplo que al cerrar una tarea, su fecha de vencimiento se maque como terminada.
- **Elegantt App:** Nos permite introducir un diagrama de Gantt para visualizar rápidamente la línea temporal de tareas que tenemos planificada.
- **Deadlines:** crea una vista en la que a la izquierda aparecen las tareas asignadas en orden de mayor a menor cercanía a la fecha de vencimiento y a la derecha las que no están asignadas ordenándolas también de mayor a menor fecha de vencimiento.

6.3. Sprint 1

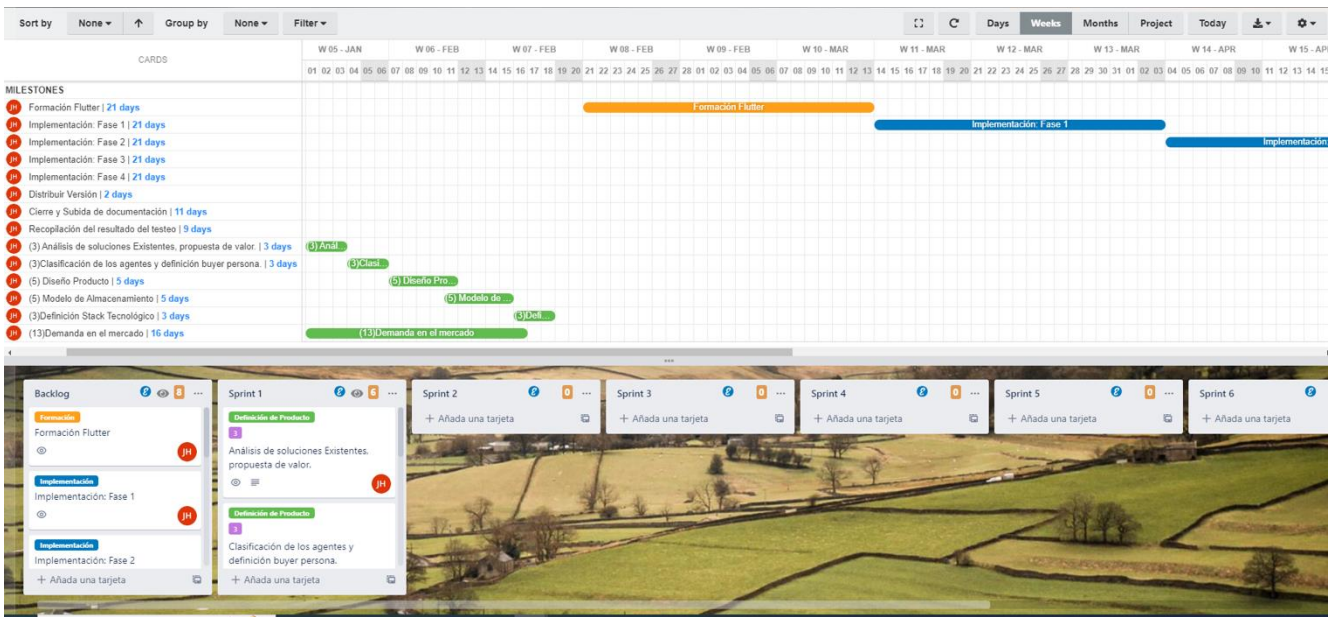


Figura 6.1: Estado de tablero, Sprint 1

El Sprint 1 pretende generar una idea global del producto, tener la visión adecuada acorde al mercado, por ello se realizan una serie de encuestas y se analiza el mercado para poder marcar las líneas de trabajo sobre el producto, en este Sprint también se define el conjunto de herramientas software con las que vamos a trabajar.

En la figura anterior se puede ver el resultado del primer Sprint Planning, no solo se hace una estimación de las días-horas que puede llevarnos, sino también una estimación de la dificultad que entraña cada tarea, en este caso la más compleja es extraer la información de lo que demanda el mercado, por eso al valorarlo en relación con la escala de Fibonacci (1, 3, 5, 8, 13) la otorgamos la máxima puntuación, trece puntos de historia.

Horas estimadas VS definitivas:

Sprint 1			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Analisis de Propuestas Existentes	6	10	
Clasificación de Agentes y definición del Buyer Persona	5	10	
Diseño de Producto	7	9	
Modelo de Almacenamiento	8	8	
Definición de Stack Tecnológico	5	5	
Demanda en el Mercado	12	19	
	43	61	18

Tabla 6.2: Horas estimadas frente a las realizadas, Sprint 1

6.4. Sprint 2

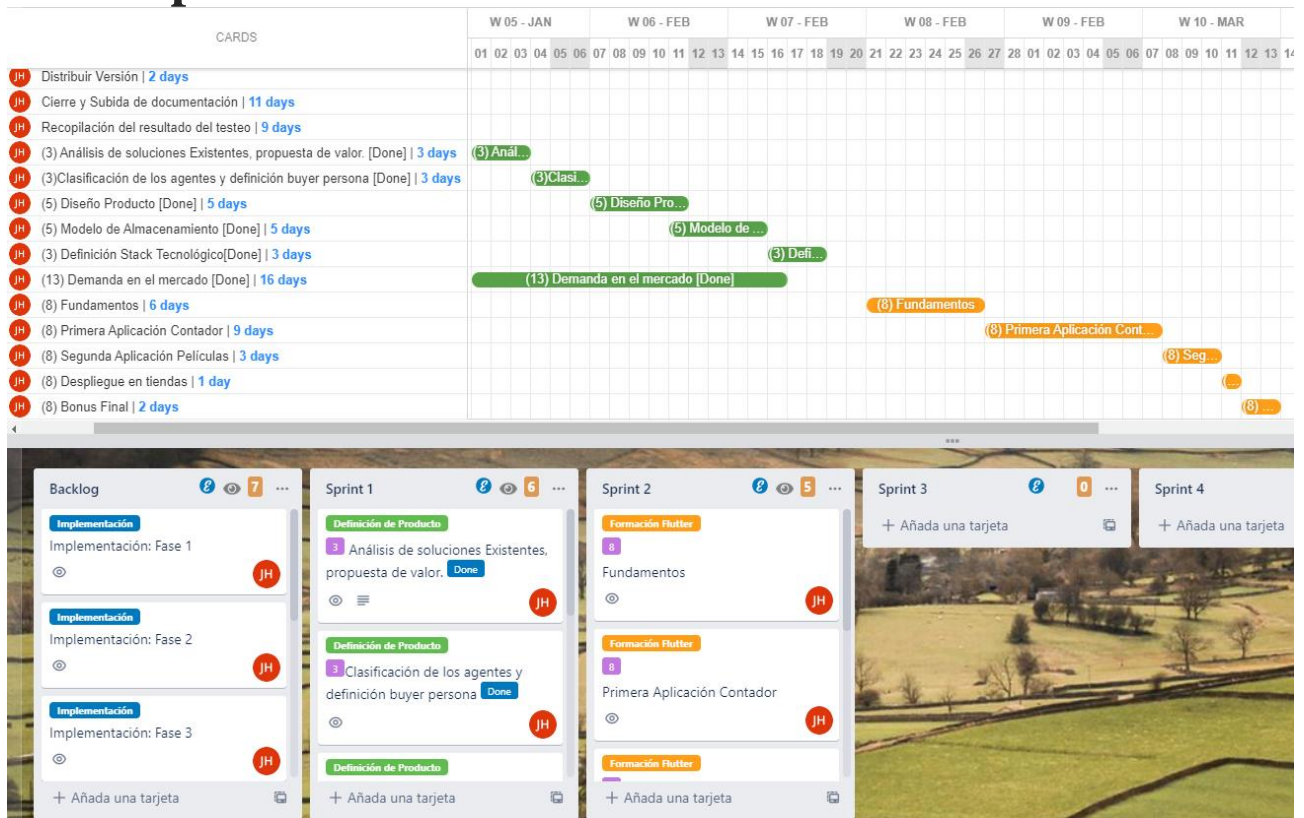


Figura 6.2: Estado de tablero, Sprint 2

El objetivo del Sprint 2 es completar un curso de Flutter, este se ha seleccionado no solo por la reputación de su realizador Fernando Herrera [5], también por que se aplica la metodología de learning by doing. Otro detalle importante es que debía acercarse al tiempo que disponemos en el sprint para cumplir realizarlo, en este caso el curso se compone de 37,5 horas de video, y se estimó que podría llevar 43 horas su realización.

A raíz del Sprint Retrospective, teniendo una idea más clara de lo que se va a hacer y conociendo el curso, esta tarea se ha fraccionado en historias, de esta manera se han definido 5:

- **Fundamentos (8) Puntos de historia**
- **Primera Aplicación Contador (8) Puntos de historia**
- **Segunda Aplicación Películas (8) Puntos de historia**
- **Despliegue en tiendas (8) Puntos de historia**
- **Bonus final (8) Puntos de historia**

Como no se ha programado anteriormente en Flutter se ha valorado todo con el mismo nivel de complejidad. La estimación de horas se ha realizado dividiendo las horas del curso entre los 21 días del Sprint de esta manera las horas que ocupa cada parte equivalen a los días de Sprint estimados.

Horas estimadas VS definitivas:

Sprint 2			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Fundamentos	12	16	
Primera Aplicación Contador	16	28	
Segunda Aplicación Películas	7	16	
Despliegue en tiendas	3	0	
Bonus Final	5	0	
	43	60	17

Tabla 6.3: Horas estimadas frente a las realizadas, Sprint 2

6.5. Sprint 3

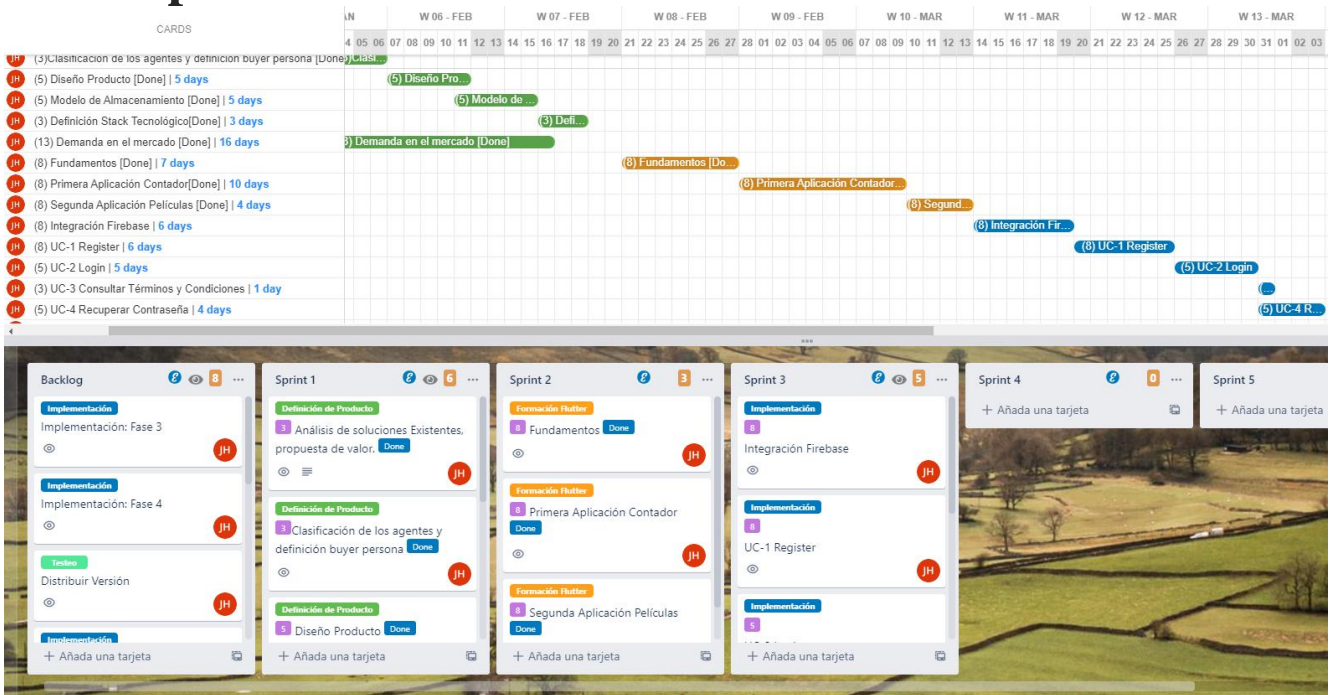


Figura 6.3: Estado de tablero, Sprint 3

En el Sprint Planning, se tienen en cuenta las desviaciones que se han dado en el Sprint anterior, por ello, sobre los casos de uso a implementar como historias, incluiremos en el Sprint backlog todos los relacionados con la misma temática y plantearemos el Sprint con los que se estima podrán realizarse. Si da tiempo se irán incluyendo más historias al Sprint Goal y si no se sacarán de la planificación del Sprint Goal, esto es lo que aplicaremos en los siguientes cuatro Sprints de Implementación.

Sprint Backlog:

- **Integración de Firebase. (8) Puntos de historia**
- **UC-1 Register (8) Puntos de historia**
- **UC-2 Login (5) Puntos de historia**
- **UC-3 Consultar Términos y Condiciones (3) Puntos de historia**
- **UC-4 Recuperar Contraseña (5) Puntos de historia**
- **Diseñar Home Page (13) Puntos de historia**

Sprint Goal:

- **Integración de Firebase. (8) Puntos de historia**
- **UC-1 Register (8) Puntos de historia**
- **UC-2 Login (5) Puntos de historia**
- **UC-3 Consultar Términos y Condiciones (3) Puntos de historia**
- **UC-4 Recuperar Contraseña (5) Puntos de historia**

Horas estimadas VS definitivas:

Sprint 3			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Integración de Firebase.	12	15	
UC- Register	11	18	
UC- Login	9	15	
UC- Consultar Términos y Condiciones	2	3	
UC- Recuperar Contraseña	9	10	
	43	61	18

Tabla 6.4: Horas estimadas frente a las realizadas, Sprint 3

6.6. Sprint 4

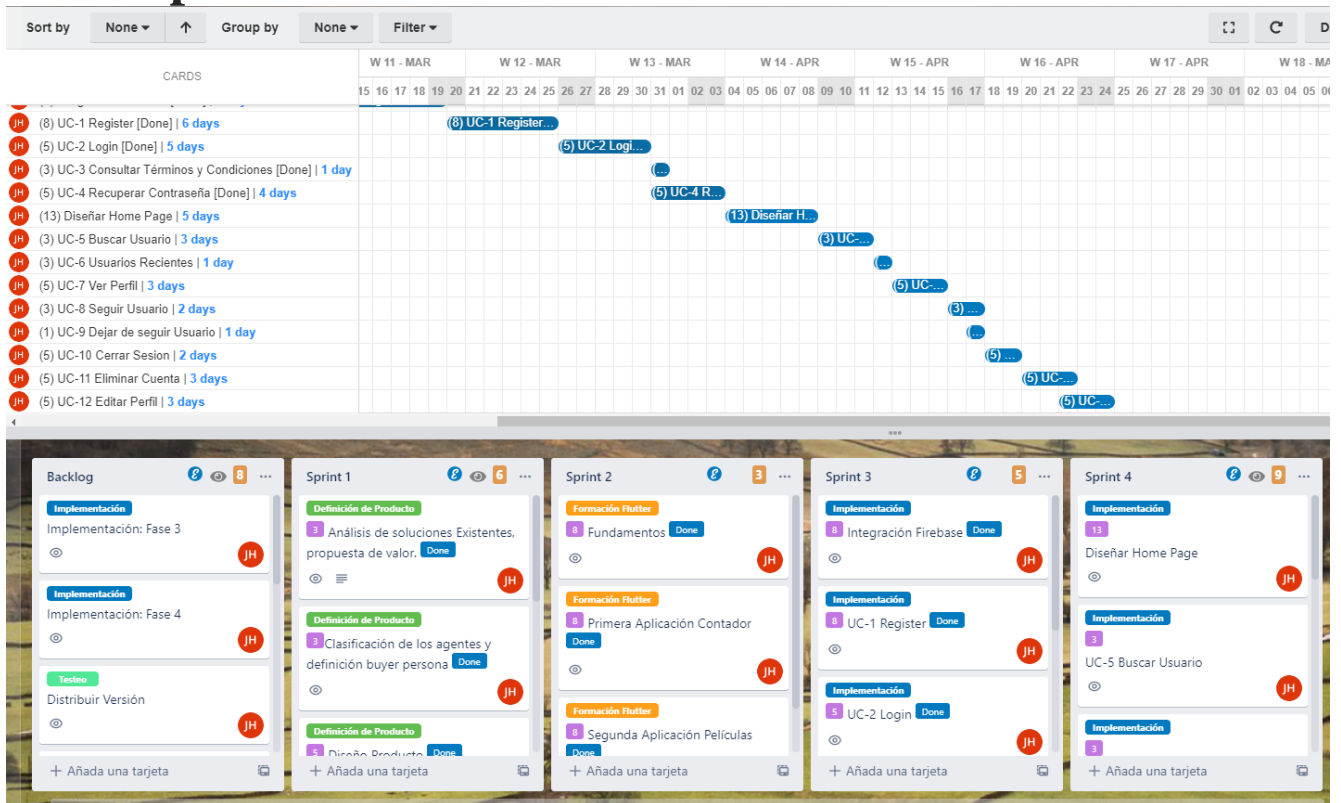


Figura 6.4: Estado de tablero, Sprint 4

Sprint Backlog:

- **Diseñar Home Page (13) Puntos de historia**
- **UC-5 Buscar Usuario (3) Puntos de historia**
- **UC-6 Usuarios recientes (3) Puntos de historia**
- **UC-7 Ver Perfil (5) Puntos de historia**
- **UC-8 Seguir Usuario (3) Puntos de historia**
- **UC-9 Dejar de Seguir Usuario (1) Puntos de historia**
- **UC-10 Cerrar Sesión (5) Puntos de historia**
- **UC-11 Eliminar Cuenta (5) Puntos de historia**
- **UC-12 Editar Perfil (5) Puntos de historia**
- **UC-13 Crear Chat (13) Puntos de historia**

Sprint Goal:

- **Diseñar Home Page (13) Puntos de historia**
- **UC-5 Buscar Usuario (3) Puntos de historia**
- **UC-6 Usuarios recientes (3) Puntos de historia**
- **UC-7 Ver Perfil (5) Puntos de historia**
- **UC-8 Seguir Usuario (3) Puntos de historia**
- **UC-9 Dejar de Seguir Usuario (1) Puntos de historia**
- **UC-10 Cerrar Sesión (5) Puntos de historia**
- **UC-11 Eliminar Cuenta (5) Puntos de historia**
- **UC-12 Editar Perfil (5) Puntos de historia**

Horas estimadas VS definitivas:

Sprint 4			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Diseñar Home Page	10	15	
Buscar Usuario	6	7	
Usuarios recientes	2	2	
Ver Perfil	6	8	
Seguir Usuario	3	5	
Dejar de Seguir Usuario	2	5	
Cerrar Sesion	4	5	
Eliminar Cuenta	4	7	
Editar Perfil	6	12	
	43	66	23

Tabla 6.5: Horas estimadas frente a las realizadas, Sprint 4

6.7. Sprint 5

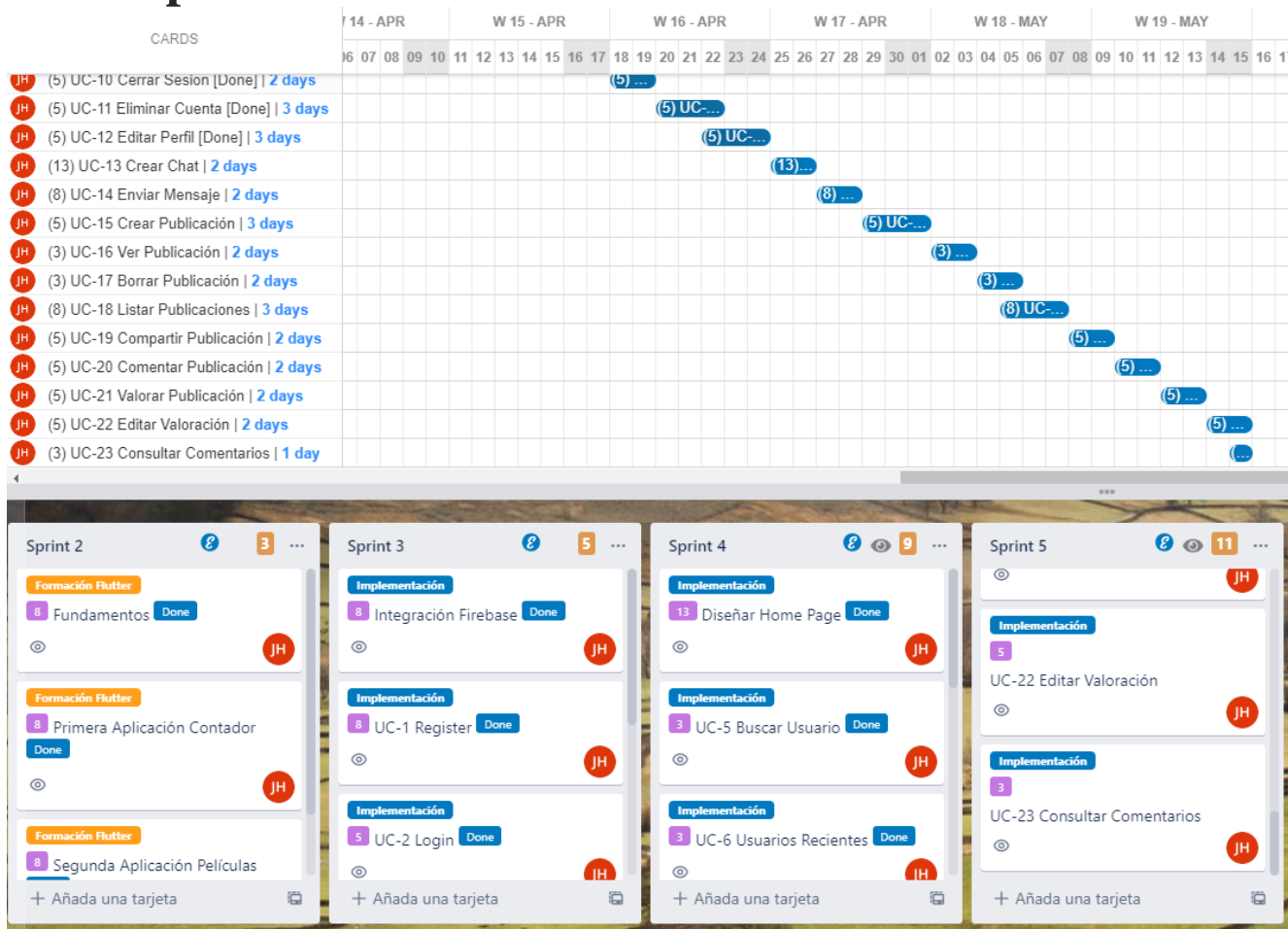


Figura 6.5: Estado de tablero, Sprint 5

Sprint Backlog:

- **UC-13 Crear Chat (8) Puntos de historia**
- **UC-14 Enviar Mensaje (8) Puntos de historia**
- **UC-15 Crear Publicación (5) Puntos de historia**
- **UC-16 Ver Publicación (3) Puntos de historia**
- **UC-17 Borrar Publicación (3) Puntos de historia**
- **UC-18 Listar Publicaciones (8) Puntos de historia**
- **UC-19 Compartir Publicación (5) Puntos de historia**
- **UC-20 Comentar Publicación (5) Puntos de historia**
- **UC-21 Valorar Publicación (5) Puntos de historia**
- **UC-22 Editar Valoración (5) Puntos de historia**
- **UC-23 Consultar Comentarios (3) Puntos de historia**

- **UC-24 Eliminar Comentario (3) Puntos de historia**
- **UC-25 Reportar Publicación (3) Puntos de historia**
- **UC-26 Ocultar Publicación (3) Puntos de historia**
- **UC-27 Borrar Chat (5) Puntos de historia**

Sprint Goal:

- **UC-13 Crear Chat (8) Puntos de historia**
- **UC-14 Enviar Mensaje (8) Puntos de historia**
- **UC-15 Crear Publicación (5) Puntos de historia**
- **UC-16 Ver Publicación (3) Puntos de historia**
- **UC-17 Borrar Publicación (3) Puntos de historia**
- **UC-18 Listar Publicaciones (8) Puntos de historia**
- **UC-19 Compartir Publicación (5) Puntos de historia**
- **UC-20 Comentar Publicación (5) Puntos de historia**
- **UC-21 Valorar Publicación (5) Puntos de historia**
- **UC-22 Editar Valoración (5) Puntos de historia**
- **UC-23 Consultar Comentarios (3) Puntos de historia**

Horas estimadas VS definitivas:

Sprint 5			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Crear Chat	4.5	6	
Enviar Mensaje	4.5	5	
Crear Publicación	5.5	9	
Ver Publicación	4.5	5	
Borrar Publicación	3	3	
Listar Publicaciones	4.5	8	
Compartir Publicación	3	3	
Comentar Publicación	4	4	
Consultar Comentarios	4.5	5	
Valorar Publicación	3	4	
Editar Valoración	2	4	
	43	56	13

Tabla 6.6: Horas estimadas frente a las realizadas, Sprint 5

6.8. Sprint 6

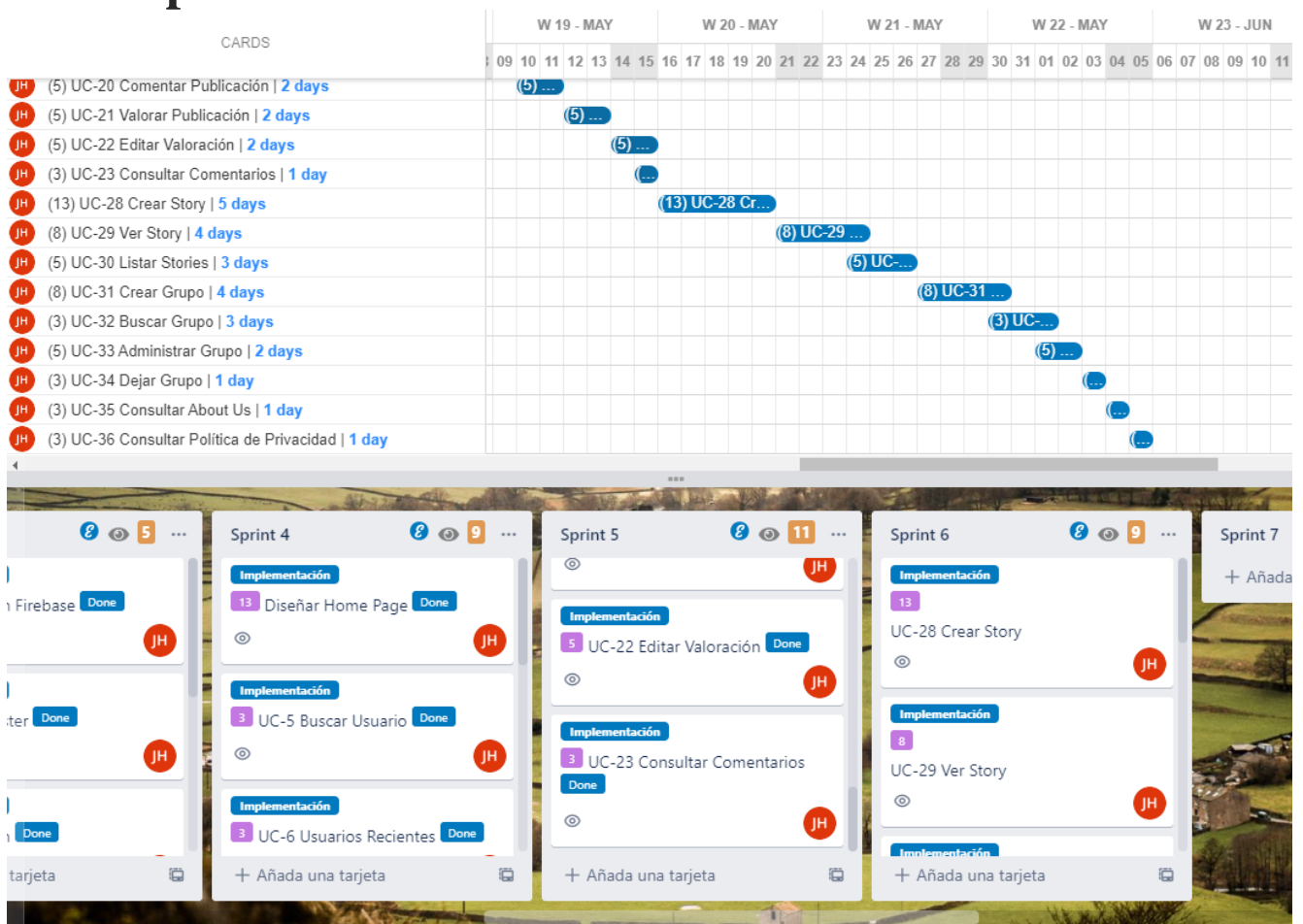


Figura 6.6: Estado de tablero, Sprint 6

Sprint Backlog:

- **UC-28 Crear Story (13) Puntos de historia**
- **UC-29 Ver Story (8) Puntos de historia**
- **UC-30 Listar Stories (5) Puntos de historia**
- **UC-31 Crear Grupo (8) Puntos de historia**
- **UC-32 Buscar Grupo (3) Puntos de historia**
- **UC-33 Administrar Grupo (5) Puntos de historia**
- **UC-34 Dejar Grupo (3) Puntos de historia**
- **UC-35 Consultar About Us (3) Puntos de historia**
- **UC-36 Consultar Política de Privacidad (3) Puntos de historia**
- **UC-37 Borrar Grupo (3) Puntos de historia**
- **UC-38 Borrar Story (3) Puntos de historia**

- **UC-39 Consultar Notificaciones (13) Puntos de historia**
- **UC-40 Seleccionar Idioma (13) Puntos de historia**

Sprint Goal:

- **UC-28 Crear Story (13) Puntos de historia**
- **UC-29 Ver Story (8) Puntos de historia**
- **UC-30 Listar Stories (5) Puntos de historia**
- **UC-31 Crear Grupo (8) Puntos de historia**
- **UC-32 Buscar Grupo (3) Puntos de historia**
- **UC-33 Administrar Grupo (5) Puntos de historia**
- **UC-34 Dejar Grupo (3) Puntos de historia**
- **UC-35 Consultar About Us (3) Puntos de historia**
- **UC-36 Consultar Política de Privacidad (3) Puntos de historia**

Horas estimadas VS definitivas:

Sprint 6			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Crear Story	10	13	
Ver Story	7	7	
Listar Stories	5	4	
Crear Grupo	7	11	
Buscar Grupo	3.5	5	
Administrar Grupo	4.5	8	
Dejar Grupo	2	3	
Consultar Política de Privacidad	2	2	
Consultar About Us	2	2	
	43	55	12

Tabla 6.7: Horas estimadas frente a las realizadas, Sprint 6

6.9. Sprint 7

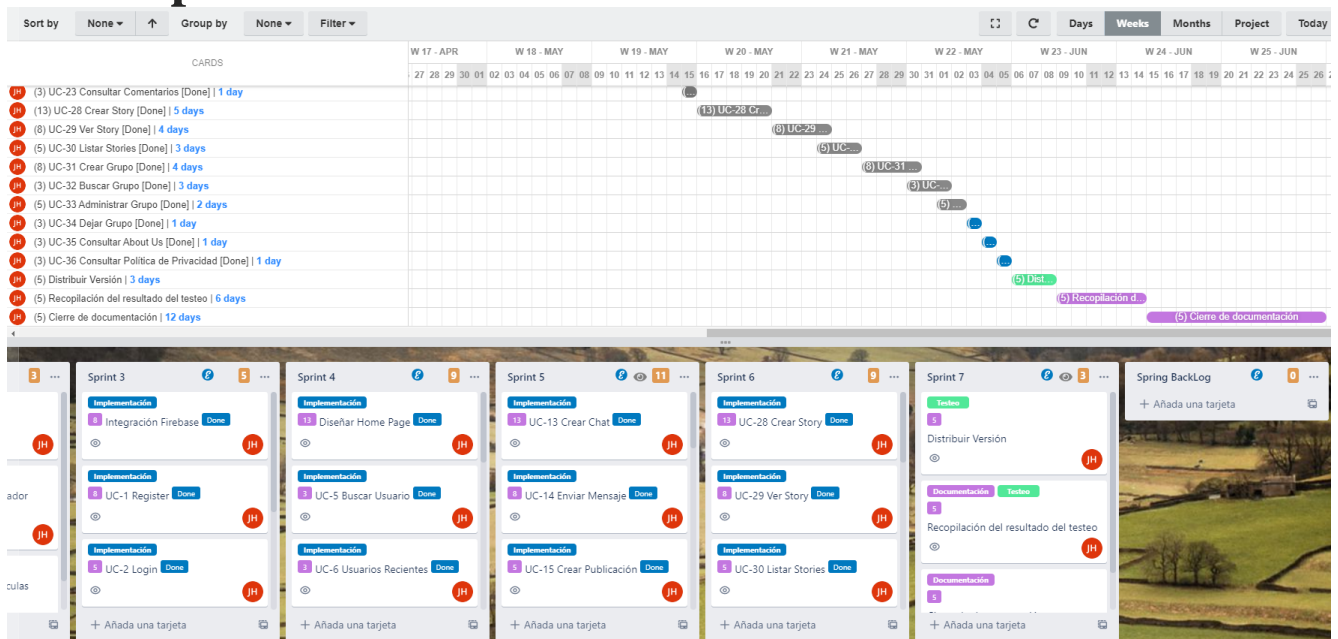


Figura 6.7: Estado de tablero, Sprint 7

Sprint Backlog:

- **Distribuir Versión (5) Puntos de historia**
- **Recopilatorio de Resultados del testeo (5) Puntos de historia**
- **Cierre de documentación (5) Puntos de historia**

Sprint Goal:

- **Distribuir Versión (5) Puntos de historia**
- **Recopilatorio de Resultados del testeo (5) Puntos de historia**
- **Cierre de documentación (5) Puntos de historia**

Horas estimadas VS definitivas:

Sprint 7			
Historia	Horas Estimadas	Horas Realizadas	Diferencia Total
Distribuir Versión	6	6	
Recopilación de Resultados del testeo	13	11	
Cierre de documentación	24	32	
	43	49	6

Tabla 6.8: Horas estimadas frente a las realizadas, Sprint 7

6.10. Análisis

Como resultado de la planificación, hemos obtenido una desviación importante con respecto a la estimación inicial de horas. La visión inicial de capacidades y estimación de horas ha sido muy optimista, no ha sido un tema relacionado con la complejidad de cada tarea, sino más bien que como desarrollador la falta de experiencia y conocimiento de Flutter no ha permitido tener la agilidad deseada.

De este modo la valoración inicial de tareas para realizar durante el TFG ha sido demasiado ambiciosa con lo que el propósito de finalizar una primera versión de MVP no se ha conseguido, quedando varias tareas en el backlog.

Se ha realizado un sobre esfuerzo para conseguir finalizar una primera versión que, aunque no tenga todo lo deseado, si transmite el valor que tiene el proyecto, de ahí que la desviación de horas sea tan importante, 108 sobre 300 que originalmente se habían estimado.

Como lecciones aprendidas quedan que la experiencia nos va a dar mayor conocimiento para realizar las estimaciones en cuanto a tiempo y complejidad de cada tarea, así como para comprender los posibles imprevistos que puedan aparecer. Otra valoración que se tiene en cuenta para el futuro, es que es muy importante dedicar espacios de tiempo útiles y suficientes para cada tarea objetivo, la simultaneidad de tener que gestionar el TFG, trabajo, familia y otras responsabilidades que envuelven a la creación de una empresa ha hecho que en muchos casos la productividad se viera afectada.

6.11. Presupuesto

Cuando hacemos el presupuesto para un cliente, es muy importante señalar que consideramos que contratan a una empresa para ejecutar el proyecto, debemos tener en cuenta una serie de variables, como son el coste de personal, los costes indirectos y el margen de beneficio y el IVA:

6.11.1. Costes de personal

En este se tiene en cuenta que para la ejecución del proyecto la empresa dispone de un programador junior que recibirá el apoyo de un programador senior y el proyecto constará del seguimiento de un jefe de proyectos.

El coste será el resultado de su salario más las aportaciones que la empresa realiza a la seguridad social por hora multiplicado por las horas estimadas para la realización del proyecto por cada uno de trabajadores relacionados.

Para las horas se ha estimado que el Project Manager dada su experiencia, no necesitaría más de ocho horas para hacer el seguimiento de este proyecto, sin embargo, el desarrollador senior debería dedicar 14 horas para dar soporte al programador junior.

El grueso de horas del proyecto las ocupara el programador junior, a las horas ya utilizadas 408, hay que sumarle la estimación de lo que queda pendiente en backlog y las nuevas tareas para la resolución de Issues. De esta manera se estima que podríamos estar hablando de otros dos Sprints, es decir 86 horas más para hacer un total de 494 horas.

6.11.2. Costes indirectos

Los costes indirectos son aquellos que se producen por la propia actividad de la empresa y que no se asocian directamente al proyecto en sí, sin embargo, hay que considerarlos para que el resultado financiero sea consistente.

Para estimarlos deberíamos relativizar estos costes fijos y variables asociados a la existencia de la empresa y repartirlos entre el número y peso que cada proyecto tiene sobre la capacidad productiva de la empresa. Se ha estimado que su valor debería ocupar un 10% de presupuesto de cada proyecto.

Una buena estimación de este valor debería llevarnos a alcanzar el punto de equilibrio.

6.11.3. Margen Neto / Operativo

En este caso estaríamos hablando del porcentaje de beneficio neto con relación a los ingresos totales de las ventas, el profesor de finanzas en la Stern School of Business de Nueva York, Aswath Damodaran ha hecho un estudio sobre los datos financieros de las empresas cotizadas de estados unidos [19] del que podemos extraer que este margen para la industria dedicada al software de aplicaciones tiene un valor de 19,66%.

6.11.4. Presupuesto final

	Salario/hora	Cotización/hora	Horas Requeridas	Coste trabajador
Salario Junior	9.84	2.60	494	6,143 €
Salario Senior	21.86	2.60	14	342 €
Proyect Manager	27.32	2.60	8	239 €
			Costes directos	6,725 €
		10%	Costes indirectos	1,363 €
		19.66%	Margen Operativo	2,680 €
		21%	IVA	2,862 €
			Presupuesto total	13,630 €

Tabla 6.9: Desglose del presupuesto del proyecto

Finalmente, al cliente se le facturaría por un importe de 13.630 euros. A petición del cliente, el presupuesto del proyecto se acompaña de una estimación de costes del servidor en Firebase [20] para el primer año de vida de la aplicación sin carácter vinculante. Estos precios no contemplarían ningún servicio de mantenimiento ni soporte, el encargado sería el cliente.

Realtime DataStore	Mes/usuario	Tamaño bytes	Coste GB mes	Total x mes
Mensajes chat	50	50	5 €	0€
Mensajes Transferidos	600	50	1 €	0€
Cloud FireStore	Mes/usuario	Tamaño bytes		
Publicaciones	4	120		0€
O escritura	12			0€
O lectura	400			0€
O borrado	3			0€
Cloud Store	Mes/usuario	Tamaño Mb		
Fotos	12	0.8		1.82€
Transferencia	150	0.6		77€
Operaciones	620			2.5€
Cloud Functions	Mes			
Invocaciones	10000000			0 €
GB-Segundo	1200000			0 €
Tiempo CPU				0 €
Herramientas Redes				
Salida	30			3 €
Minutos Cloud Build	450			1 €
Almacenamiento contenedores				0 €
Egress Cost	N2 Belgica			
CPU	11.42			11.42€
GB memoria	1.53			12.24€
TOTALES 1º Año			1307,76€	109 €

Tabla 6.10: Desglose estimación costes Firebase, Año 1º

Capítulo 7

7. Documentación Técnica

7.1. Requisitos del sistema

Con los requisitos se intenta hacer una descripción completa [21] con lo que el sistema debe hacer. Estos requisitos se suelen extraer de los casos de uso, la mayoría de los requisitos no funcionales existen por restricciones en el diseño o por estándares de calidad.

7.1.1 Requisitos funcionales

Los requisitos funcionales definen una función del sistema o de sus componentes.

FN-1: El sistema debe permitir al usuario crear publicaciones.

FN-2: El sistema deber facilitar al usuario establecer chat con otro usuario o con otros usuarios.

FN-3: El sistema da la posibilidad al usuario de crear Stories.

FN-4: El sistema permite al usuario ver sus publicaciones y las de los usuarios que le son visibles.

FN-5: El sistema debe dar la posibilidad de comentar sus publicaciones y de usuarios que son visibles para él.

FN-6: El sistema debe permitir a los usuarios ver el perfil de todos los usuarios.

FN-7: El sistema debe permitir al usuario registrarse.

FN-8: El sistema debe de poder permitir al usuario recuperar la contraseña.

FN-9: El sistema debe permitir al usuario cerrar sesión.

FN-10: El sistema debe permitir al usuario borrar la cuenta.

FN-11: El sistema debe facilitar el acceso a ver los términos y condiciones.

FN-12: El sistema debe facilitar la edición del perfil de propio usuario.

FN-13: El sistema debe permitir a cualquier usuario seguir o dejar de seguir a otro usuario.

FN-14: El sistema dará la opción de buscar usuarios

FN-15: El sistema debe permitir a las búsquedas de usuario ordenándolos por su fecha de ingreso.

FN-16: El sistema permitirá a los usuarios crear grupos de chat y agregar a sus seguidores.

FN-17: El sistema facilitara la opción de buscar grupos.

FN-18: El sistema listara los grupos a los que el usuario pertenece.

FN-19: El sistema permitirá al usuario perteneciente a un grupo administrar el grupo.

FN-20: El sistema debe permitir a los usuarios de un grupo, abandonar el mismo.

FN-21: El sistema debe permitir al creador de un grupo poder borrarlo.

FN-22: El sistema debe facilitar a los usuarios enviar un mensaje a través del chat o grupo.

FN-23: El sistema debe de dar al usuario que ha creado una publicación la opción de borrar esa publicación.

FN-24: El sistema debe de dar la opción al usuario de compartir las publicaciones que le son visibles.

FN-25: El sistema debe permitir al usuario consultar las notificaciones recibidas.

FN-26: El sistema debe dar la opción al usuario de consultar la política de privacidad.

FN-27: El sistema debe dar la opción al usuario de consultar el “About Us” que explica el proyecto.

FN-28: El sistema debe dar la opción al usuario de cambiar el idioma al iniciar la aplicación.

FN-29: El sistema debe listar las publicaciones que son visibles para el usuario en la pantalla de Home.

FN-30: El sistema debe de dar la opción al usuario de reportar una publicación.

FN-31: El sistema debe de dar la opción al usuario de ocultar una de las publicaciones que le son visibles.

FN-32: El sistema debe de dar la opción al usuario de ocultar una de las publicaciones que le son visibles.

FN-33: El sistema debe de dar la opción al usuario de poder eliminar sus comentarios.

FN-34: El sistema facilitara al usuario a acceder a los comentarios de las publicaciones que le son visibles.

FN-35: El sistema dará la opción a los usuarios de poder valorar aquellas publicaciones que le son visibles.

FN-36: El sistema dará la opción a los usuarios de poder editar las valoraciones que ha realizado sobre las publicaciones.

FN-37: El sistema debe permitir a los usuarios poder ver las historias de usuarios a los que sigue.

FN-38: El sistema facilitará a los usuarios listar historias de los usuarios en la pantalla de Home

FN-39: El sistema facilitará a cualquiera de los usuarios que componen un chat, el poder borrarlo.

FN-40: El sistema facilitara a cualquier usuario borrar una de sus historias.

FN-41: El sistema debe permitir el acceso al usuario mediante un email y contraseña.

FN-42: El sistema debe dar reportar la creación de comentarios, valoraciones, mensajes y seguimiento al usuario que sea creador de una publicación, participa en un chat o grupo o comienza a ser seguido.

7.1.2. Requisitos de Información

Requisitos de información, definen la información que ha de persistir en el sistema para con su gestión poder dar soporte a los requisitos funcionales.

RI-1: El sistema almacenara información sobre las credenciales que dan acceso al usuario, correo, nombre, contraseña.

RI-2: El sistema almacenara los mensajes que intercambian los usuarios.

RI-3: El sistema almacenara los id de los usuarios a los que un usuario sigue.

RI-4: El sistema almacenara las imágenes que forman parte de la publicación, las publicaciones tendrán un campo que contendrá una URL que da acceso a estas imágenes.

RI-5: El sistema debe guardar la información relativa a usuario que genera algún tipo contenido, esto es su id, descripción, y momento de creación, así como documentos adjuntos.

RI-6: El sistema debe guardar la información que sobre cada evento que genera una notificación, quien la crea, quien la recibe, y el tipo de evento en si mismo.

RI-7: El sistema clasificara los eventos notificables, en: like, comment, follow y message y guardara la información correspondiente.

RI-8: El sistema identificara inequívocamente en una tabla aquellos mensajes que han sido reportados.

RI-9: El sistema debe almacenar la información de descripción e imágenes adjuntas de las historias.

7.1.3. Requisitos no Funcionales:

Estos requisitos especifican los criterios con los que se pueden evaluar las operaciones que realiza el sistema, su funcionamiento.

RNF-1: El sistema debe aplicar métodos de seguridad en el acceso.

RNF-2: Un usuario no puede agregar a otro a un grupo si este no le sigue primero.

RNF-3: Un usuario no puede ver publicaciones de usuarios a los que no sigue.

RNF-4: El flujo de pantallas y el UI debe ser fácilmente comprensible.

RNF-5: Deben existir versiones de la aplicación para Android y iOS.

RNF-6: El tiempo de respuesta del sistema no sobrepasara los dos minutos.

RNF-7: El porcentaje de fallos del sistema no puede superar el 3,5%.

7.2. Casos de uso

Los casos de uso son artefactos que nos permiten elaborar una secuencia de acciones que se han de seguir para conseguir un resultado de valor observable. En nuestro caso, solo tenemos un actor que es el usuario final, el cual, va a realizar una serie de acciones con el objetivo de conseguir resultados relacionados con:

- Su cuenta, perfil y sesión.
- Grupos de usuarios y chats.
- Publicaciones, comentarios y Stories.
- Quienes somos, términos y política de privacidad.

7.2.1. Diagrama de casos de uso

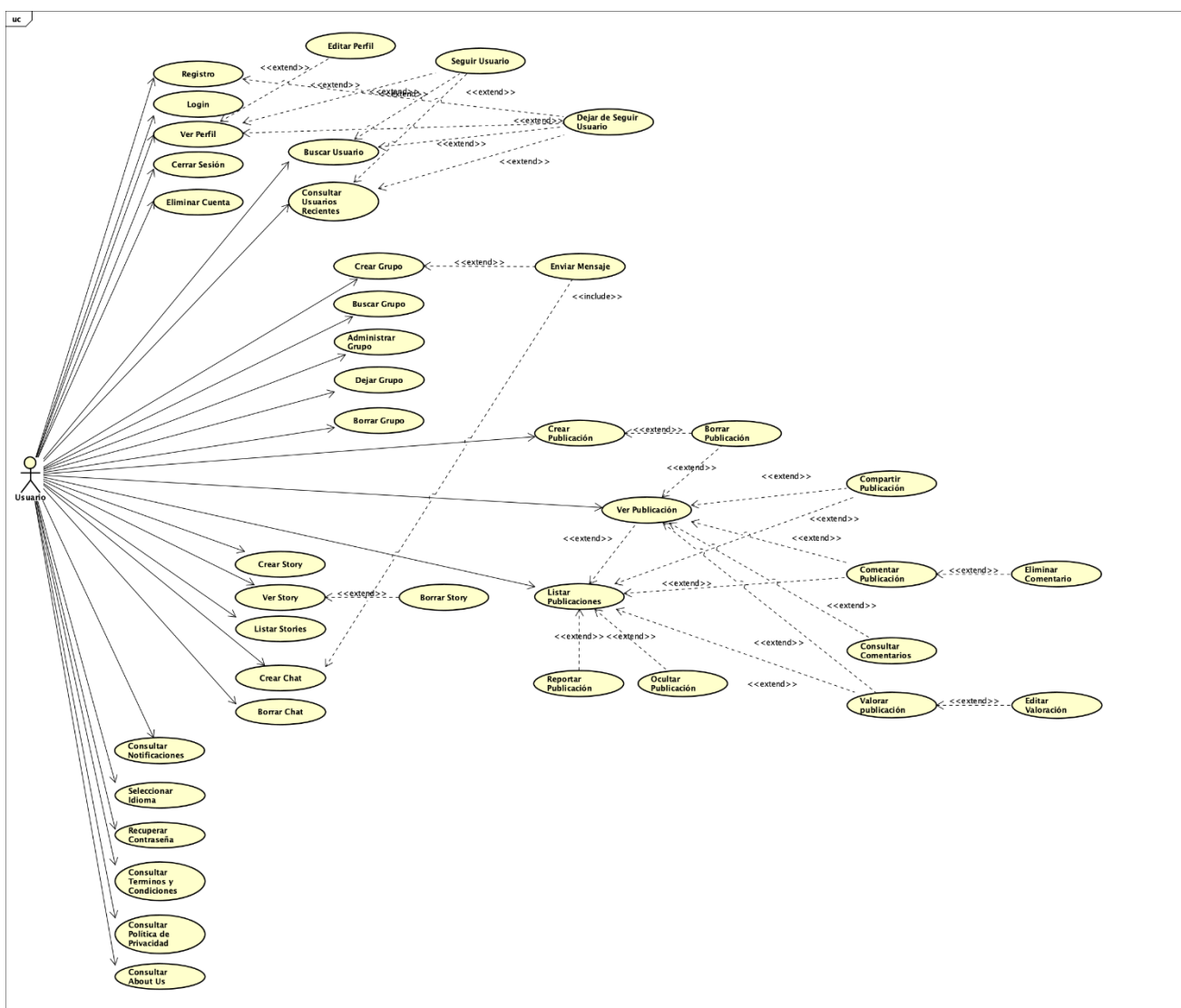


Figura 7.1: Diagrama de casos de uso

7.2.2. Caso de Uso: Login

Nombre e Id del Caso de Uso	UC-1 Login
Actor	Usuario
Descripción	El usuario completa los campos usuario y contraseña para poder acceder a usar la aplicación.
Precondiciones	PRE-1. El Usuario no ha iniciado sesión en la aplicación.
Postcondiciones	POST-1. Escenario de Éxito: El usuario inicia sesión en el sistema y puede acceder a su uso.
Flujo Normal	<p>FN1. El actor introduce el email y contraseña</p> <p>FN2. El sistema a través del servicio de autenticación comprueba las credenciales y carga la vista de Home.</p>
Flujo Alternativo 1	<p>FA1. El usuario selecciona la opción de Login con Google Account.</p> <p>FA2. El sistema verifica el acceso al sistema de autenticación, y muestra la pasarela de acceso a través del sistema de autenticación con Google Account.</p> <p>FA3. El actor introduce el email y contraseña</p> <p>FA4. El sistema de autenticación comprueba si las credenciales son correctas y se lo verifica al sistema a continuación esta carga la vista de Home.</p>
Flujo Alternativo 2	FA1. Las credenciales introducidas son incorrectas, el sistema muestra un mensaje de error y el caso de uso queda sin efecto.
Excepciones	EA4. El sistema de autenticación informa al sistema que la autenticación es incorrecta, este notifica al usuario este error y el caso de uso queda sin efecto.
Prioridad	Alta

7.2.3. Caso de Uso: Registro

Nombre e Id del Caso de Uso	UC-2 Registro
Actor	Usuario
Descripción	El usuario completa una serie de campos de información y acepta las condiciones de uso para crear un nuevo perfil y así poder acceder a la red social.
Precondiciones	PRE-1. El Usuario no existe en el sistema.
Postcondiciones	POST-1. Escenario de Éxito: El usuario aparece como nuevo usuario en el sistema.
Flujo Normal	<p>FN1. El usuario selecciona la opción de Registrarse en la Login page.</p> <p>FN2. El sistema presenta una pantalla nueva que consta de un formulario con los campos de registro y una opción con los términos de uso.</p> <p>FN3. El usuario relleno correctamente los campos y acepta los términos y condiciones y pulsa el botón Registrarse ahora.</p> <p>FN4. El sistema crea un perfil con la información aportada y presenta una pantalla nueva que da la opción a subir un avatar.</p> <p>FN5. El usuario sube un avatar.</p> <p>FN6. El sistema notifica que se ha subido a base de datos correctamente.</p> <p>FN7. El usuario pulsa siguiente.</p> <p>FN8. El sistema muestra una nueva pantalla con los últimos usuarios registrados para poder seguirlos.</p> <p>FN9. El usuario selecciona alguno de estos usuarios y pulsa siguiente.</p> <p>FN10. El sistema persiste la selección en base de datos y pasa a la siguiente y muestra el Home page.</p>

Flujo Alternativo 1	FA4. El sistema comprueba que hay campos sin completar o mal formados e informa al usuario del error mediante una notificación en pantalla. El caso de uso retorna al paso FN3
Flujo Alternativo 2	FA4. El sistema comprueba que el nombre de usuario o el correo electrónico ya están en uso, y notifica del error al usuario. El caso de uso retorna al paso FN3
Flujo Alternativo 4	FA.6. El usuario no sube avatar, directamente pasa al paso FN7
Flujo Alternativo 5	FA8. El usuario no selecciona usuarios y pulsa siguiente directamente. FA9. El sistema muestra el Home page.
Excepciones	E1. El usuario ha dejado algún campo sin rellenar. E2. El usuario no ha aceptado los términos de uso.
Prioridad	Alta

7.2.4. Caso de Uso: Crear Publicación

Nombre e Id del Caso de Uso	UC-3 Crear Publicación
Actor	Usuario
Descripción	El usuario crea realiza una publicación en la red social que consta de descripción y en si el lo elige una foto.
Precondiciones	PRE-1. El Usuario a iniciado sesión y ha accedido correctamente.
Postcondiciones	POST-1. Escenario de Éxito: El actor ha creado una nueva publicación.
Flujo Normal	<p>FN1. El usuario selecciona la opción de crear una publicación.</p> <p>FN2. El sistema presenta una pantalla nueva que consta de un cuadro de texto y una opción para adjuntar una foto.</p> <p>FN3. El usuario relleno correctamente los campos y pulsa el botón Publicar.</p> <p>FN4. El sistema crea la publicación en el servidor y devuelve al usuario a la pantalla de Home.</p>
Flujo Alternativo 1	FA4. El sistema comprueba que hay campos sin completar o mal formados e informa al usuario.
Prioridad	Alta

7.2.6. Caso de Uso: Ver Perfil

Nombre e Id del Caso de Uso	UC-3 Ver Perfil
Actor	Usuario
Descripción	El usuario accede a un perfil para ver la información disponible.
Precondiciones	PRE-1. El usuario se encuentra registrado y ha iniciado sesión con éxito en el sistema.
Postcondiciones	POST-1. Escenario de Éxito: El usuario visualiza la información del perfil que ha seleccionado.
Flujo Normal	FN1. El usuario selecciona la opción de ver perfil. FN2. El sistema presenta en una pantalla la información del perfil seleccionado.
Prioridad	Media

7.3. Modelo de dominio

El modelo de dominio plantea representar a través de un vocabulario y conceptos clave el dominio del problema [22]. En él se identifican las relaciones y restricciones entre entidades que pertenecen al ámbito del problema.

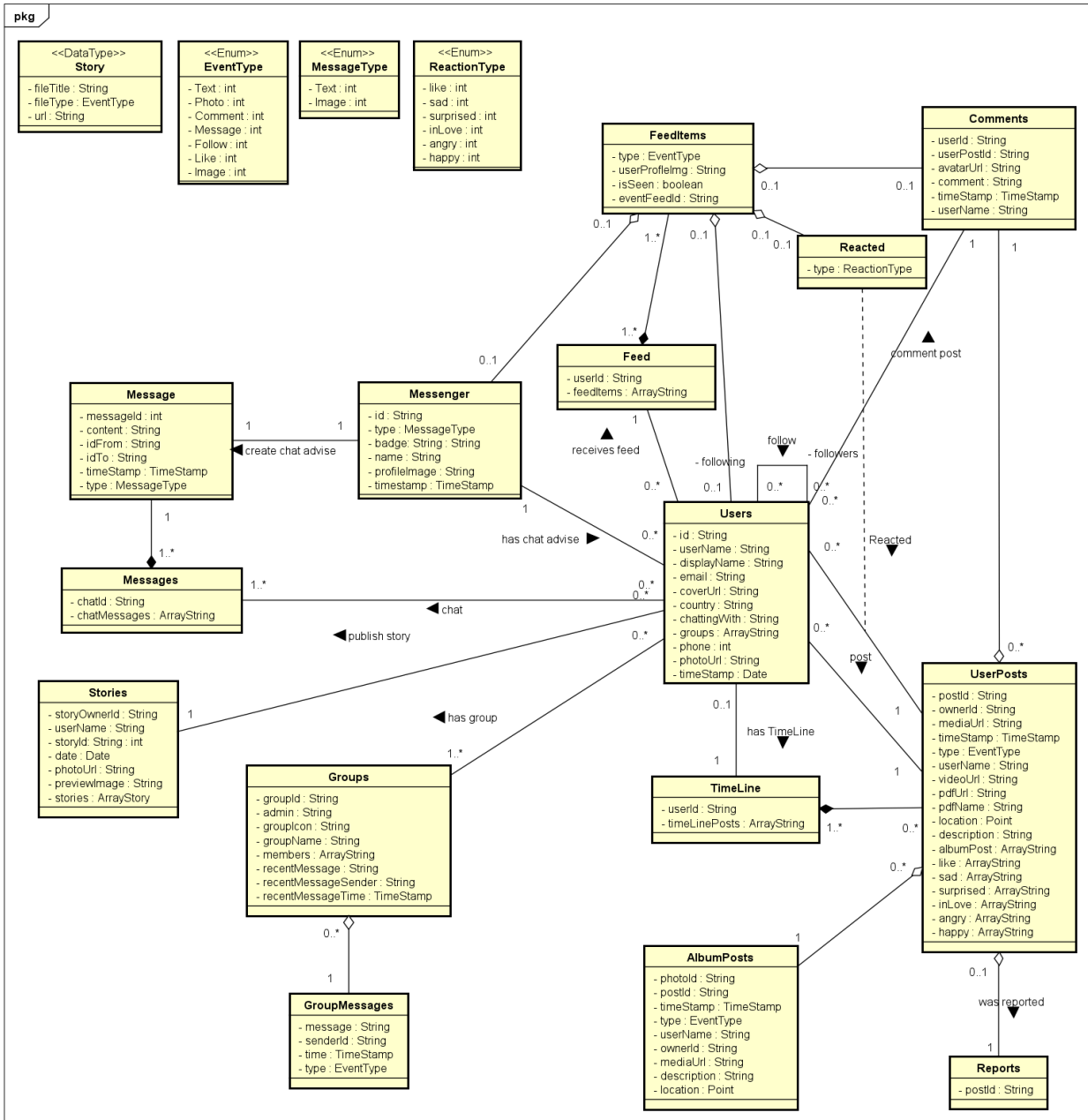


Figura 7.2: Modelo de dominio

- Los **usuarios** pueden tener entre cero o más chats, publicaciones, grupos o historias.
- Las **publicaciones** pueden tener varios comentarios y a su vez cada comentario solo será creado por una persona para un post.
- El usuario puede recibir ninguna o varias **notificaciones** por mensajes, reacciones, comentarios o por que otro usuario ha comenzado a seguirle.

7.4. Diagramas de secuencia

Con los diagramas de secuencia se quiere representar la secuencia de eventos entre el usuario y el sistema cuando se desarrolla un caso de uso. Dada la complejidad y extensión de los diagramas de secuencia, se muestran los casos de uso más representativos.

7.4.1. Diagrama de secuencia del Caso de uso: Login

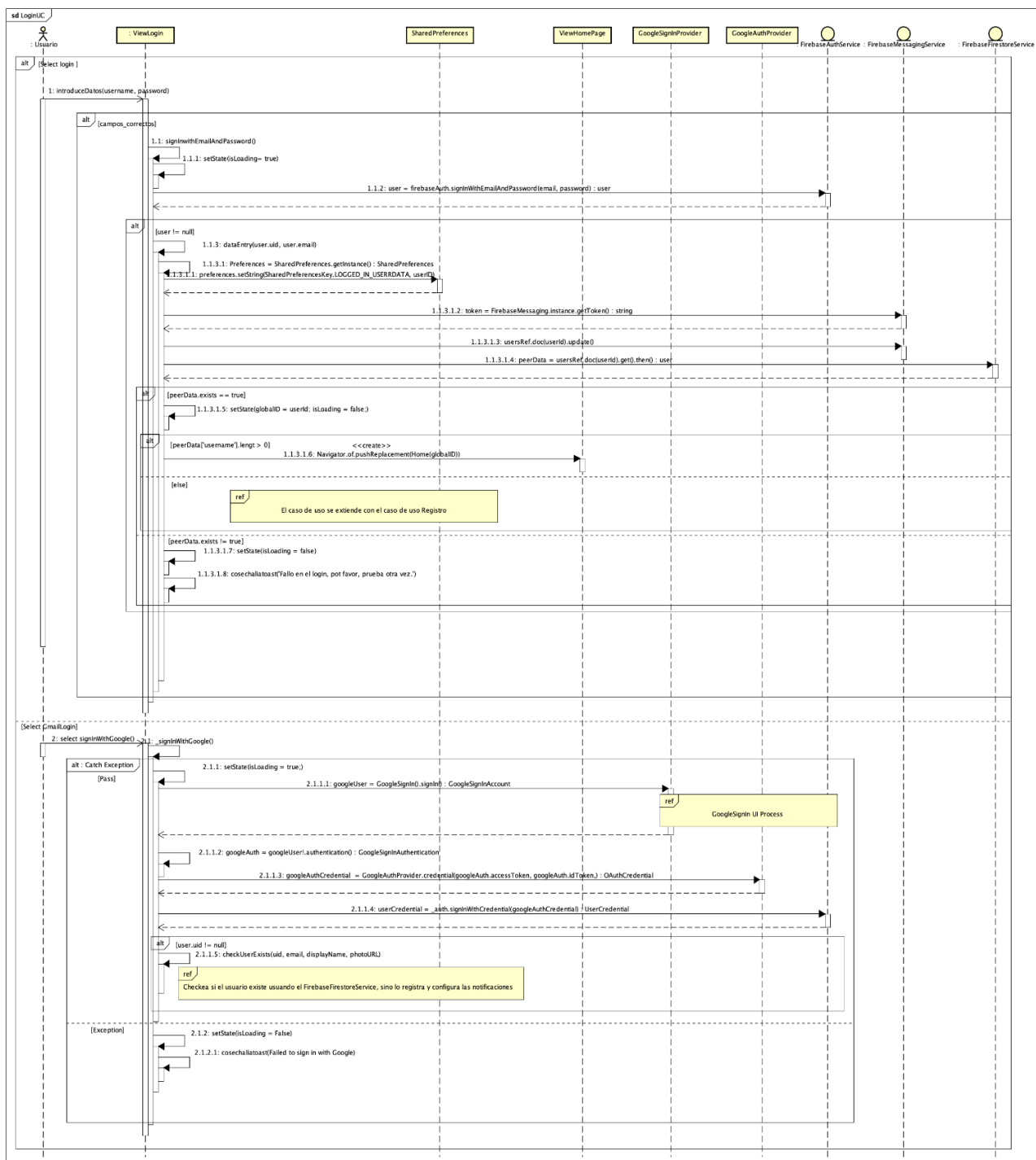


Figura 7.3: Diagrama de secuencia del caso de uso Login

7.4.2. Diagrama de secuencia del Caso de uso: Registro

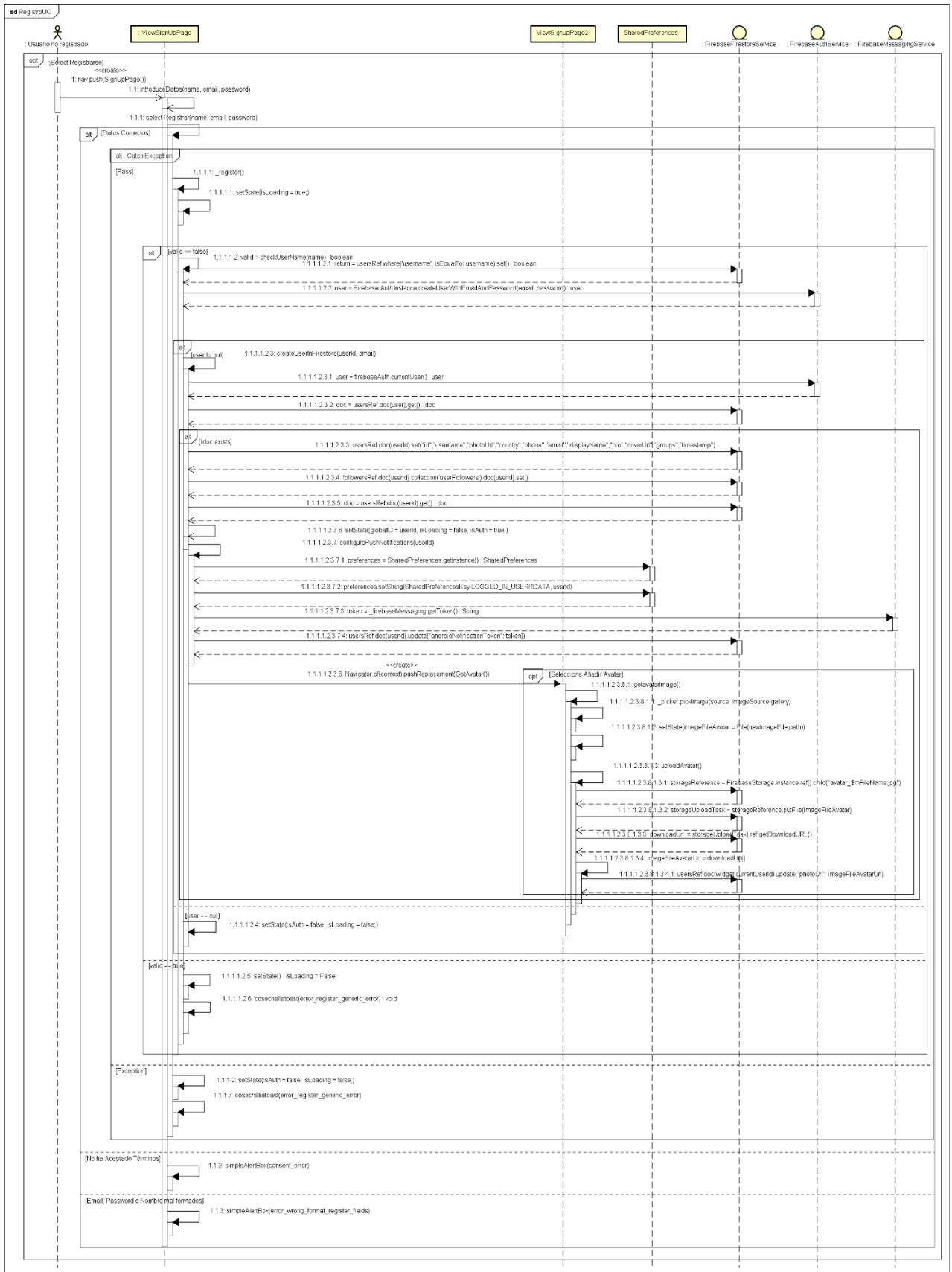


Figura 7.4: Diagrama de secuencia del caso de uso Registro

7.4.3. Diagrama de secuencia del Caso de uso: Ver Perfil

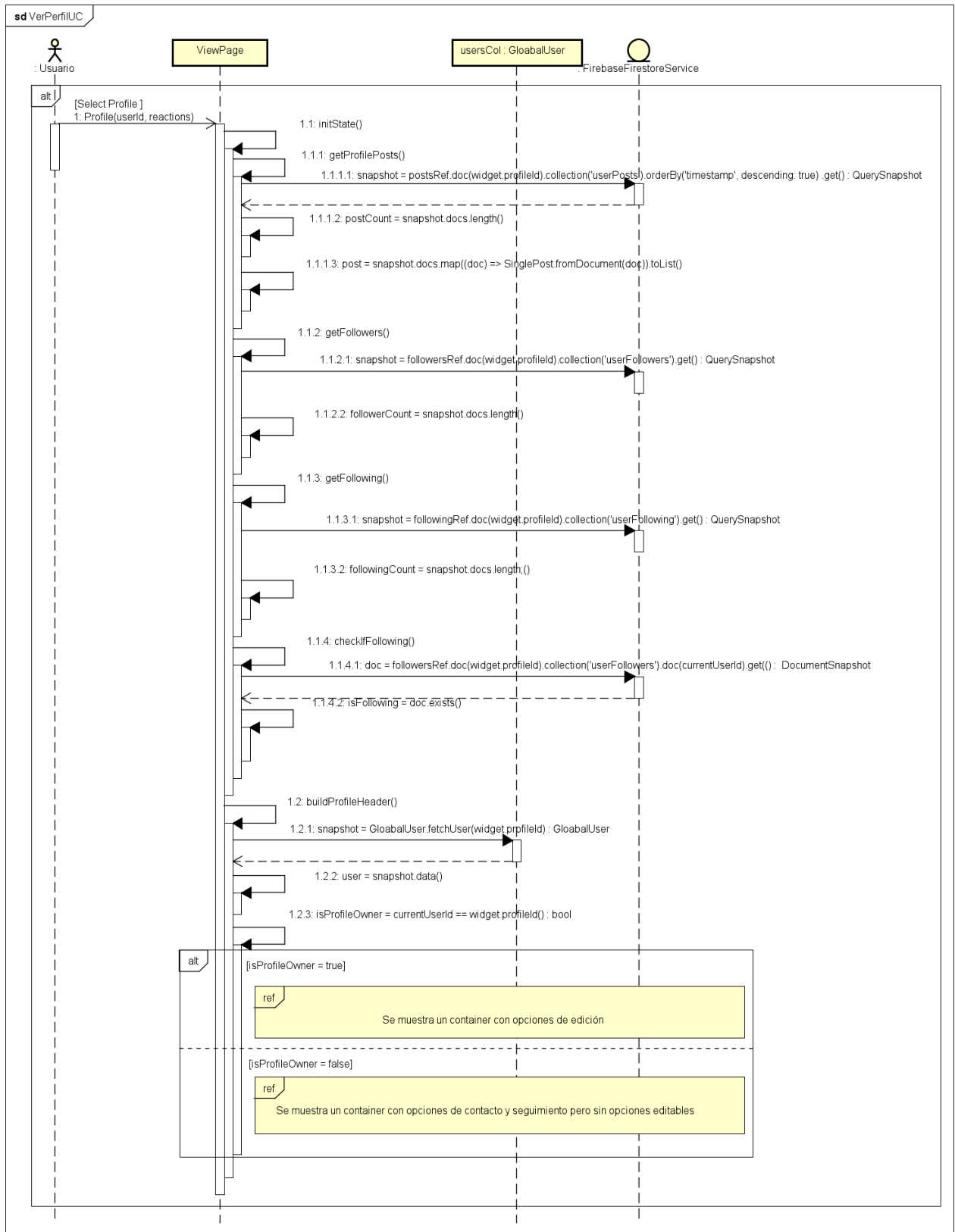


Figura 7.5: Diagrama de secuencia del caso de uso Ver Perfil

7.4.4. Diagrama de secuencia del Caso de uso: Crear Publicación

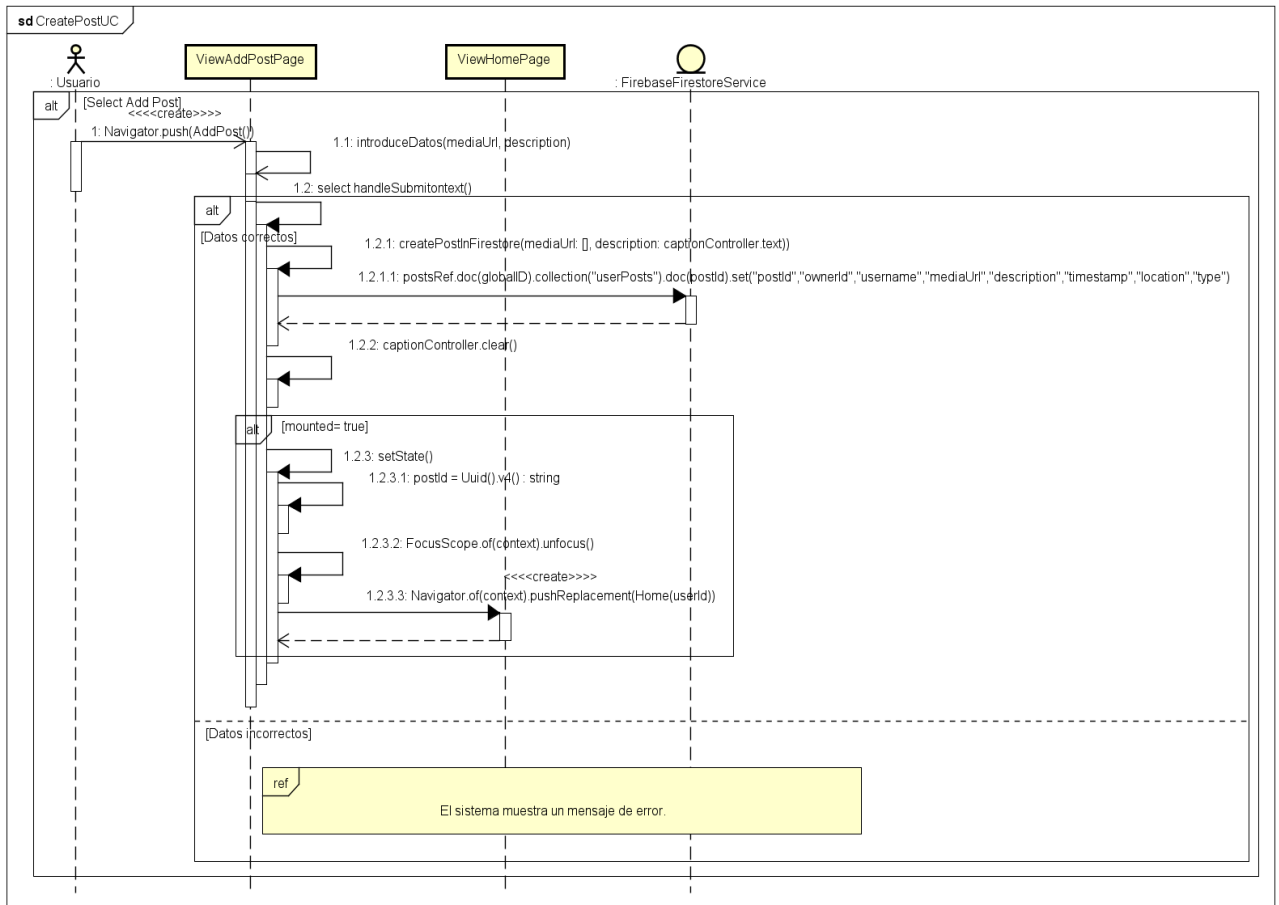


Figura 7.6: Diagrama de secuencia del caso de uso Crear Publicación

7.5. Diagrama de paquetes

El diagrama de paquetes nos permite hacer la representación estructural del software desarrollado [21], en este caso tenemos dos elementos principales, las páginas que serían principalmente las vistas, y después los widgets personalizados, que se utilizan para componer algunas de estas. Son estas dos entidades las que componen la vista y de ahí su relación con todas las demás.

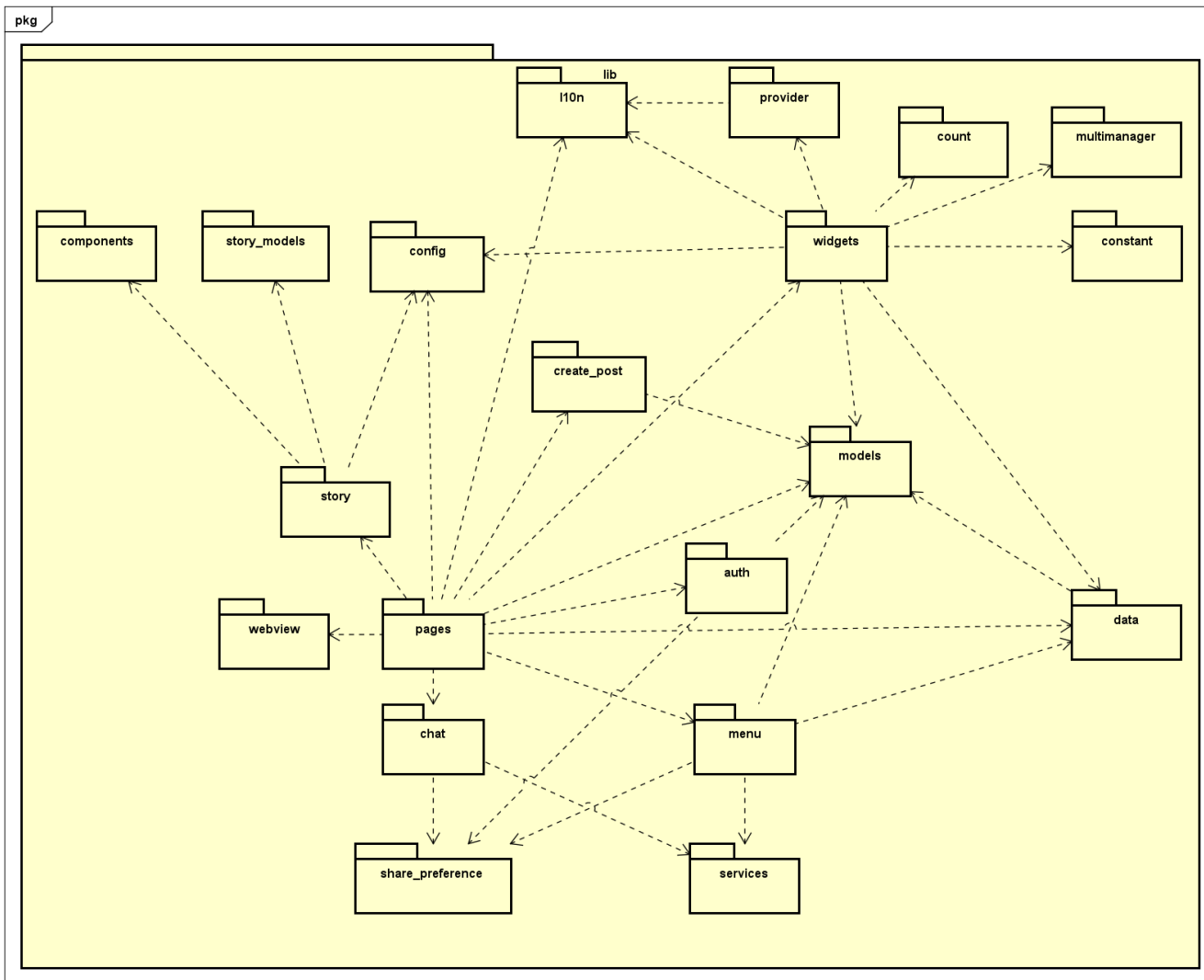


Figura 7.7: Diagrama de paquetes

7.6. Diagrama de despliegue

En el diagrama de despliegue vemos la arquitectura de ejecución del sistema [23], en este caso, los flujos que recorre la información entre las aplicaciones y el servidor Firebase.

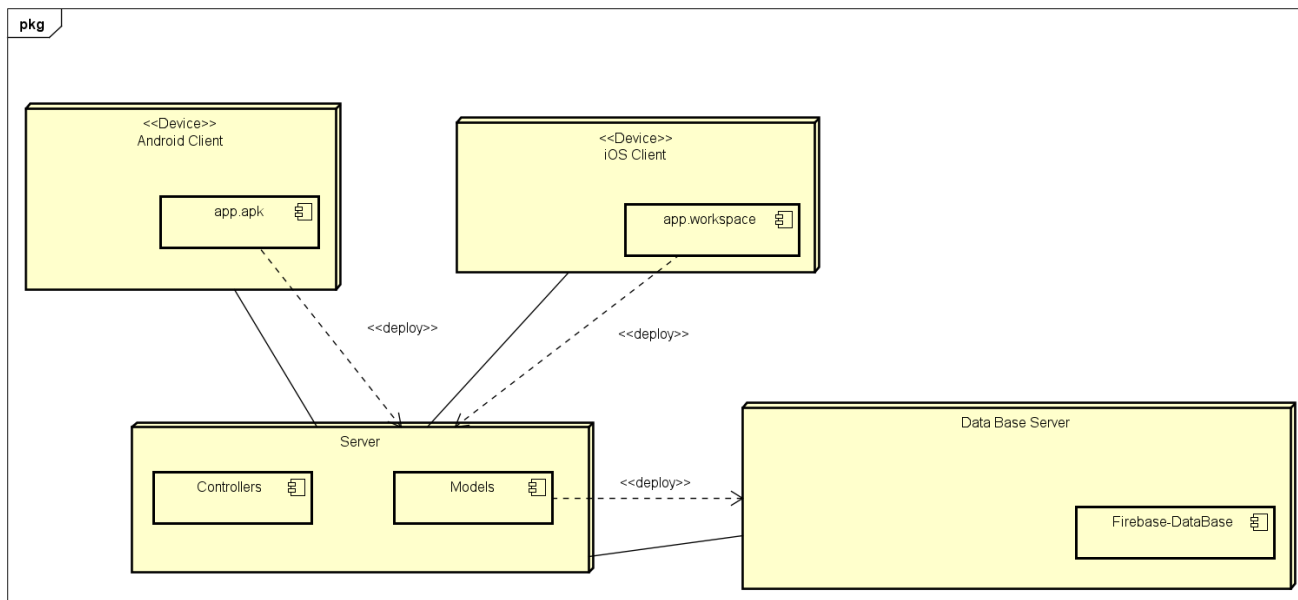


Figura 7.8: Diagrama de despliegue

7.7. Diseño

7.7.1. Arquitectura

A la hora de tomar la decisión de que arquitectura aplicar, se ha tenido en cuenta la complejidad de cada uno de los patrones [23] y su necesidad.

El patrón BLoC [24] aunque es uno de los que más se aplica actualmente y a buen seguro será el que aplique en el futuro para la realización de otros proyectos no se ha tenido en cuenta, ha sido así por la falta de experiencia y el riesgo de desviaciones en la planificación que podrían darse.

De manera que por su facilidad de aplicación y familiaridad se ha implementado el patrón MVC [25]. Este patrón tiene tres componentes:

Vista: Es esa parte que se encarga del diseño y representación.

Controlador: Contiene la lógica que actualiza el modelo y / o vista en respuesta a las acciones de los usuarios en la vista.

Modelo: En él se definen los datos que van a manejarse en la aplicación, normalmente también se le otorga la misión de notificar los cambios de aquellos datos que requieren de una actualización en la vista.

Cuando realizamos una aplicación, a veces cabe la posibilidad de aplicar varios patrones según la funcionalidad que queremos implementar. En este caso la internacionalización de la app, a requerido de la aplicación del patrón Provider [26], este patrón nos permite sacar la lógica de los widgets y concentrarla en un ChangeNotifier que se va a encargar de avisar a todos estos widgets. Para el proyecto era lo más práctico, si un usuario cambia de idioma, este ChangeNotifier va a cambiar el idioma para todos esos Widget de manera que el Widget mostrara el string que en el idioma correspondiente

Otro detalle muy importante para el correcto funcionamiento de la aplicación es el uso de Firebase, esto nos ha permitido cuando es necesario, establecer una relación directa entre los datos que se están mostrando en pantalla y su estado en base de datos, es que Flutter nos proporciona un Widget especial, StreamBuilder que crea un vínculo entre la información mostrada y la información en Firebase, generando una actualización inmediata en el momento que se produce algún cambio en el servidor.

```
return StreamBuilder<QuerySnapshot>(
  stream: commentsRef
    .doc(postId)
    .collection('comments')
    .orderBy("timestamp", descending: false)
    .snapshots(),
```

Figura 7.9: Ejemplo de StreamBuilder

En este caso un cambio en comentarios de la publicación con id= postId, generara una actualización en la vista.

Capítulo 8

8. Pruebas

Para las pruebas se ha realizado un testeo de usuarios, se ha instalado la aplicación en distintos dispositivos para que pudieran experimentar con la aplicación de primera mano, idealmente este enfoque aporta información relacionada con UX y el UI [27], de manera que permite acercar mucho más el producto final a lo que el usuario idealiza antes si quiera de que este llegue a mercado.

8.1. Testeo con usuarios

La estrategia ha sido que algunos testers primero hicieran toma de contacto con la aplicación sin ningún tipo de añadido, nada más que una descripción de lo que pretende el producto:

“Red social para que toda la gente del campo tenga un medio propio para comunicarse.”

A un segundo grupo, se les ha hecho la misma descripción, pero a la hora de tomar contacto con la aplicación se les ha invitado a realizar una serie de tareas:

- **Registro.**
- **Login.**
- **Recuperar contraseña.**
- **Crear una publicación.**
- **Crear una historia.**
- **Seguir un usuario.**
- **Chatear con un usuario.**
- **Crear un grupo con usuarios.**
- **Consultar notificaciones.**

8.2. Resultados

8.2.1. Fallos de comprensión

En términos generales, se ha podido apreciar al acompañar al usuario que algunas pantallas no son del todo claras para ellos:

Seguir usuarios en el registro:

El mensaje no es claro, el título de pantalla es Usuarios Recientes, los usuarios sugieren que diga, por ejemplo, Usuarios que podrías conocer o Usuarios que puedes seguir. También sugieren otra opción Usuarios cerca que puedes seguir y que sean los que se encuentran cerca.

Registro sin repetición de Contraseña:

Con el propósito de agilizar el acceso y teniendo en cuenta que parte de los usuarios son gente mayor, solo se ha dejado un campo de Contraseña para recibir sugerencias durante las entrevistas.

Como conclusión los usuarios prefieren tener que repetir la contraseña, pero también prefieren ver lo que escriben.

Crear una publicación me lleva al lugar inadecuado:

Al crear una publicación la app te lleva a tu perfil que es donde aparecen todas las publicaciones del usuario. El usuario desea que le lleve a una pantalla con sus publicaciones o a la pantalla de Home donde aparecería como la última en la línea temporal de publicaciones.

Al crear una publicación no se actualiza en perfil:

Cuando publicamos y retornamos a la pantalla de perfil, esta no se actualiza para mostrar la última.

Falta una opción para crear historias:

En la pantalla en la que consultas las historias no solo la tuya sino la de tus amigos, falta una opción para publicar historias, solo aparece en el perfil o home, hay lo ven más natural.

Crear una historia me lleva al lugar inadecuado:

El usuario desea ir a la pantalla dedicada a historias.

Acceso a fotos directo:

Cuando agregamos una imagen no podemos hacer directamente una foto. Detalle muy importante, hay que adaptarse a esa necesidad.

8.1.2. Fallos a nivel de UI

No se ve la barra de estado del móvil:

Es algo que todos los usuarios han echado en falta pero que se me ha pasado por alto, no hay contraste entre la barra de estado y el color de background en algunas pantallas. La información por las que se dan cuenta es por la hora y/o por el estado de la batería del móvil.

Icono de envío distinto:

El icono de envío es distinto entre la pantalla de comentarios y la de chat.

El giro de pantalla no está bloqueado:

Esto solo lo ha notificado un usuario, a la hora del desarrollo no se ha tenido en cuenta el girar la pantalla en lo que sería pasar de Portrait a usarlo como Landscape. De manera que en algunas pantallas la distribución del UI se descoloca.

Alguna pantalla muestra el avatar de usuario en cuadrado en vez de en círculo.

La Publicación no te lleva a los comentarios:

Hay dos accesos a los comentarios en la tarjeta dedicada a la publicación, al principio iba a ir hay pero se me paso quitarlo, hay que retirarlo.

Sin imagen en la notificación:

Uno de los usuarios recibe la notificación sin imagen adjunta, esto no ocurre en el resto de los dispositivos. Aquí es donde se ve que hay diferencias entre las capas de personalización “ROMs” para Android que existen [28]. Este usuario tiene un Vivo Y21s con sistema operativo Android 11 y capa de personalización **FunTouch OS 11.1**.

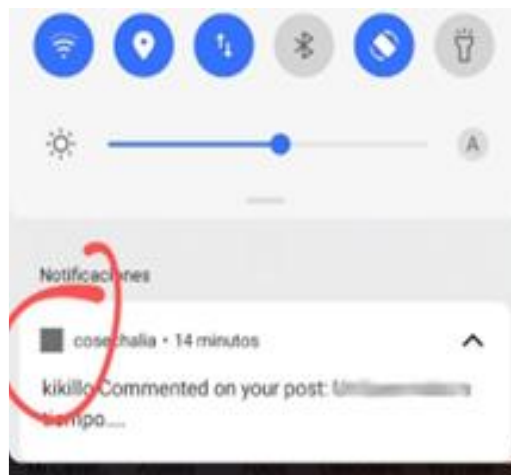


Figura 8.1: Issue- notificación sin imagen

8.1.3. Fallos de actualización

Fallo de actualización del perfil:

Si damos atrás al actualizar el perfil, la pantalla que muestra no tiene los datos de ese usuario actualizados.

Notificaciones no te llevan al detalle:

Algunas notificaciones no te llevan al chat en el que tienes un nuevo mensaje.

Actualización de comentarios en el Home:

Cuando publicamos una nueva publicación, se actualiza el Home, pero los comentarios no se actualizan correctamente. Ellos mantienen la posición que tenían.

Misma imagen de avatar en registro:

Al registrar dos usuarios en el mismo móvil a través de la app, cuando registramos el segundo se nos muestra el avatar del primero. Se ha comprobado que no se guardaría, pero está claro que el borrado local del usuario al cerrar la sesión no se hace correctamente.

8.2. Conclusión

La revisión por parte de estos testers, ha resultado ser fundamental porque aportan mejoras a todos los niveles, entre los resultados comentados solo se ha colocado el feedback más relevante, pero hay muchos más detalles sobre los mensajes en cada pantalla, inconsistencias en los colores entre pantallas y sugerencias que añaden valor al producto y que lo hacen más robusto. Como consecuencia de este trabajo, el backlog del proyecto ha crecido bastante y hasta que no estén resueltos los fallos comentados, no se dará por concluido el MVP.

Capitulo 9

9. Conclusiones

9.1. Conclusiones y mejoras futuras

Con la finalización de este Trabajo de Fin de Grado se da por terminada una primera etapa de lo que es una idea para llevar a mercado, los principales objetivos se han cumplido. Tenemos un MVP definido a falta de introducir las últimas mejoras extraídas de las pruebas de usabilidad. A través de este TFG se ha podido desarrollar un producto útil que a buen seguro tendrá su hueco en el sector.

La red social que se ha desarrollado permitirá a los agentes que actúan en el campo mejorar su comunicación utilizando las últimas tecnologías de manera que este software será un apoyo más para avanzar en la digitalización del campo.

En lo personal, se ha conseguido materializar una idea que se perseguía desde hace mucho tiempo, una de las cosas más satisfactorias es comprender como durante la realización de este proyecto he estado utilizando el conocimiento que se me ha enseñado durante el grado y que gracias a esto no solo se tiene la capacidad de adaptarnos a nuevas tecnologías como Flutter sino de comprender y ejecutar proyectos software.

9.2. Líneas para seguir en el futuro

El objetivo futuro es hacer evolucionar el producto introduciendo una serie de herramientas como Google Analytics esto junto con más entrevistas con usuarios, permitirá aumentar las funcionalidades para con un producto más robusto y ajustado a mercado.

Analizar la correlación entre el aumento de usuarios y el consumo de recursos en Firebase para elegir el momento adecuado e iniciar la migración a un servidor propio.

Y todo aquello que el producto vaya solicitando el proyecto a medida que escala y que no están relacionadas directamente con la ingeniería del software, como puede ser, el marketing, finanzas, operaciones, recursos humanos y ventas.

A. Manual de usuario

Para poder instalar la aplicación en cada plataforma debemos seguir ejecutar una serie de comandos

A.1. Android

Para generar el archivo ejecutable en Android (apk) se utiliza el comando:

flutter build apk --release pero esta necesita ser firmada.

El proceso para genera el ejecutable a instalar comenzara y después de unos minutos tendremos en un directorio del proyecto dedicado a los ejecutables, en este caso la ruta de acceso sería:

proyecto/build/app/outputs/flutter-apk/app-release.apk

A.2. iOS

Para generar el archivo ejecutable en iOS (ipa) se utiliza el comando:

flutter build ipa --release

El proceso para genera el ejecutable a instalar comenzara y después de unos minutos tendremos en un directorio del proyecto dedicado a los ejecutables, en este caso la ruta de acceso sería:

proyecto/ build/ios/archive/app-release.ipa

Bibliografía

- [1] Funcas. Eduardo Bandrés, Vanessa Azón La despoblación de la España interior. 2020
- [2] Datosmacro.com Castilla y León: Socio-Demografía Pirámide poblacional. Diciembre 2021
<https://datosmacro.expansion.com/demografia/estructura-poblacion/espana-comunidades-autonomas/castilla-leon>
- [3] Europe Direct Castilla y León Rural, despoblación, Hacia una nueva definición de la despoblación
<http://www.edcastillayleonrural.eu/hacia-una-nueva-definicion-de-despoblacion-en-la-union-europea/>
- [4] Gobierno de España, Ministerio para la transición ecológica y el reto demográfico, Plan de recuperación y 130 medidas frente al reto demográfico. 2021
- [5] Udemy, Fernando Herrera, Flutter: Tu guía completa de desarrollo para iOS y Android. 2022
<https://www.udemy.com/course/flutter-ios-android-fernando-herrera>
- [6] Flutter, Documentación oficial de Flutter. 2022. <https://flutter.dev/>
- [7] Google, Firebase, Documentación oficial de Firebase. 2022. <https://firebase.google.com/docs>
- [8] Back4App, Mati Presta, ¿Qué es Firebase? Todos los secretos desbloqueados. 2022.
https://blog.back4app.com/es/que-es-firebase/#Ventajas_de_Firebase
- [9] Firebase, Firebase Performance Monitoring. 2022 <https://firebase.google.com/docs/perf-mon>
- [10] 3androides, ¿Por qué usar Flutter?. 2022 <https://www.3androides.com/actualidad/136-por-que-usar-flutter#:~:text=Flutter%20se%20programa%20en%20DART%3A,-Quiz%C3%A1%20muchos%20pod%C3%A9is&text=El%20motivo%20no%20es%20otro,operativo%20de%20Google%20llamado%20Fuchsia.>
- [11] Adobe, AdobeXD, XD to Flutter Plugin. 2022 <https://github.com/AdobeXD/xd-to-flutter-plugin#installation--setup>
- [12] Adobe, Flutter Package: Adobe xd. 2022 https://pub.dev/packages/adobe_xd
- [13] Felgo, In-Depth Comparison: Best Framework to Create Mobile Apps. 2020
<https://blog.felgo.com/cross-platform-app-development/react-native-flutter-felgo-framework-comparison>
- [14] Hipertextual, Software, Fuchsia OS, el nuevo sistema operativo de Google, empieza a llegar a los primeros dispositivos.2021<https://hipertextual.com/2021/08/fuchsia-os-el-nuevo-sistema-operativo-de-google-empieza-a-llegar-a-los-primeros-dispositivos>
- [15] Google, Flutter, The official package repository for Dart and Flutter apps. 2022 <https://pub.dev/>
- [16] Donetonic, Funtionalidades, Scrum, ¿Qué es scrum?. 2021 https://donetonic.com/es/que-es-scrum/#Los_Artefactos_en_Scrum_Scrum_Artifacts

- [17] Atlassian, Max Rehkopf Agile Coach, Gestion ágil de proyectos. 2022
<https://www.atlassian.com/es/agile/project-management/epics-stories-themes>
- [18] Trello, Funciones, Plantillas. 2022 <https://trello.com/templates>
- [19] Stern School of Business de Nueva York, Aswath Damodaran, Margin by Sector (US). 2022.
https://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/margin.html
- [20] Google, Firebase, prices. 2022. <https://firebase.google.com/pricing?hl=es-419#blaze-calculator>
- [21] Miguel Ángel Laguna. Modelado de sistemas Software. 2016.
- [22] Félix Prieto y Esperanza Manso. Fundamentos de ingeniería de software. 2015.
- [23] Yania Crespo González-Carvajal, Diseño de software, 2016.
- [24] HenryCode, tutorials, 2020. <https://github.com/GiancarloCode/bloc-pattern-example>
- [25] Yashwant kumar, FlutterDevs, Design Patterns in Flutter- Part 1(MVC), April 17, 2020.
<https://medium.flutterdevs.com/design-patterns-in-flutter-part-1-c32a3ddb00e2>
- [26] JavaCodeGeeks, Ayusch Jain, Software Development, Flutter Provider Pattern Explained, 2020.
<https://www.javacodegeeks.com/2020/05/flutter-provider-pattern-explained.html>
- [27] Margarita Gonzalo Tasis y Carmen Hernández Diez, Interacción Persona Computadora, 2016.
<https://www.questionpro.com/blog/es/pruebas-con-usuarios-o-test-con-usuarios/>
- [28] ArsTECHNICA, Caseh Johnston, Android fragmentation: One developer 3,997 devices. 2012.
<https://arstechnica.com/gadgets/2012/05/android-fragmentation-one-developer-encounters-3997-devices/>