

# **GESTA - Gestión de activos**

**Rebeca de Pedro Medina**

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles

**Francesc D'Assis Giralt Queralt**

Junio 2020



Esta obra está sujeta a una licencia de Reconocimiento-NonCommercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>GESTA - Gestión de activos</i>
<b>Nombre del autor:</b>	<i>Rebeca de Pedro Medina</i>
<b>Nombre del consultor/a:</b>	<i>Francesc D'Assís Giralt Queralt</i>
<b>Fecha de entrega (mm/aaaa):</b>	06/2020
<b>Titulación:</b>	<i>Master universitario de Desarrollo de aplicaciones para dispositivos móviles</i>
<b>Área del Trabajo Final:</b>	<i>Trabajo final de máster DADM</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Activo, gestión y rendimiento.</i>

**Resumen del Trabajo:**

Desde la primera fase de análisis y planificación del trabajo, hasta la implementación de la aplicación, el desarrollo de este trabajo de final de master ha supuesto un reto constante.

Quería implementar una aplicación que diese solución a problemas reales y surgió la idea de crear GESTA, una aplicación para gestión de activos. La mayoría de empresas requieren gestionar sus activos, bien porque estos sean la principal fuente de ingresos en su modelo de negocio, o bien porque los utilicen en la realización de sus actividades. Por ello, GESTA ha sido principalmente diseñada para que usuarios de muchas tipologías de empresas puedan utilizarla como herramienta de trabajo o como herramienta de apoyo a su trabajo.

El principal cometido en la fase de diseño ha sido diseñar una aplicación fácil de utilizar, intuitiva y con una curva de aprendizaje muy rápida.

En la fase de implementación se ha trabajado buscando velocidad de respuesta, una aplicación robusta y que sea fácilmente escalable, ya que el proyecto GESTA es un proyecto que bien podría hacerse un hueco en el mercado actual.

Desarrollar una aplicación completa en tan poco tiempo me ha permitido observar cuan importante es un buen análisis y un buen diseño previo a realizar la implementación, ya que de otra forma, los tiempos, inevitablemente, se hubiesen alargado, y muy posiblemente, no hubiese podido finalizar el proyecto en tan poco tiempo.

**Abstract:**

From the first stage of work analysis and planning to the implementation of the application, the development of this final master project has been a constant challenge.

I wanted to implement an application that would solve real problems, and the idea of creating GESTA, an asset management application, appeared in my mind.

Most companies have to manage their assets, either because they are the main source of incomes in their business model, or because they use them to carry out their activities. For this reason, GESTA has been mainly designed to be used for many users of different kind of companies can use it as their main work tool or as a tool to support their work.

The main task in the design stage has been to design an application that is easy to use, is very intuitive and with a very fast learning curve.

In the implementation stage, hard work has been carried out seeking for high speed of response, trying to develop a very sturdiness and easily scalable application, in the aim that GESTA be a project that could well find its place in the actual apps market.

The fact of develop a competitive application in such a short time has allowed me to observe how important is to count with a good analysis and good design prior to carry out the implementation, since otherwise the times would inevitably have lengthened, and quite possibly, could not have ended the project in such a short time.

# Índice

1. Introducción .....	8
1.1 Contexto y justificación del Trabajo .....	8
1.2 Objetivos del Trabajo .....	10
1.3 Enfoque y método seguido .....	11
1.4 Planificación del Trabajo .....	13
1.5 Estudio de riesgos y medidas de contingencia .....	15
1.6 Breve sumario de productos obtenidos .....	15
2. Diseño y arquitectura .....	16
2.1 Usuarios y contextos de uso .....	16
2.1.1 Perfiles de usuarios .....	16
2.1.2 Fichas de persona .....	17
2.1.3 Contexto de uso .....	20
2.2. Diseño conceptual .....	21
2.2.1. Escenarios de uso .....	21
2.2.2 Necesidades funcionales y de diseño .....	23
2.3 Prototipado .....	25
2.3.1 Estructura de navegación .....	25
2.3.2 Diagrama de interacción .....	26
2.3.3 Prototipo de baja fidelidad .....	27
2.3.4 Diseño para la interacción .....	30
2.3.5 Prototipo de alta fidelidad .....	31
2.4 Evaluación .....	34
2.5 Casos de uso .....	35
2.6 Diseño de la arquitectura .....	39
2.6.1 Base de datos .....	39
2.6.2. Arquitectura de la aplicación .....	40
3. Desarrollo .....	42
3.1 Organización del proyecto .....	42
3.2 Guía de implementación .....	44
3.2 Tecnologías utilizadas .....	45
3.2.1. Firebase .....	45
3.2.1.1 Realtime Database .....	45
3.2.1.2 Storage .....	46
3.2.1.3 Firebase Authentication .....	46
3.2.2 Room .....	47
3.2.3 Play services .....	47
3.2.4 Lectura de etiquetas de activos .....	48
3.2.5 Material Design .....	48

3.2.6. Android Jetpack.....	48
3.3. Problemáticas detectadas durante la implantación y soluciones aplicadas	
49	
3.3.1 Claves primarias.....	49
3.3.2 Visualización de documentos .....	49
3.3.3. Autenticación .....	50
3.3.4 Grupos de trabajo .....	50
4. Plan de pruebas .....	52
5. Conclusiones .....	55
7. Bibliografía.....	57

## Lista de figuras

Ilustración 1. Logo de Fractal .....	9
Ilustración 2. Logo de IsbCant .....	9
Ilustración 3. Logo de Gestión de Activos IT .....	9
Ilustración 4. Logo de Sistema de Inventario Inteligente.....	9
Ilustración 5. Mobile Operating System Market Share Worldwide from Jan 019 - Jan 2020 .....	12
Ilustración 6. Mobile Operating System Market Share Spain from Jan 019 - Jan 2020 .....	12
Ilustración 7. Desarrollo en cascada .....	12
Ilustración 8. Diagrama de Gantt - Asignación de días y horas al proyecto por fases .....	14
Ilustración 9. Diagrama de Gantt - Asignación de días y horas a la fase del plan de trabajo .....	14
Ilustración 10. Diagrama de Gantt - Asignación de días y horas a la fase de diseño .....	14
Ilustración 11. Diagrama de Gantt -Asignación de días y horas a la fase de implementación .....	15
Ilustración 12. Diagrama de Gantt - Asignación de días y horas a la fase de documentación .....	15
Ilustración 13. Ficha de persona - Víctor.....	17
Ilustración 14. Ficha de persona - Marta .....	18
Ilustración 15. Ficha de persona - Elisa .....	19
Ilustración 16. Diagrama de navegación .....	25
Ilustración 17. Diagrama de interacción .....	26
Ilustración 18. Diagrama de casos de uso .....	35



Ilustración 19. Diagrama de base de datos - Entidad -Relación .....	39
Ilustración 20. Diagrama de la arquitectura de la aplicación .....	40
Ilustración 21. Organización del proyecto en paquetes.....	42
Ilustración 22. Comunicación entre capas .....	44

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

La gran mayoría de las empresas, sea cual sea su tipología, gestiona activos, bien sea para utilizarlos en el negocio o con fines de venta, como maquinaria o como bienes inmuebles. Todas estas empresas, bien sea un técnico, un responsable de área, comerciales y otros muchos perfiles de usuarios, requieren conocer las características de estos activos, su estado, su disponibilidad y frecuentemente su ubicación, por multitud de motivos diferentes como pueden ser realizar su mantenimiento preventivo, realizar reparaciones en ellos, ponerlos en uso, darlos de baja, consultar su estado para usos futuros, venderlos, etc. todo ello con el fin de extender su vida útil y maximizar su valor, con el consecuente ahorro de costes.

A nivel de individuo particular, se podría decir que existe una necesidad similar, resultando, generalmente, el volumen de activos a gestionar mucho menor, y no siendo tan imprescindible la gestión mencionada anteriormente aunque sí aplicable y recomendable.

Por todo lo comentado anteriormente, la necesidad de disponer de una herramienta que ayude a gestionar toda la información de activos, principalmente en el caso de empresas, es indispensable.

En base a este contexto, se ha realizado un estudio de mercado con el objetivo de analizar los productos actuales existentes que cubran las necesidades comentadas anteriormente: la gestión de activos. Se observa que, actualmente, y dejando a un lado las aplicaciones de control de stock de almacenes que están enfocadas al control de stock y no a la gestión de activos, sorprendentemente la oferta de aplicaciones disponible no es muy extensa y suele formar parte de otras aplicaciones cuyo objetivo principal no es la propia gestión de activos, sino el mantenimiento de activos (preventivo y correctivo) implicando, además, un coste de compra del producto elevado y un proceso de configuración y implantación costoso.

Para la realización de este estudio se ha accedido al mercado de aplicaciones más importante para dispositivos con SO Android, Google Play, donde figuran aplicaciones relacionadas con la gestión de activos como Fractal, IsbCant, Gestión de Activos IT o Sistema de Inventario Inteligente.

A continuación se detallan las características principales de estas aplicaciones:



Ilustración 1. Logo de Frattal

Frattal - Gestión de Activos y Mantenimiento, es una aplicación enfocada principalmente a la gestión del mantenimiento. Es una aplicación con una funcionalidad muy completa, pero tiene un coste elevado, tanto a nivel económico en su compra como de implantación. No se trata de una aplicación autónoma, ya que requiere de otras plataformas para su configuración y puesta en marcha. [1]

IsbCant - Gestión de Activos, es una aplicación que está limitada a un determinado conjunto de activos muy concreto como son portátiles, móviles, discos duros y libros.[2]



Ilustración 2. Logo de IsbCant



Ilustración 3. Logo de Gestión de Activos IT

Gestión de Activos IT es una aplicación, que a parte de tener unas valoraciones bastante pobres por parte de sus usuarios en cuanto a usabilidad, es una aplicación con una interfaz de aspecto bastante antiguo y con una funcionalidad limitada y enfocada a departamentos de tecnología y alquiler de activos informáticos (IT). Su uso requiere una suscripción y una extensión de Android SO llamada ginstr launcher. [3]

Sistema de Inventario Inteligente - Móvil & Web, es una aplicación que está enfocada al control de stock, pues se centra en el control de la cantidad de artículos disponibles de un mismo tipo. Permite el uso de etiquetas QR para la identificación de artículos, así como para su generación, aunque no permite la identificación de artículos mediante lectura NFC. A pesar de que aparezca una aplicación relacionada con la gestión de inventario, realmente no lo es. [4]



Ilustración 4. Logo de Sistema de Inventario Inteligente

Ampliando un poco más la búsqueda, ya fuera fuera del mercado de aplicaciones móviles, encontramos sistemas de gestión de activos y mantenimiento formados por un ecosistema de varias plataformas, pero que requieren de un dispositivo con pantalla grande para, mediante una aplicación web o de escritorio, configurar el sistema para su uso. Encontramos aplicaciones como por ejemplo Kyros y su gestión de recursos móviles y gestión de activos. Este tipo de software, a parte de requerir de una implantación costosa, tiene un coste elevado respaldado por el valor añadido de la funcionalidad compleja que ofrece el producto. [5]

## 1.2 Objetivos del Trabajo

A continuación se enumeran los objetivos funcionales y no funcionales del proyecto a realizar.

### Objetivos funcionales

- Implementar una aplicación para dispositivos móviles (smartphone y tabletas) que permita gestionar activos de forma sencilla y eficiente sin requerir de otras plataformas para su uso y configuración.
- La aplicación debe permitir registrar grupos de usuarios (por ejemplo empresas) que compartan información sobre activos en tiempo real.
- La aplicación debe permitir registrar usuarios y asociarlos a los grupos mencionados anteriormente.
- La aplicación debe permitir dar de alta nuevos activos, bien sea de forma manual o mediante lectura de un código (NFC, CB (Código de barras) o QR).
- La aplicación debe permitir consultar y editar la información de los activos existentes, bien sea de forma manual o mediante lectura de un código (NFC, CB (Código de barras) o QR) y clasificarlos.
- La aplicación debe permitir geolocalizar un activo y consultar la geolocalización de un activo.
- La aplicación debe realizar las funcionalidades sobre activos tenga el dispositivo o no, acceso a internet.

### Objetivos no funcionales

- La aplicación debe implementar un sistema de encriptación para los datos confidenciales de los usuarios tales como las contraseñas de usuarios.
- La aplicación requerirá de un servidor y una base de datos que permita guardar la información de grupos de usuarios, usuarios, activos, características de activos y compartir esta información entre usuarios pertenecientes al mismo grupo.

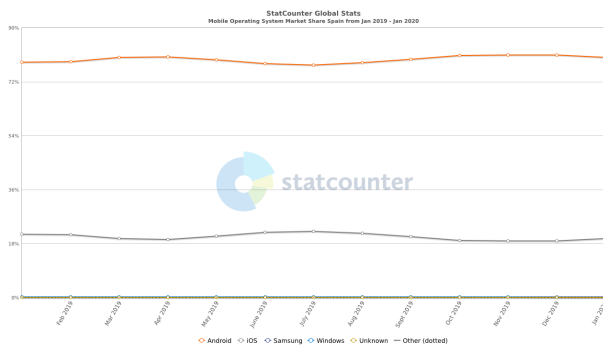
- La aplicación requerirá de una base de datos local para consultar los datos de activos cuando el dispositivo no tenga acceso a internet y no pueda acceder a la información disponible en el servidor.
- La aplicación debe implementar un sistema que permita sincronizar ambas bases de datos, siempre que el dispositivo tenga acceso a internet.
- La aplicación debe implementar un sistema que permita la lectura de etiquetas NFC, CB (Código de barras) y QR.
- La aplicación debe implementar un sistema que permita la geolocalización de activos.
- La aplicación debe implementar una interfaz de uso sencilla y rápida para poder ser usada por usuarios sin requerir un conocimiento experto en tecnología por parte de los mismos. Debe tener una curva de aprendizaje rápida.

### 1.3 Enfoque y método seguido

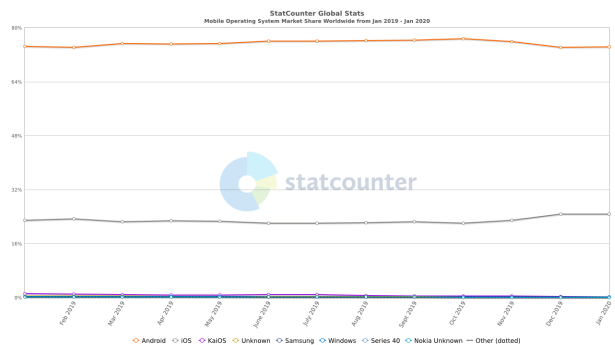
Las posibles opciones disponibles para cumplir con los objetivos planteados en el proyecto son o bien el desarrollo de una aplicación nativa o bien el desarrollo de una aplicación WEB híbrida que funcione en smartphone y tabletas, ya que se requiere poder trabajar tanto de forma conectada como de forma desconectada, es decir, con acceso a internet o sin acceso a internet, en dispositivos utilizados por usuarios que pueden estar en itineraria, y tener acceso a los sensores del dispositivo tales como el lector NFC o el sensor GPS.

Se determina que se desarrollará una aplicación móvil nativa, ya que se busca el mejor rendimiento y la mejor experiencia de usuario a la hora de utilizar la aplicación, con el fin de captar usuarios frente a la competencia existente para este producto concreto. También se busca un uso eficiente del hardware del dispositivo, como serían los sensores de geolocalización (GPS y redes móviles), la cámara para las lecturas de QR y códigos de barras, el lector NFC, etc. con el fin de garantizar la accesibilidad a la información por parte del usuario de forma inmediata.

Al no disponer de tiempo ni recursos suficientes para implementar la aplicación para diferentes sistemas operativos (SO), se pretende desarrollar la aplicación únicamente para dispositivos con SO Android (Smartphone y tabletas), ya que actualmente (Enero del 2020), Android es el SO móvil más utilizado a nivel mundial con casi un 75% de cuota de mercado, a nivel europeo con un 72,58% y a nivel estatal (España) con un 80,11%. [6]



Ilustraciones 5. Mobile Operating System Market Share Worldwide from Jan 019 - Jan 2020



Ilustraciones 6. Mobile Operating System Market Share Spain from Jan 019 - Jan 2020

A parte del gran uso generalizado de dispositivos con SO Android, el coste de los mismos es mucho más reducido que el de su competencia directa, los productos de la compañía Apple, por lo que en este caso se incrementan aún más el porcentaje de uso indicado anteriormente en los usuarios potenciales del producto a desarrollar que normalmente no invierten grandes sumas de dinero en sus dispositivos móviles.

El lenguaje para desarrollar la aplicación será Java, debido al conocimiento previo de este lenguaje por parte de la autora del trabajo final de master, además de ser uno de los lenguajes oficiales para desarrollar aplicaciones para Android.

El producto, debido a que los objetivos están bien definidos desde el inicio del proyecto, al contar con un tiempo de implementación muy limitado con poco margen a cambios o ampliaciones en la funcionalidad a implementar, se desarrollará en fases secuenciales siguiendo el estilo conocido como desarrollo en cascada (waterfall), siguiendo el plan de trabajo definido en el siguiente capítulo. [7]

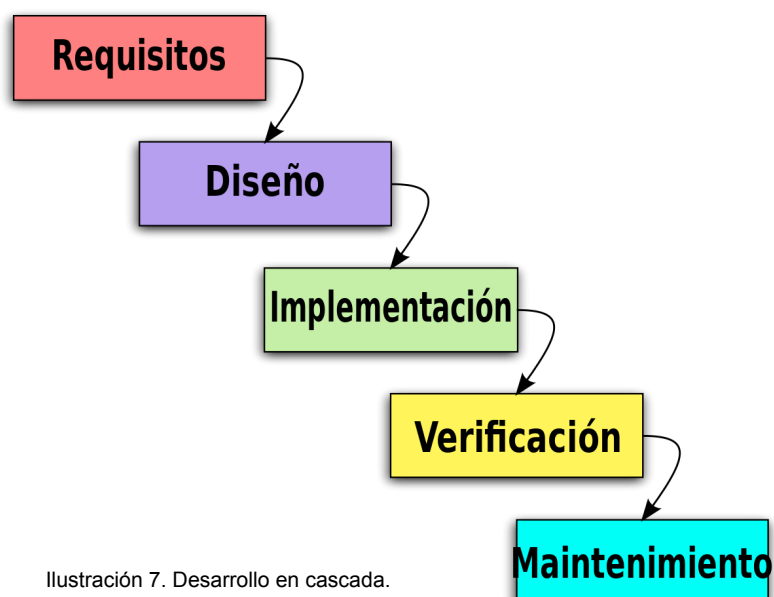


Ilustración 7. Desarrollo en cascada.

## 1.4 Planificación del Trabajo

Como recursos para el desarrollo e implementación se dispone de un portátil con sistema operativo Windows 10 profesional y Android Studio instalado para implementar la aplicación para Android en código nativo con Java. Se dispone de un smartphone Samsung A40 con SO Android 9.0 y un smartphone Samsung A5 con SO Android 8.0 para la realización de pruebas con dispositivos reales y múltiples emuladores generados desde el propio entorno de desarrollo Android Studio.

Como mecanismo para la persistencia de datos, debiendo ser accesible por diferentes dispositivos que tengan que compartir información entre ellos por pertenecer a un mismo grupo, se utilizará la base de datos que proporciona Firebase. La base de datos no relacional que almacena datos en formato JSON de Firebase se considera la mejor opción en este caso, ya que no se dispone de tiempo para implementar un servicio web que permita el acceso a datos almacenados en otra base de datos, ni se dispone de un servidor propio para instalar una base de datos de otro proveedor.

Los datos de activos deben sincronizarse y almacenarse a su vez en los diferentes dispositivos móviles para que puedan estar disponibles si el dispositivo no tiene acceso a internet, por lo que se hará uso de una base de datos local SQLite y la librería Room. Room es una librería que permite la abstracción sobre SQLite conservando la potencia y el rendimiento. Actualmente es la librería recomendada por Google para trabajar con SQLite.

El proyecto se desarrollará en fases secuenciales, como ya se ha comentado anteriormente, a las que se ha asignado un tiempo en base a la complejidad del desarrollo y a las fecha de entrega previstas de cada fase.

A continuación se muestran unas imágenes de los tiempos y fases de desarrollo correspondientes a cada fase en forma de diagramas de Gantt, donde se puede observar la asignación de tiempo en días y horas.

La numeración que figura en las casillas diarias por cada trabajo a realizar de cada fase representa el número de horas asignadas ese día a esa tarea. Como se puede observar, el tiempo dedicado a los días correspondientes a fines de semana es superior al resto de días, como también lo es el dedicado los viernes respecto a otros días entre semana. Esto se debe a la jornada laboral de la autora del trabajo final de master que marca estas diferencias.

	Inicio	Final	Días	Horas
<b>Plan de trabajo</b>	19/2/20	11/3/20	21	39
Idea y concepto			4	5
Estudio de mercado			6	10
Análisis y definición de objetivos			2	8
Plan de trabajo - diagrama de Gantt			6	8
Documentación			4	8
<b>Diseño</b>	12/3/20	1/4/20	21	48
Usuarios y contexto de uso (Fichas de persona & escenarios) - Documentación			10	24
Arquitectura y wireframes (Diagramas)			11	24
<b>Implementación</b>	2/4/20	13/5/20	42	120
Diseño de BDD			3	6
Implementación BDD (Firebase & Sql Lite con Room)			4	13
Integración de datos Firebase - App (Room)			4	10
Desarrollo de actividades			15	42
Implementación lectura y funcionalidad NFC, CB (Código de barras) y QR			6	14
Implementación de la funcionalidad relacionada con la geolocalización			2	12
Pruebas			5	8
Documentación			5	15
<b>Documentación</b>	14/5/20	5/6/20	23	51
Manual de la aplicación			3	8
Memoria del proyecto			13	27
Video			7	16

Ilustración 8. Diagrama de Gantt - Asignación de días y horas al proyecto por fases

	Días	Horas	FEBRERO 2020																							
			19	20	21	22	23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10	11		
<b>Plan de trabajo</b>	21	39																								
Idea y concepto	4	5	1	1	1	2																				
Estudio de mercado	6	10				5	1	1	1	1	1															
Análisis y definición de objetivos	2	8											3	5												
Plan de trabajo - diagrama de Gantt	6	8												1	1	1	1	1	3							
Documentación	4	8																		5	1	1	1			

Ilustración 9. Diagrama de Gantt - Asignación de días y horas a la fase del plan de trabajo

	Días	Horas	MARZO 2020																							
			12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1			
<b>Diseño</b>	21	48																								
Usuarios y contexto de uso (Fichas de persona & escenarios) - Documentación	10	24	1	2	5	5	1	1	1	1	2	5														
Arquitectura y wireframes (Diagramas)	11	24												5	1	1	1	1	2	5	5	1	1	1		

Ilustración 10. Diagrama de Gantt - Asignación de días y horas a la fase de diseño



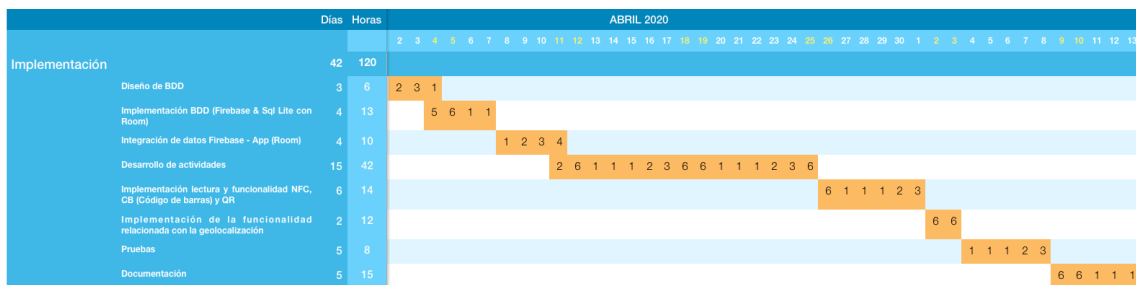


Ilustración 11. Diagrama de Gantt -Asignación de días y horas a la fase de implementación

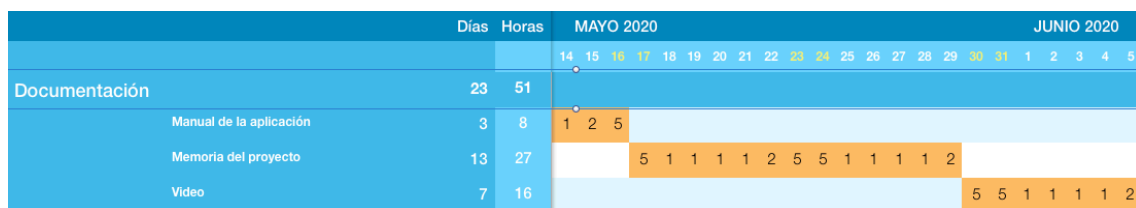


Ilustración 12. Diagrama de Gantt - Asignación de días y horas a la fase de documentación

## 1.5 Estudio de riesgos y medidas de contingencia

Durante el desarrollo del proyecto se pueden presentar problemas, como por ejemplo un análisis incorrecto de los tiempos requeridos por cada tarea, que genere que no se pueda seguir el plan de trabajo presentado. En este caso, está previsto que se puedan realizar ajustes asignando mayor tiempo a ciertas tareas y restándoselo a otras. En caso de requerir el proyecto de un mayor número de horas dedicadas general, se podrían obtener, principalmente, de fines de semana y de viernes, aumentando estos tiempos incluso en un 15%.

## 1.6 Breve resumen de productos obtenidos

1. Producto: fichero apk correspondiente a la aplicación desarrollada para dispositivos Android y manual de usuario de la aplicación.
2. Memoria del proyecto.
3. Presentación en vídeo.

## 2. Diseño y arquitectura

### 2.1 Usuarios y contextos de uso

En el desarrollo de ese proyecto se ha utilizado el método DCU, es decir, diseño centrado en el usuario, siguiendo el esquema de análisis, diseño y evaluación.

#### 2.1.1 Perfiles de usuarios

La primera etapa del método DCU en la fase de diseño del proyecto, consiste en identificar los perfiles de usuarios potenciales de la aplicación. Para el producto a desarrollar se han podido diferenciar tres perfiles diferentes en base al uso que el usuario puede darle a la aplicación a desarrollar:

- Usuarios que utilizan la aplicación como herramienta de apoyo a su trabajo, es decir, usuarios que necesitan acceder a la información que les proporciona el sistema facilitando su trabajo. Por ejemplo, el usuario accede al inventario de activos, consulta su ubicación, sus características, con el fin de realizar algún tipo de intervención sobre ellos, una reparación, una instalación, etc.
- Usuarios que utilizan la aplicación como parte fundamental de su negocio, siendo esta la herramienta de gestión del mismo, como por ejemplo empresas que realizan instalaciones de aparatos de cualquier tipo, proporcionan un servicio de renting de activos, venta, etc, y utilizan la aplicación para controlar y gestionar el inventario así como su estado.
- Usuarios que utilizan la herramienta como parte de un sistema de administración para tener control del inventario de los activos de una empresa, el valor de su inmovilizado, gestionar compras, etc.

## 2.1.2 Fichas de persona

Con el objetivo de mostrar en forma de arquetipo los perfiles de usuarios de la aplicación definidos en el punto anterior, se han generado 3 fichas de personas, cada una de ellas representativa de cada uno de estos perfiles.

- Primer perfil: Víctor, usuario que requiere la aplicación como apoyo a su trabajo de informático instalador y de mantenimiento [8].

### Víctor



*"Intento ser bueno en lo que hago, y sé que puedo hacerlo mejor con ayuda de las herramientas adecuadas."*

**Edad:** 38

**Estudios:** Ingeniero informático de sistemas

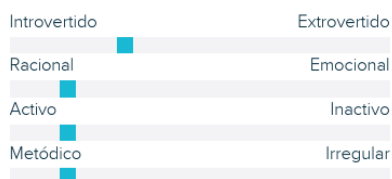
**Familia:** Casado sin hijos

**Residencia:** Barcelona, España

**Personalidad:** Metódico

**Conocimiento tecnológico:** Experto

### Personalidad



### Objetivos

- Viajar a Japón y recorrer el país.
- Tener hijos.
- Recibir un aumento de sueldo.

### Necesidades

- Mejorar su sueldo y ahorrar dinero.
- Un sistema de gestión que le ayude a localizar y gestionar los problemas a resolver en su trabajo para mejorar los tiempos y su rendimiento.
- Tener más tiempo libre para disfrutarlo con su mujer.

### Biografía

Víctor, natural de Barcelona, ciudad donde vive y trabaja, lleva 3 años trabajando en una empresa multinacional de transporte. Él, junto con otro compañero, se encargan del mantenimiento de toda la infraestructura informática de la sede que esta empresa tiene en Barcelona. Su horario es de lunes a viernes de 8:00 a 18:00, aunque tiene que permanecer en la ciudad de guardia fines de semana alternos por si surge una emergencia. Los fines de semana que no tiene guardia aprovecha para escaparse con su mujer al pueblo donde viven los padres de esta, en la parte alta del Pirineo catalán, y hacer senderismo, práctica que ambos suelen realizar juntos y que disfrutan enormemente.

Últimamente, Víctor y su mujer hablan de ampliar la familia, pues este año ella ha cumplido los 31 y no quiere esperar a ser muy mayor para ser madre, pero ambos están de acuerdo en que antes de hacerlo, tienen que viajar a Japón y recorrer el país, uno de los sueños de sus vidas, especialmente para Víctor. Este es uno de los motivos por el que Víctor busca un aumento de sueldo desde hace ya varios meses, lo que les ayudaría a ahorrar, pero es consciente que necesita mejorar el rendimiento en su trabajo, solventar los problemas más rápido, para impresionar a su jefe, a la par que tener el valor suficiente para ir a solicitarle el aumento deseado.

Ilustración 13. Ficha de persona - Víctor

- Segundo perfil, Marta, usuaria que requiere la aplicación como parte fundamental de su negocio para gestionar su actividad [9].

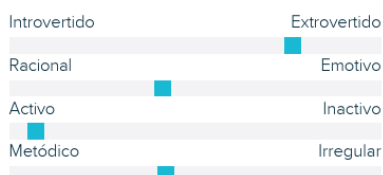
## Marta



*"Para que mi negocio funcione, debo exprimir todos los recursos al máximo"*

**Edad:** 41  
**Estudios:** Administración y dirección de empresas  
**Familia:** Separada  
**Residencia:** Esplugues del Llobregat, Barcelona  
**Personalidad:** Emprendedora  
**Conocimiento tecnológico:** Medio

### Personalidad



### Objetivos

- Que su empresa prospere y crezca
- Ser la única directora de su negocio
- Viajar más

### Necesidades

- Mejorar el rendimiento de la empresa en general, y especialmente la parte que ella dirige para hacerla más rentable
- Una aplicación fácil de utilizar que le permita tener control en todo momento de los activos de los que dispone y el estado de los mismos
- Aumentar sus ingresos

### Biografía

Marta, nacida en Valencia, siempre supo qué quería montar y dirigir su propio negocio. Cuando conoció a su actual ex-pareja, Jaime, decidieron invertir y montar una empresa de renting y venta de maquinaria especializada para el sector de la agricultura juntos. A pesar de que han pasado dos años desde que se separaron, han sabido mantener una buena relación, tanto en lo personal como en los negocios, lo que ha ayudado al crecimiento continuo de la empresa. Tras la separación, Jaime volvió a Valencia y desde entonces dirige la empresa en esa zona. Marta, por su parte, se encarga de la dirección y gestión del negocio en Cataluña. Marta tiene a dos personas trabajando a su cargo que se ocupan principalmente del transporte y el mantenimiento de las máquinas de renting mientras ella se encarga de la parte de ventas.

El sueño de Marta es poder ampliar el negocio y abrir otras sedes en otras partes del país, así como poder comprarle su parte del negocio a su marido y dirigirlo todo ella misma. Marta no ha tenido hijos, pero no se arrepiente, ya que es una persona enormemente extrovertida lo que hace que siempre tenga amigos con los que quedar para comer, cenar o tomar una copa, a parte de permitirle conseguir contactos beneficiosos para el negocio con mucha facilidad. Lo que le falta a Marta es tiempo libre. El poco que tiene lo emplea en hacer deporte, que hace que mantenga una buena forma física a parte de ayudarla a liberar las tensiones que genera al dirigir un negocio propio, y también a salir con amigos. Sin embargo, Marta anhela poder realizar un viaje a algún país extranjero y desconectar varios días de responsabilidades.

Ilustración 14. Ficha de persona - Marta

- Tercer perfil, Elisa, usuaria que utilizaría la aplicación como herramienta para administrar, por ejemplo, los activos de su empresa [10].

## Elisa



*"Necesito pasar más tiempo en casa, pero también quiero seguir desarrollando mi carrera profesional. Necesito agilizar las tareas que realizo de alguna forma."*

**Edad:** 36

**Estudios:** Administración y dirección de empresas

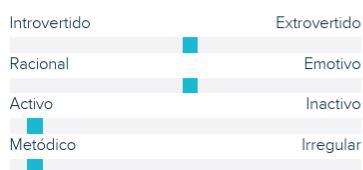
**Familia:** Casada con 2 hijos

**Residencia:** Hospitalet de Llobregat, Barcelona

**Personalidad:** Organizada

**Conocimiento tecnológico:** Medio

### Personalidad



### Objetivos

- Reducir su jornada laboral para poder pasar más tiempo con su familia
- Simplificar sus tareas para desarrollarlas en el menor tiempo posible, tanto a nivel laboral como personal
- Conseguir que sus hijos completen sus estudios

### Necesidades

- Que la empresa le facilite una reducción de jornada
- Poder realizar el trabajo asignado de forma más rápida para que en el transcurso de una posible jornada reducida pueda seguir realizando el grueso de las tareas
- Herramientas de gestión que faciliten el desarrollo de esas tareas

### Biografía

Elisa se casó muy joven y tuvo gemelos. Pese al aumento de actividad en su vida personal ocasionado por el cuidado de dos niños pequeños, consiguió terminar de formarse como administrativa y conseguir un puesto en una empresa que opera a nivel nacional dedicada al sector inmobiliario y de construcción como administrativa y gestora de compras. Es una persona extremadamente organizada con un gran sentido de responsabilidad, tanto a nivel personal con su familia como a nivel profesional con su trabajo. Es por esto que en la empresa en la que trabaja está muy bien valorada y Elisa lo sabe. Ella sabe que puede conseguir una reducción de jornada sin mayor problema, y siente que la necesita, ya que considera que debe pasar más tiempo en casa con los gemelos, pues cree que están en "una edad difícil". Sin embargo, para Elisa, su trabajo es muy importante y quiere prosperar. A pesar de no llevar muchos años en la empresa, debido a sus cualidades, ha ido adquiriendo responsabilidades que no quiere perder, y por tanto necesita poder continuar desarrollando sus tareas asignadas a pesar de la posible reducción de jornada. Sabe que es complicado abarcarlo todo, por lo que la única solución que encuentra es mejorar el rendimiento en todo lo que hace, ser mejor en casa y ser mejor en el trabajo. Elisa está convencida que puede conseguirlo.

Ilustración 15. Ficha de persona - Elisa

### 2.1.3 Contexto de uso

Una vez identificados los perfiles de usuario, y generadas las personas que los representan, se analiza el contexto de uso de la aplicación. Siguiendo los patrones descritos por Wolfram Nagel se procede a definir el contexto de uso del producto a desarrollar en base a tres parámetros básicos:

- Entorno: El producto a desarrollar, la aplicación móvil GESTA, debe poder ser utilizada en un variado número de entornos diferentes. Desde su uso en un espacio público, como puede ser una calle de una ciudad o de una población, por ejemplo en el caso de Marta, que requiere desplazarse de un lado a otro y poder consultar la información en cualquier lugar, hasta un espacio privado como puede ser una oficina o un edificio, como en el caso de Víctor y Elisa, que utilizan la aplicación en el espacio de su empresa, cualquier usuario debe poder, desde su dispositivo móvil tener acceso a cualquier funcionalidad que ofrece la aplicación en cualquiera de estos entornos.
- Situación: El usuario debe poder utilizar la aplicación desde un dispositivo móvil con SO Android, pudiendo tratarse de un smartphone o una tableta. Esto debe permitir al usuario poder utilizar la aplicación al tiempo que se desplaza caminando, como en el caso de Marta, o bien estando de pie, como en el caso de Víctor, o bien estando cómodamente sentado, como sería el caso de Elisa. Debido a esta variedad de situaciones, el usuario puede encontrarse en lugares con mejor o peor luminosidad, con mayor o menor comodidad para manejar el dispositivo, susceptible a un nivel mayor o menor de interrupciones, con acceso o no a la red móvil, lo que implica que la aplicación debe obedecer a un uso sencillo, con una navegabilidad simple, con componentes bien diferenciados y bien localizados, mostrando la información bien visible, contrastada, para que al usuario le resulte sencillo, intuitivo y rápido realizar cualquier operación.
- Modo de uso: GESTA es una herramienta de trabajo, y por ello su uso obedecerá, principalmente al llamado modo “lean forward”, en el que el usuario está concentrado y activo mientras utiliza la aplicación. Sin embargo, no siempre puede ser así, ya que en el caso de que la aplicación se utilice como herramienta de apoyo al desarrollo de otro trabajo, su uso puede verse interrumpido constantemente, acercándose más al concepto de “lean back”, ya que la aplicación se utilizaría de forma esporádica y cuya información se consumiría de forma pasiva.

## 2.2. Diseño conceptual

Una vez finalizada la etapa de investigación en la que se han definido los perfiles de usuarios potenciales de la aplicación, se han generado las fichas de persona a partir de los mismos y se ha analizado el contexto de uso, se han generado varios escenarios, cada uno de los cuales corresponde a un perfil de usuario, con el objetivo de sintetizar los datos obtenidos en la fase de análisis y describir como la aplicación va a ser utilizada por el usuario.

### 2.2.1. Escenarios de uso

- Primer escenario. Víctor.

Son las 8:15 de la mañana de un martes. El jefe de Víctor los ha reunido a él y a su compañero para informarles que se va a iniciar la sustitución del equipo informático de toda un área del primer piso del edificio. El total de equipos a sustituir suma un total de 15 CPUs con sus periféricos, es decir, pantallas, teclados, ratones, etc. Este proceso debe hacerse de forma muy rápida, ya que el personal del área debe estar sin trabajar cuando menos tiempo mejor, por lo que la tarea requiere de una buena organización. Víctor ve en la tarea una oportunidad para demostrar que puede hacer un buen trabajo. Víctor analiza el trabajo a realizar: cada equipo a sustituir debe ser dado de baja, mientras que para cada equipo nuevo, componentes incluidos, se requiere de la instalación correspondiente del hardware y del software, y finalmente debe inventariarse. El jefe de Víctor les informa que, al mismo tiempo que se realiza este trabajo se debe seguir cubriendo el mantenimiento del resto del edificio y dar servicio a los usuarios.

Víctor enseguida toma el liderazgo de la tarea de instalación, mientras que su compañero se encargará del trabajo de mantenimiento y soporte a usuarios. Víctor calcula el tiempo que puede llevarle la tarea. Gracias a la nueva aplicación, GESTA, inventariar los equipos, es poco más que pasar el móvil por la etiqueta de código de barras y seleccionar 4 características. Dar de baja los equipos sustituidos aún es más sencillo, ya que como están inventariados, buscarlos en la aplicación es pasar el móvil sobre su etiqueta de código de barras. En este aspecto, GESTA ahorrará a Víctor la pesada tarea de tener que rellenar pesadas fichas o cumplimentar excels en un ordenador. El trabajo pesado consistirá en la instalación del software de los equipos, proceso que no puede acelerar. Pero Víctor está decidido a demostrar con este trabajo que merece un aumento de sueldo. Sabe que su jefe estará controlando, mediante GESTA, la progresión del trabajo de Víctor en las instalaciones de los equipos. Víctor está convencido que mejorará el rendimiento esperado en este trabajo lo que le terminará de abrir las puertas al aumento de sueldo deseado.

- Segundo escenario. Marta

Marta se ha levantado a las 6:15 de la mañana. Ha salido a correr 45 minutos, se ha duchado, ha desayunado y a las 8:05 está en la furgoneta

camino a un pueblecito del Penedés. El chico que tenía que llevar una máquina recolectora de uva a unas bodegas de la zona se ha puesto enfermo y el otro compañero no puede cubrirle, pues tiene sus propias tareas que realizar para otros clientes. En esta época del año hay bastante demanda en el sector de la viticultura. Marta se dirige al almacén a recoger la máquina. Antes de salir, ha comprobado mediante GESTA que la máquina a alquilar está en buen estado y se encuentra en el almacén al que se dirige. Una vez la cargue en el almacén, se dirigirá a las instalaciones del cliente, depositará la máquina donde le indiquen, hará el trámite del renting con el encargado, cambiará el estado del activo conforme deja de estar disponible y geolocalizará la máquina para registrar su nueva ubicación. Este proceso, que consiste en buscar el equipo leyendo tan solo su etiqueta NFC con el móvil, seleccionar el nuevo estado, y pulsar un botón para geolocalizar la máquina (información muy útil para cuando tengan que volver a recogerla o bien volver por si la máquina requiere de alguna reparación), fue lo que terminó de convencerla para comprar GESTA. Las máquinas de renting de las que dispone, están distribuidas por muchos lugares de Cataluña, lugares que a veces no son fácilmente localizables.

Una vez finalizada la entrega y el trámite del renting, Marta vuelve a poner rumbo a Barcelona, donde calcula que llegará sobre las 14:00, la hora de comer. Mientras conduce, Marta piensa que quizás deba invertir en nuevas máquinas dedicadas a la recolección de uva, pues en ese momento hay una fuerte demanda, pero no está segura. Cuando aparque la furgoneta, de camino al restaurante donde ha quedado para comer con un par de amigas, comprobará las máquinas disponibles de ese tipo, cuantas están alquiladas y cuantas quedan disponibles, así como el historial de las mismas para comprobar el tiempo que suelen permanecer alquiladas. Con esta información y la del presupuesto disponible anual para compras, evaluará si vale la pena invertir en ello o quizás sea más rentable adquirir otro tipo de maquinaria.

- Tercer escenario. Elisa

Son las 16:45. En 15 minutos Elisa termina su jornada laboral. Encima de su mesa aún tiene unos cuantos pedidos por realizar. En ese momento ve como entra en su correo una solicitud de incorporación de un nuevo delineante. Para el día siguiente. Siente crecer una sensación de enfado. Ya estamos otra vez, piensa, a tan solo 15 minutos para su hora de salida le piden que gestione una petición para el día siguiente cuando es evidente que la información existe desde hace días. Intenta calmarse, los pedidos tendrán que esperar. Revisa el listado de material que requiere el nuevo delineante. Coge su móvil y accede a GESTA. Busca cada uno de los objetos del listado y revisa si hay alguno disponible en el almacén. Genera la orden con el número identificativo de cada producto y se la envía al responsable de área, David. David aún se enfadará mas que ella ante la urgencia, pues esto va a implicar que tendrá que pasar por el almacén a recoger el material, realizar la instalación y dejarlo todo preparado esa misma tarde, por lo que va a tener que hacer horas extras. Al menos la petición le llegará de inmediato al móvil, y, como también utiliza GESTA, puede ir directamente al almacén y localizar rápidamente el material con los códigos que le ha enviado Elisa.

Por suerte para Elisa, la tarea no le ha llevado ni 5 minutos, así que le va a dar tiempo a hacer al menos uno de los pedidos pendientes. Aprovecha para



añadir al pedido ciertos productos que, al buscar los disponibles para el nuevo delineante, se ha dado cuenta que no va a quedar apenas stock disponible de los mismos.

## 2.2.2 Necesidades funcionales y de diseño

Antes de iniciar la fase de creación de prototipos, donde se pasa a dar forma a la aplicación que se va a desarrollar, se reiteran las necesidades funcionales y de diseño de la aplicación, que se pasan a resumir a continuación:

- La aplicación debe permitir gestionar activos, crearlos, editarlos y visualizar su información.
- Estos activos, y toda la información relacionada, solo debe ser visible para miembros del mismo grupo de trabajo, que principalmente representará una empresa. La aplicación debe permitir el acceso seguro a la información por parte de estos usuarios, por lo que debe ofrecer un método de autenticación seguro.
- La aplicación no debe requerir de otras plataformas, debe ser autónoma, por lo que debe permitir crear usuarios y grupos de trabajo, así como configurar toda la información que luego vaya a ser utilizada en la creación, configuración y visualización de los activos.
- La aplicación debe permitir crear activos de forma rápida, haciendo uso de información ya existente cómo puede ser una etiqueta NFC, un código de barras, un código QR u otros datos.
- La aplicación debe permitir acceder a los activos y a su información relacionada de forma rápida, haciendo uso de cualquier dato del que disponga el usuario: una etiqueta NFC, un código de barras, un código QR, una descripción, la tipología del activo, etc. por lo que tiene que permitir la búsqueda por cualquiera de los datos del activo.
- La aplicación debe permitir consultar el histórico de un activo en el que debe figurar información tal como sus cambios de estado o sus cambios de ubicación, de forma que el usuario pueda observar rápidamente el ciclo de vida de ese activo.
- La aplicación debe permitir detectar y acceder a la información de geolocalización de los activos, para establecer dónde se encuentra un activo determinado en un momento determinado y poder acceder a dicha localización en todo momento.

- La aplicación debe permitir mantener la información de activos disponible y actualizada, en la medida de lo posible, en los terminales de los usuarios, de forma que la aplicación siga prestando el mismo servicio aún en caso de no disponer de acceso a internet.
- Así mismo, la aplicación debe permitir actualizar la información local generada en un sistema de persistencia de datos accesible para el resto de dispositivos de los usuarios del mismo grupo, con el fin de compartir la información entre ellos.
- La aplicación debe presentar una interfaz de uso agradable y que permita visualizar la información de forma clara, limpia y con un contraste adecuado, así como una identificación correcta de los tipos de elementos a utilizar, si son elementos pulsables o solo muestran información, todo ello con el fin de hacerla intuitiva y facilitar su uso, aun en condiciones no adecuadas, por ejemplo, de luminosidad.
- El tamaño y disposición de los elementos debe ser adecuado. Se pretende conseguir una buena relación entre el tamaño de la pantalla y la información a mostrar en ella. Los elementos pulsables deben tener un tamaño adecuado, pensando en aquellos usuarios con manos grandes para que no tengan problemas de uso. También la disposición de los elementos debe ayudar al uso del dispositivo a una mano en los casos en los que el dispositivo lo permita.
- La aplicación debe tener una estructura de navegación sencilla y mantenerla en toda la aplicación, para mejorar su usabilidad y conseguir una curva de aprendizaje rápida por parte de todos los usuarios, sean expertos o no.

## 2.3 Prototipado

### 2.3.1 Estructura de navegación

Una vez definidas las necesidades funcionales y de diseño, se pasa a definir la estructura de navegación principal de la aplicación, intentando que sea lo más sencilla posible y que permita al usuario saber donde está y de donde viene en todo momento:

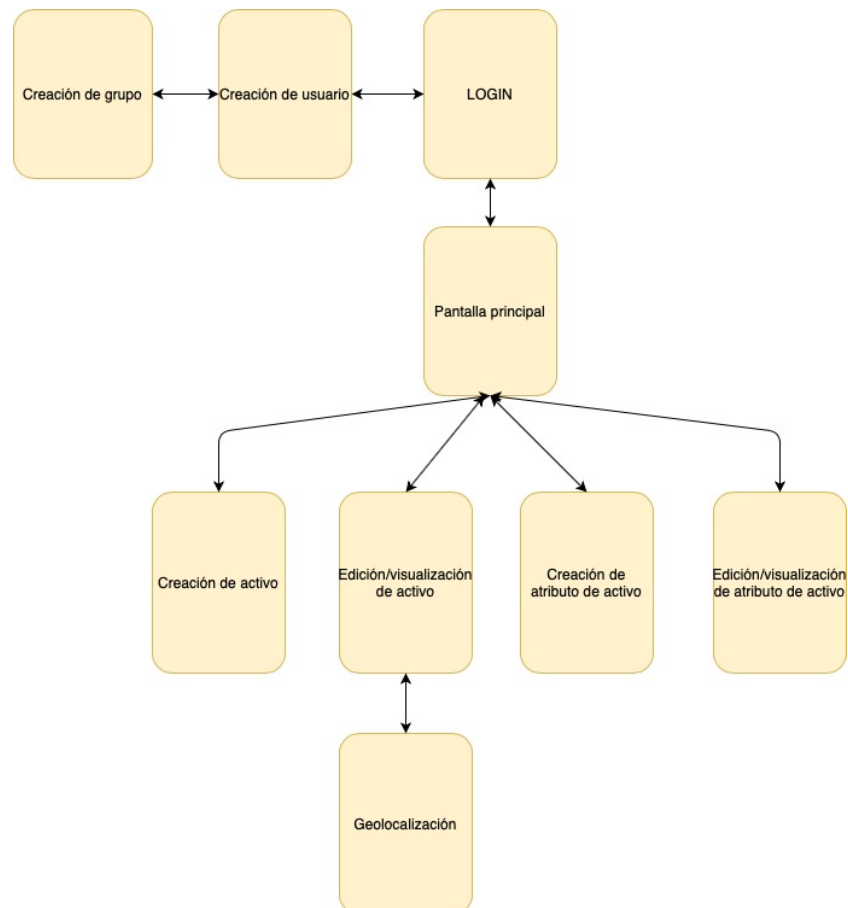


Ilustración 16. Diagrama de navegación

Cómo se puede observar en el diagrama, la estructura de navegación principal de la aplicación obedece a una estructura jerárquica con muy pocos niveles de profundidad. El usuario parte de una pantalla de login que permite, tanto el acceso a la pantalla principal de la aplicación, como a la configuración de nuevos usuarios y grupos. La pantalla principal de la aplicación contiene el listado de activos y permite acceder a la información de los mismos, a crear nuevos activos o a configurar las estructuras de atributos de activos (estados, tipologías o ubicaciones). Es un tipo de estructura que permite al usuario orientarse fácilmente.

### 2.3.2 Diagrama de interacción

Con el fin de profundizar más en los procesos de interacción del usuario con la aplicación y como paso previo a la creación del prototipo de bajo nivel, se desarrolla el diagrama de interacción, basado en la estructura de navegación planteada en el punto anterior y los requisitos funcionales y de diseño de la aplicación del punto 2.2.2.

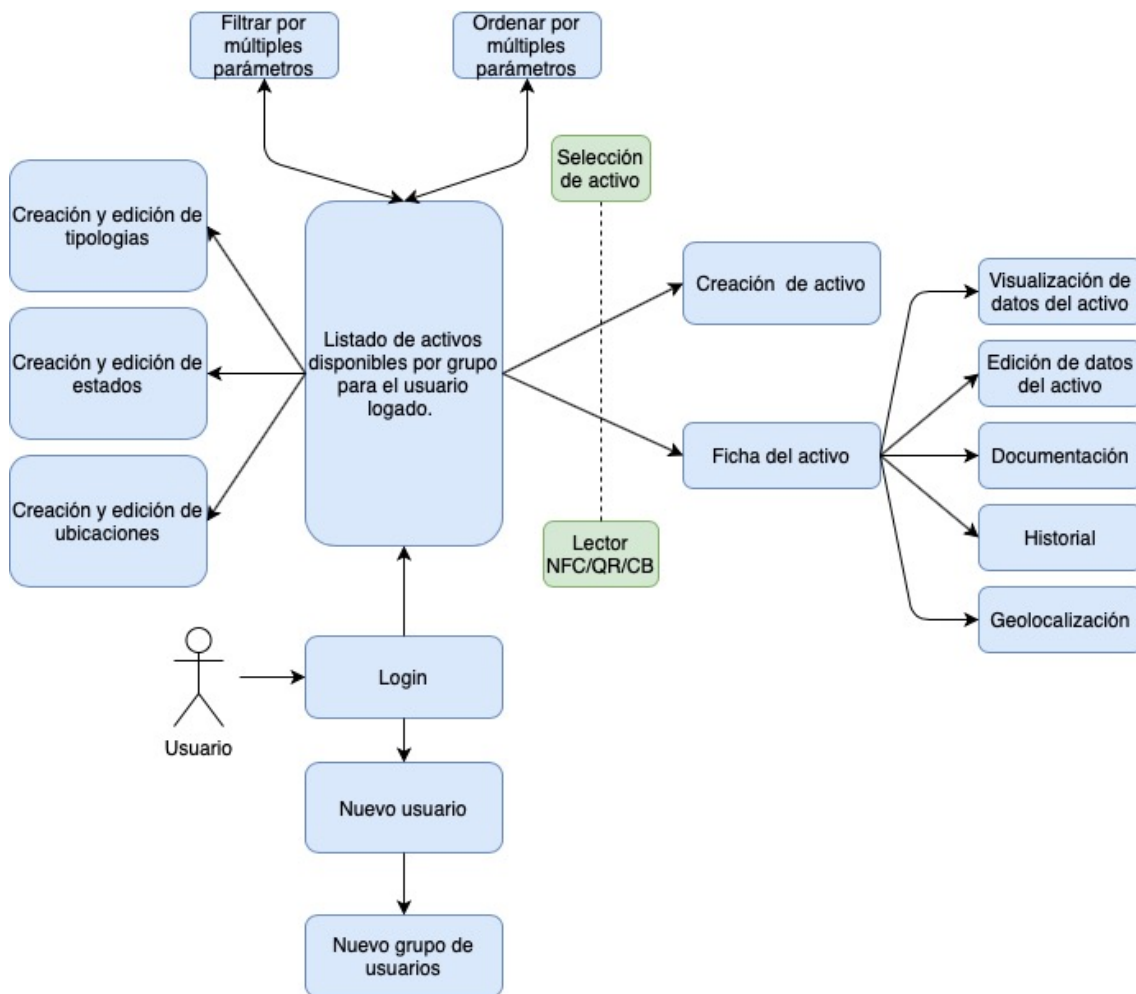
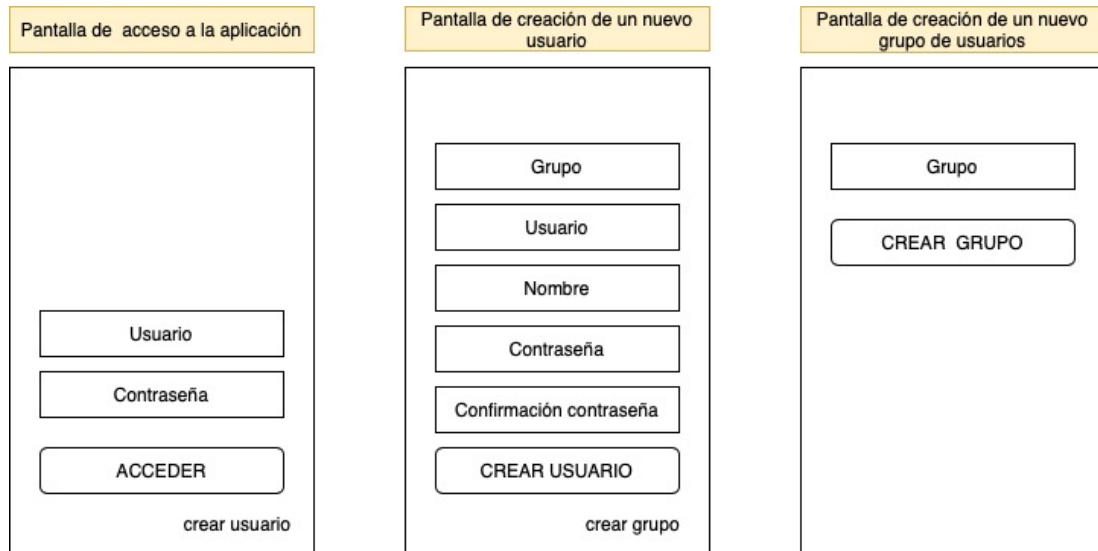


Ilustración 17. Diagrama de interacción

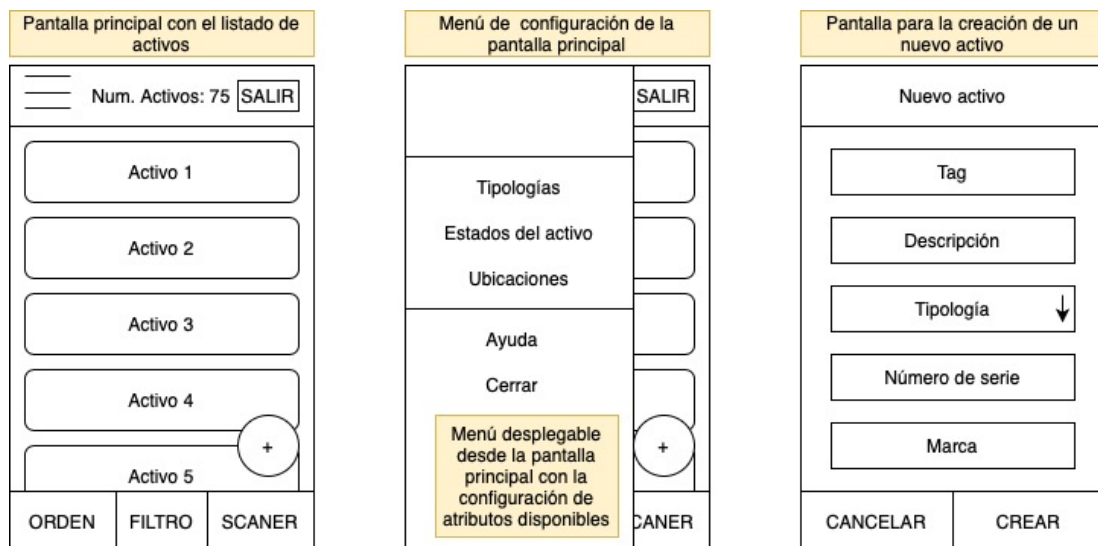
El diagrama de iteración muestra las principales pantallas de la aplicación y sirve de guía para el desarrollo de los prototipos que se muestran continuación.

### 2.3.3 Prototipo de baja fidelidad

Mediante la creación de un prototipo de baja fidelidad se pretende mostrar, a modo de boceto, la disposición de los elementos dentro de las pantallas de la aplicación, para obtener una idea de como el usuario va a interactuar con los mismos.



En la primera imagen, arriba, se puede apreciar la pantalla de acceso, donde el usuario tendrá que introducir su usuario y su contraseña para acceder a la pantalla principal, la cual contiene el listado de activos disponibles. Desde esta misma pantalla de acceso, el usuario, si es la primera vez que accede al sistema, podrá crear un nuevo usuario (segunda imagen arriba), y dentro de la pantalla que permitirá crear dicho usuario también podrá crear un grupo (tercera imagen arriba). El grupo puede representarle únicamente a él o bien a un conjunto de usuarios, por ejemplo una empresa, y será lo que determine la visibilidad sobre los activos y sus características. Siempre que se genere un nuevo usuario, este debe vincularse a un grupo, indicando el código único del mismo.



La pantalla principal (primera y segunda imagen arriba) contendrá un listado de los activos disponibles, así como, siempre visible, los botones para ordenar dicho listado por una serie de parámetros, filtrar los elementos del listado por una serie de parámetros y activar el scanner de la cámara para acceder directamente a un activo escaneando su etiqueta QR o CB (código de barras), o, en caso de no encontrarlo, crear uno nuevo activo. Desde esta pantalla también se podrá pasar el dispositivo por una etiqueta NFC con el mismo fin. La pantalla principal dispondrá de un menú desplegable lateral desde el que se podrán configurar los listados de atributos disponibles para los activos del grupo. Esto hará que cada grupo disponga de los valores en los atributos que mas se ajusten a sus necesidades. Desde el botón flotante en el margen inferior derecho se podrá crear un nuevo activo.

La pantalla de creación del activo (tercera imagen arriba) permitirá registrar los valores del nuevo activo.

Pantalla de edición de activo. Datos básicos			Pantalla de edición de activo. Documentación			Pantalla de edición de activo. Historial		
ACTIVO: Núm. TAG			ACTIVO: Núm. TAG			ACTIVO: Núm. TAG		
Datos básicos	Documentos	Historial	Datos básicos	Documentos	Historial	Datos básicos	Documentos	Historial
Tag			Documento 1			Registro de histórico 1		
Descripción			Documento 2			Registro de histórico 2		
Tipología ↓			Documento 3			Registro de histórico 3		
Número de serie			Crear nuevo documento			Registro de histórico 4		
CANCELAR GUARDAR			CANCELAR GUARDAR			CANCELAR GUARDAR		

La pantalla de edición será accesible pulsando sobre el activo del listado de activos de la pantalla principal

La pantalla de edición del activo permitirán visualizar la información del activo, modificarla, y/o añadir información complementaria al mismo. Esta pantalla será accesible desde la lectura de un código NFC, QR, código de barras o bien seleccionando el activo desde el listado de la pantalla principal. Esta pantalla permitirá navegar entre una serie de opciones del activo como son: datos básicos del activo, documentación asociada e historial de movimientos y cambios de estado.

Visualización y edición de un documento de activo

ACTIVO: Núm. TAG		
Datos básicos	Documentos	Historial
<div style="border: 2px solid black; padding: 5px;"> <p>Documento1</p> <p style="text-align: center;">DOCUMENTO</p> <p>Descripción</p> </div>		
CANCELAR		GUARDAR CAMBIOS
CANCELAR	GUARDAR	

Creación de un nuevo documento de activo

ACTIVO: Núm. TAG		
Datos básicos	Documentos	Historial
<div style="border: 2px solid black; padding: 5px;"> <p>Nuevo documento</p> <p style="text-align: center;">SUBIR DOCUMENTO</p> <p>Descripción</p> </div>		
CANCELAR		CREAR
CANCELAR	GUARDAR	

Las pantallas de la imagen superior representan la forma en que pretenden mostrarse y visualizarse los datos de documentación de un activo. Pulsando en la pestaña correspondiente a la parte de documentos en la pantalla de edición del activo podrá visualizarse el listado de documentos, con opciones para editar la información, así como para añadir nuevos documentos, posiblemente haciendo uso de pantallas de diálogo.

Listado de tipologías

Tipologías	
Tipología 1	Eliminar
Tipología 2	Eliminar
Tipología 3	Eliminar
<input type="button" value="CANCELAR"/> <input type="button" value="+"/>	

Creación de tipología

Tipologías	
Tipología 1	Eliminar
<div style="border: 2px solid black; padding: 5px;"> <p>Nueva tipología</p> <p style="text-align: center;">Tipología</p> </div>	
<input type="button" value="CANCELAR"/> <input type="button" value="CREAR"/>	
<input type="button" value="CANCELAR"/> <input type="button" value="+"/>	

Listado de estados

Estados	
Estado 1	Eliminar
Estado 2	Eliminar
Estado 3	Eliminar
<input type="button" value="CANCELAR"/> <input type="button" value="+"/>	

Creación de estado

Estados	
Estado 1	Eliminar
<div style="border: 2px solid black; padding: 5px;"> <p>Nuevo estado</p> <p style="text-align: center;">Estado</p> </div>	
<input type="button" value="CANCELAR"/> <input type="button" value="CREAR"/>	
<input type="button" value="CANCELAR"/> <input type="button" value="+"/>	

Las pantallas superiores representan pantallas de configuración de valores de atributos, accesibles desde el menú desplegable lateral de la pantalla principal. Al pulsar sobre cada opción en el menú lateral del menú principal, se mostrará el listado de valores disponibles, pudiendo editarlos pulsando sobre ellos o bien crear uno nuevo pulsando el botón flotante de la pantalla.

### 2.3.4 Diseño para la interacción

Con la finalidad de conseguir que la aplicación a desarrollar sea un producto fácil de utilizar, con una interfaz gráfica sencilla pero atractiva, que sea funcional, y que tenga una curva de aprendizaje rápida para cualquier usuario que utilice el producto, las guías de diseño a seguir se establecen como sigue:

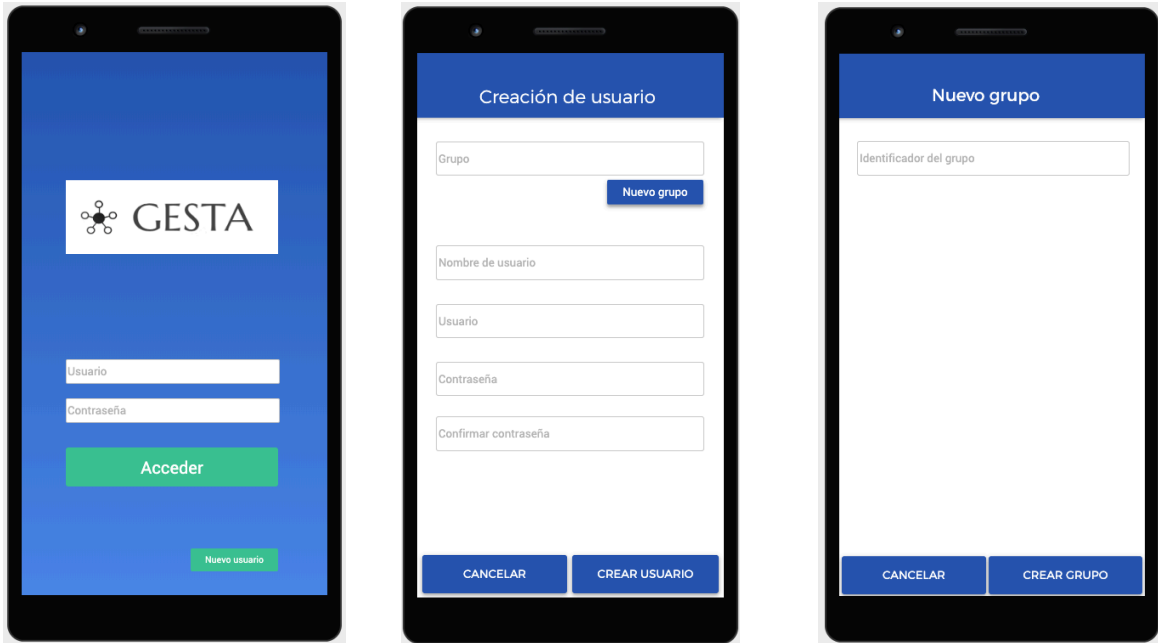
- Los elementos pulsables tienen que parecer que lo son, por lo que se hace uso de botones, listas e iconos, elementos de interacción fácilmente reconocibles por cualquier tipo de usuario que además tan solo requieren un toque para ser activados.
- La aplicación se diseña para pantallas pequeñas, es decir, para un smartphone, aunque debe poder ser utilizada perfectamente por un dispositivo más grande como una tableta. El diseño de pantalla es vertical.
- Al tratarse de una aplicación que es una herramienta de trabajo, la elección de los colores se ha priorizado por una cuestión principalmente funcional. La aplicación solo cuenta con tres colores, el principal, un azul oscuro, el negro y el blanco. Estos colores tienen como característica principal que forman un fuerte contraste entre ellos, lo que ayuda a que la presentación de la información sea clara.
- Se elige una tipografía estándar, para garantizar una buena legibilidad, y sin serifa para que el texto se presente limpio y sea cómodo de leer. El tamaño del texto es lo suficientemente grande para ser leído de forma cómoda pero no en exceso para aprovechar el espacio de la pantalla del dispositivo, que generalmente no es muy grande, y mostrar así toda la información necesaria.
- Los iconos serán los iconos que ofrece Google integrados en su herramienta de desarrollo Visual Studio, iconos que cualquier usuario acostumbrado a este sistema operativo puede identificar fácilmente.
- Se ha buscado un logo como representación de la aplicación en la línea descrita anteriormente: con un diseño sencillo, con colores que armonicen con los utilizados en la aplicación ya comentados y con un estilo profesional para una aplicación que ha de utilizarse como herramienta de trabajo.



### 2.3.5 Prototipo de alta fidelidad

Aplicando las guías marcadas en el punto anterior se genera un prototipo de alta fidelidad que debe servir para tener una imagen aproximada de lo que será la aplicación a desarrollar y que permita a potenciales usuarios realizar un proceso de test para garantizar que el producto diseñado cumple con los requisitos funcionales y de diseño necesarios.

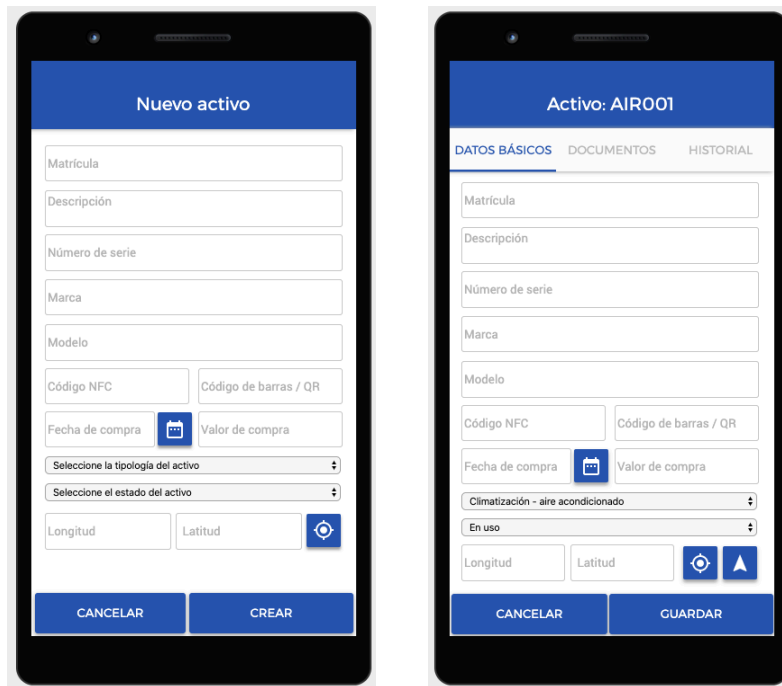
A continuación se muestra las imágenes del resultado obtenido:



Pantallas de login, creación de usuario y creación de un nuevo grupo.



Pantalla principal. Pantalla principal con el menú de opciones lateral abierto.



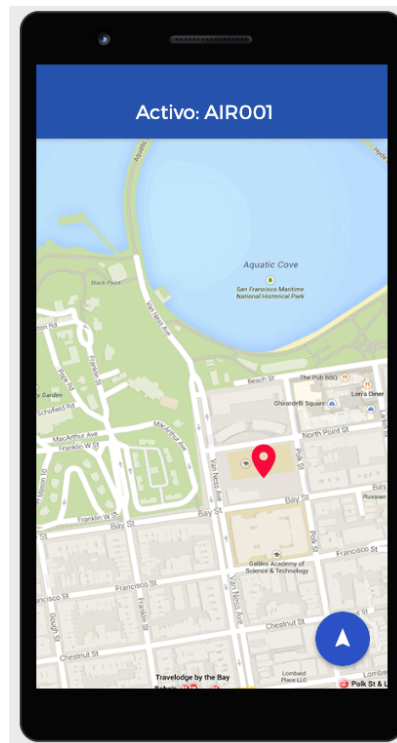
Pantalla de creación y edición del activo.



Pantalla de ordenación y filtrado.



Pantallas de visualización, edición y acceso a la creación de atributos de activos.



Pantalla de geocalización de un activo.

## 2.4 Evaluación

Se envía el prototipo de alta fidelidad a probar a varias personas, entre las cuales se encuentran posibles usuarios potenciales de la aplicación, a lo largo de varios días. Como resultados cabe destacar:

- En general la interfaz a nivel de diseño gráfico gusta.
- Los usuarios reconocen enseguida los elementos pulsables.
- Los usuarios aportan pequeñas mejoras sobre algunos elementos, por ejemplo botones en relación a un aspecto estético (sombras y relieve).
- Algunos usuarios opinan que la herramienta a nivel de colorido es demasiado sencilla. Sin embargo, la opinión de aquellos usuarios potenciales de la aplicación, es que el juego de colores escogido es correcto y se sienten cómodos con el mismo durante su uso.
- Respecto a la funcionalidad los usuarios no encuentran, en general, problemas de uso.
- La pantalla de ordenación del listado de activos es la que genera más dudas, ya que a primer golpe de vista algunos usuarios no intuyen cómo se utiliza. Se explica a los usuarios que tan solo hay que pulsar sobre los elementos a seleccionar en el orden que se desea y luego se puede pulsar sobre la flecha que aparece en cada uno de ellos para determinar si el orden ha de ser ascendente o descendente. Tras la explicación, los usuarios que realizan el test lo entienden perfectamente. Además, estos usuarios agradecen la funcionalidad que ofrece el diseño implementado, por lo que se decide mantenerlo frente a otra posible diseño más intuitivo.

## 2.5 Casos de uso

Con el objetivo de definir formalmente las funcionalidades de la aplicación derivadas del análisis de la fase DCU, se genera un diagrama de casos de uso:



Ilustración 18. Diagrama de casos de uso

A partir del diagrama anterior se genera el listado de casos de uso de la aplicación.

<b>Nombre</b>	<b>Iniciar sesión</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario abre la aplicación
<b>Flujo</b>	El usuario cumplimenta la casilla de usuario con su usuario
	El usuario cumplimenta la casilla de contraseña con su contraseña
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “acceder” para validar el usuario y la contraseña introducidos y poder acceder a la pantalla principal
<b>Nombre</b>	<b>Crear usuario</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario abre la aplicación
<b>Flujo</b>	El usuario pulsa sobre “crear usuario” en la pantalla de acceso de la aplicación
	Se abre la pantalla de creación de usuario
	El usuario cumplimenta, por lo menos, los datos obligatorios para la creación de usuario, que son el código del grupo, el usuario y la contraseña
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “crear usuario” para generar el nuevo usuario
<b>Nombre</b>	<b>Crear grupo de usuarios</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario abre la aplicación
<b>Flujo</b>	El usuario pulsa sobre “crear usuario” en la pantalla de acceso de la aplicación
	Se abre la pantalla de creación de usuario
	El usuario pulsa sobre “crear grupo” en la pantalla de creación de usuario
	Aparece la pantalla de creación de grupo
	El usuario cumplimenta el nombre del grupo
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “crear grupo” para generar el nuevo grupo. El sistema le retorna el código único del grupo generado

<b>Nombre</b>	<b>Búsqueda de activos</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario se ha identificado en la aplicación y ha accedido a la pantalla principal con la lista de activos disponibles
<b>Flujo</b>	1A. El usuario pasa el dispositivo por una etiqueta NFC
	1B. El usuario pulsa el botón con la opción de escáner
	2B. El usuario pasa el dispositivo con el escáner activo por una etiqueta QR o CB (código de barras)
	1C. El usuario pulsa sobre la opción “filtro” y cumplimenta alguna opción con la información que conoce relativa al activo que desea buscar
<b>Postcondiciones</b>	Se realiza la búsqueda y el listado de activos se actualiza mostrando el resultado
<b>Nombre</b>	<b>Ver / editar activo</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario se ha identificado en la aplicación y ha accedido a la pantalla principal con la lista de activos disponibles
<b>Flujo</b>	El usuario realiza la búsqueda de un activo
	El usuario pulsa en el listado de activos sobre el activo que desea ver o editar
	El usuario accede a la ficha del activo
	El usuario navega sobre las opciones de la ficha para ver o editar la información deseada
	El usuario modifica la información deseada en la ficha del activo en caso de querer editarlo
<b>Postcondiciones</b>	El usuario pulsa “guardar” para validar los cambios realizados en la ficha del activo o “cancelar” para salir sin guardar datos
<b>Nombre</b>	<b>Crear activo</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario se ha identificado en la aplicación y ha accedido a la pantalla principal con la lista de activos disponibles
<b>Flujo</b>	1A. El usuario realiza la búsqueda de un activo que no está registrado en el sistema
	2A. Aparece una pantalla ofreciendo la opción de crear un nuevo equipo
	1B. El usuario pulsa sobre el botón flotante con el símbolo + de la pantalla principal
	Aparece la pantalla de creación de activo
	El usuario cumplimenta los datos del nuevo activo
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “crear” de la pantalla para generar un nuevo activo

<b>Nombre</b>	<b>Ver / editar atributo de activo</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario se ha identificado en la aplicación y ha accedido a la pantalla principal con la lista de activos disponibles
<b>Flujo</b>	El usuario pulsa sobre el botón del menú lateral de la pantalla principal
	El usuario selecciona el atributo del cual quiere visualizar la información
	Aparece un listado con los atributos disponibles
	El usuario pulsa sobre el atributo que desea editar y aparece la pantalla de edición del atributo
	El usuario realiza los cambios en la nomenclatura del atributo
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “guardar” para guardar los cambios
<b>Nombre</b>	<b>Crear atributo de activo</b>
<b>Actores</b>	Usuario de la aplicación
<b>Precondiciones</b>	El usuario se ha identificado en la aplicación y ha accedido a la pantalla principal con la lista de activos disponibles
<b>Flujo</b>	El usuario pulsa sobre el botón del menú lateral de la pantalla principal
	El usuario selecciona el atributo del cual quiere visualizar la información
	Aparece un listado con los atributos disponibles
	El usuario pulsa sobre el botón flotante del listado de atributos con el símbolo +
	Aparece la pantalla para la generación de un nuevo atributo
	El usuario cumplimenta la información del nuevo atributo
<b>Postcondiciones</b>	El usuario pulsa sobre el botón “crear” para generar un nuevo atributo



## 2.6 Diseño de la arquitectura

### 2.6.1 Base de datos

La aplicación dispone de dos base de datos, una local SQLite relacional en el dispositivo en la que se debe almacenar toda la información que el usuario pueda necesitar para trabajar, tenga el dispositivo o no, acceso a internet, y otra en la nube, la base de datos de Firebase, donde se almacenarán todos los datos de todos los dispositivos para todos los usuarios de la aplicación. La aplicación debe mantener sincronizadas ambas base de datos, insertando en Firebase los registros generados en los dispositivos, descargando la información de Firebase en la base de datos local en todos los dispositivos de usuarios con visibilidad, actualizando esta información de una base de datos a otra, etc.

A continuación, se muestra el diagrama del diseño de la base de datos local.

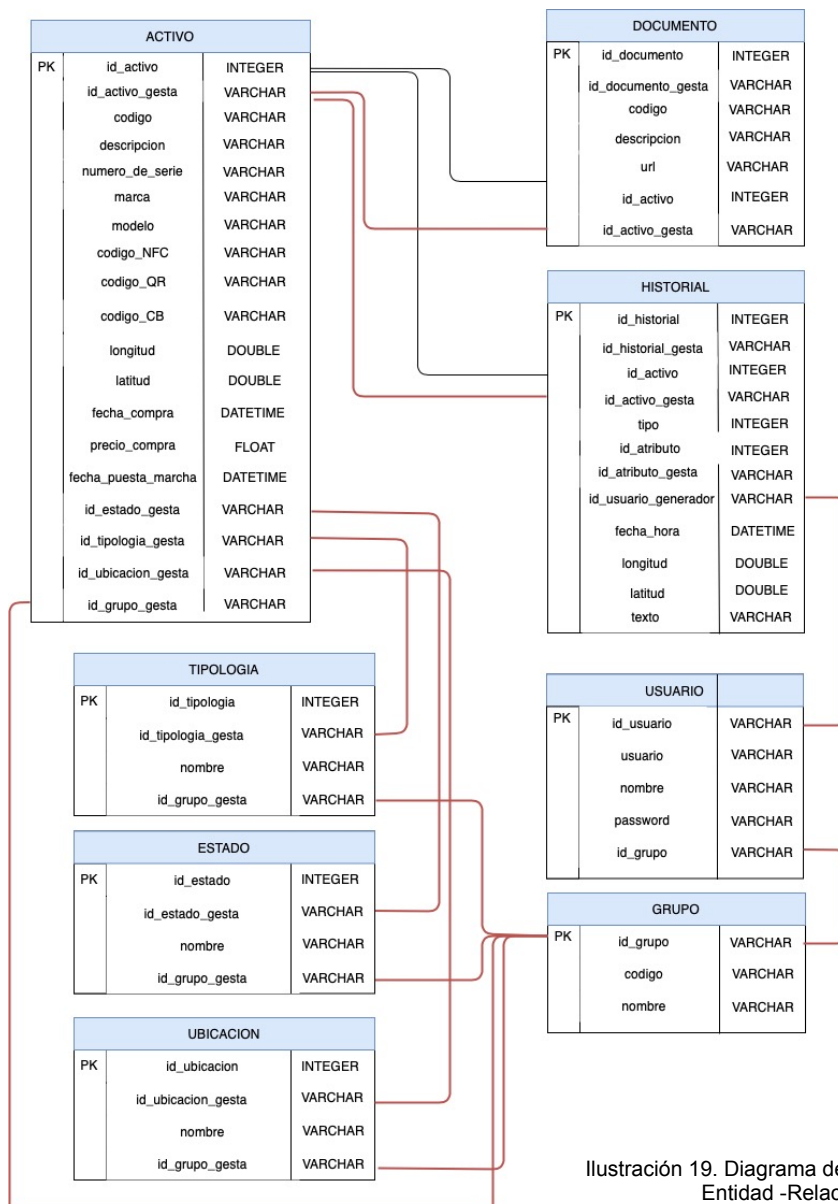


Ilustración 19. Diagrama de base de datos Entidad -Relación

Las líneas marcadas en color rojo muestran las asociaciones entre campos que corresponden a las claves primarias de la base de datos GESTA, es decir, la base de datos global alojada en Firebase. La base de datos de Firebase mantiene la misma estructura que la base de datos local SQLite y contiene los mismos campos, a excepción de los identificadores que se utilizan en la base de datos local SQLite de cada dispositivo y que actúan como claves primarias. Esto es así debido a que los dispositivos deben poder trabajar sin necesidad de tener acceso a la red, motivo por el cual se requiere que la información esté en el dispositivo obligando a definir en la base de datos local claves primarias que serán las que se utilizarán para establecer las relaciones entre entidades mientras no se pueda realizar la inserción de los nuevos registros en Firebase. Por otro lado, las entidades que representan atributos como serían las tipologías, estados o ubicaciones, solo estarán disponibles para asociarlas a los activos una vez estén dadas de alta en Firebase.

### 2.6.2. Arquitectura de la aplicación

La arquitectura escogida para implementar la aplicación se basa en el patrón MVVM (Model-View-ViewModel / Modelo-Vista-VistaModelo) lanzado por Android en su “Guía para implementar aplicaciones”. Este patrón proporciona, entre otras, las siguientes ventajas:

- MVVM proporciona un código limpio y por tanto legible.
- Cada capa de la vista modelo (ViewModel) está desacoplada de la vista, lo que permite realizar test unitarios sobre ella con facilidad, es mas reusable y le afectan menos los cambios en la vista.
- La aplicación obedece a una arquitectura de capas, donde cada componente depende únicamente del inmediatamente inferior.
- Como resultado obtenemos una aplicación robusta que ofrece una gran experiencia de usuario y que resulta fácilmente escalable.

A continuación se muestra un diagrama explicativo de la arquitectura a utilizar sobre la aplicación a implementar:

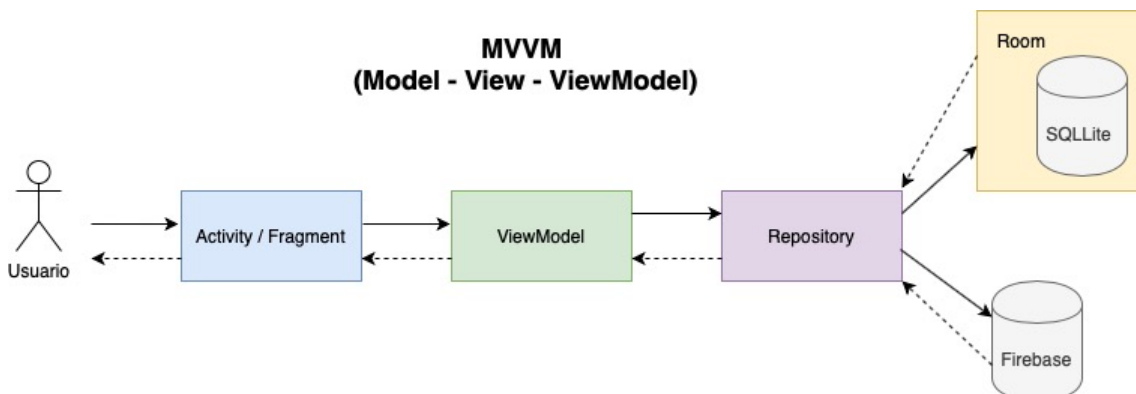


Ilustración 20. Diagrama de la arquitectura de la aplicación

Como se puede observar en el diagrama anterior, la implementación de este patrón consiste en implementar las capas de forma que cada una de ellas solo dependa de la capa inmediatamente inferior, esto es:

- La capa de vista estaría formada por las actividades y fragmentos y dependerá exclusivamente de la capa vista modelo. La vista llamará a métodos disponibles en esta capa vista modelo y se implementarán observadores en la vista para capturar cambios en los objetos que utilice la vista modelo.
- La capa vista modelo dependerá de las clases de repositorio que serán las encargadas de comunicarse con las bases de datos SQLite y Firebase.

Toda la lógica de la aplicación se gestionará en las clases vista modelo, no implementando ningún tipo de lógica en la capa de vista, exceptuando la lógica de presentación.

Se pretende que cada actividad llame a una clase vista modelo. Esta vista modelo tiene la ventaja de poder ser compartida por la actividad y por los fragmentos de la actividad si esta los tuviese.

Actividades como la edición de un activo, que, a parte de mostrar datos propios de la entidad del activo, es necesario que muestre datos de entidades como el listado de tipologías, estados, etc, pueden requerir que en la clase vista modelo correspondiente se trabaje con un modelo lógico que abarque todas estas entidades.

## 3. Desarrollo

### 3.1 Organización del proyecto

Como ya se avanzó en los capítulos de “Arquitectura de la aplicación”, se ha implementado la aplicación para dispositivos Android utilizando la arquitectura MVVM (modelo-vista-modelo de vista o model-view-viewmodel). Con el fin de mantener el contenido coherente a la arquitectura escogida y correctamente organizado, se han generado una serie de paquetes, cada uno de los cuales contiene un conjunto de archivos de un determinado tipo. En la siguiente imagen se puede observar la organización de estos paquetes dentro del proyecto de la aplicación:

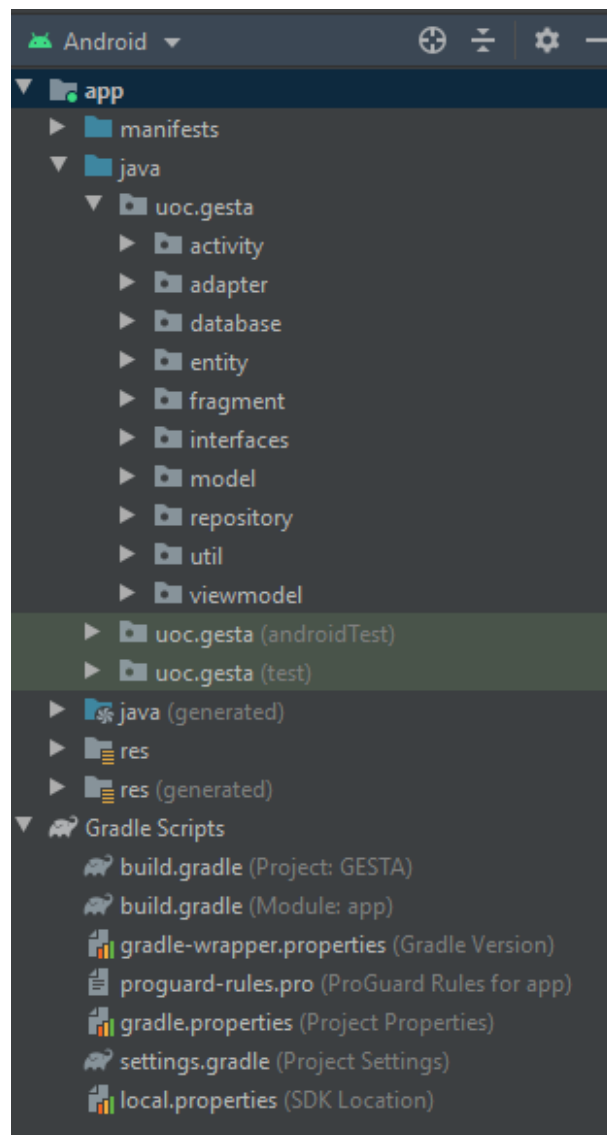


Ilustración 21. Organización del proyecto en paquetes

A continuación se describe brevemente cada paquetes y su contenido:

- Paquete **activity**: este paquete contiene todas las actividades de la aplicación, que, junto con los fragmentos, forman la parte de vista (view).
- Paquete **fragment**: este paquete contiene todos los fragmentos de la aplicación y el fichero "PageAdapter" para incluirlos en la actividad de creación/edición del activo. Como se ha comentado anteriormente, estos fragmentos, junto con las actividades comentadas en el punto anterior, forman la parte de vista (view) del proyecto.
- El paquete **adapter** contiene todos los adaptadores personalizados para los listados y para los spinner (dropdown) del proyecto.
- El paquete **database** contiene la clase que extiende de RoomDatabase en la que se define un array con las entidades de la base de datos, los objetos Dao y la versión de la aplicación. También contiene una clase con la definición de los convertidores de tipos complejos utilizados en la base de datos (fechas almacenadas como timestamp (long)) y la clase de creación y acceso a la base de datos local Room.
- El paquete **entity** contiene las clases donde se definen las entidades de la base de datos, y las interfaces Dao, donde están definidos los métodos para acceder a la base de datos local mediante Room.
- El paquete **interfaces** contiene las interfaces que se utilizan para comunicar la respuesta de los repositorios con los viewmodels del proyecto.
- El paquete **model** contiene los modelos lógicos complejos que ha sido necesario crear y que se utilizan en la aplicación cuando, por ejemplo, se requiere de estructuras que incluyen varias entidades.
- El paquete **repository** contiene los archivos de comunicación, tanto con la base de datos local SQLite a través de Room, con la base de datos de Firebase, y con el sistema de almacenamiento en la nube de Firebase (Storage).
- El paquete **viewmodel** contiene todos los ficheros que forman parte del modelo de vista. Estos ficheros contienen la lógica de la aplicación y comunican tanto con la vista (actividades y fragmentos) como con los repositorios que acceden a las bases de datos y al almacenamiento Firebase, siendo el único puente entre ellos.
- El paquete **util** contiene archivos con variables globales, métodos de encriptación de contraseñas, ficheros de sincronización, y otros métodos de utilidad.

## 3.2 Guía de implementación

Tras estudiar la situación actual sobre la fragmentación que sufren los dispositivos Android, se ha escogido la API 24 de Android (Android 7.0) como versión mínima para la aplicación, ya que Android Studio garantiza que el 73.7% de los dispositivos actuales soportan dicha versión o posteriores, además de incluir cambios de comportamiento en esta versión como modificaciones en los permisos o formas de compartir contenido de archivos privados con FileProvider, entre otros.

El idioma por defecto establecido para los textos es el español, aunque el idioma de implementación es el inglés. Se han traducido e implementado los textos de la aplicación también a inglés y catalán.

La nomenclatura de los ficheros sigue el estilo UpperCamelCase o PascalCase y se establece como se muestra a continuación:

- Nombre de las actividades: NombreActivity (AssetActivity)
- Nombre de los adaptadores: NombreAdapter (AssetDocumentAdapter)
- Nombre de las entidades: Nombre (Asset)
- Nombre de las clases Dao: NombreEntidadDao (AssetDao)
- Nombre de los fragmentos: NombreFragment (AssetBasicDataFragment)
- Nombre de las interfaces: NombreInterface (AssetStateViewModelInterface)
- Nombre de modelos lógicos: NombreModel (AssetModel)
- Nombre de las clases repositorio: NombreRepository (AssetModelRepository)
- Nombre de las clases vistas de modelo (viewmodel): NombreViewModel (AssetViewModel)

Los nombres de los métodos, así como el nombre de las variables, siguen el estándar propuesto por Google y se han escrito siguiendo el estilo lowerCamelCase.

Para comunicar mensajes desde la vista modelo a la vista se hace uso de observadores, mientras que para comunicar mensajes desde el repositorio a la vista modelo se hace uso de interfaces. El siguiente esquema muestra el patrón comentado:

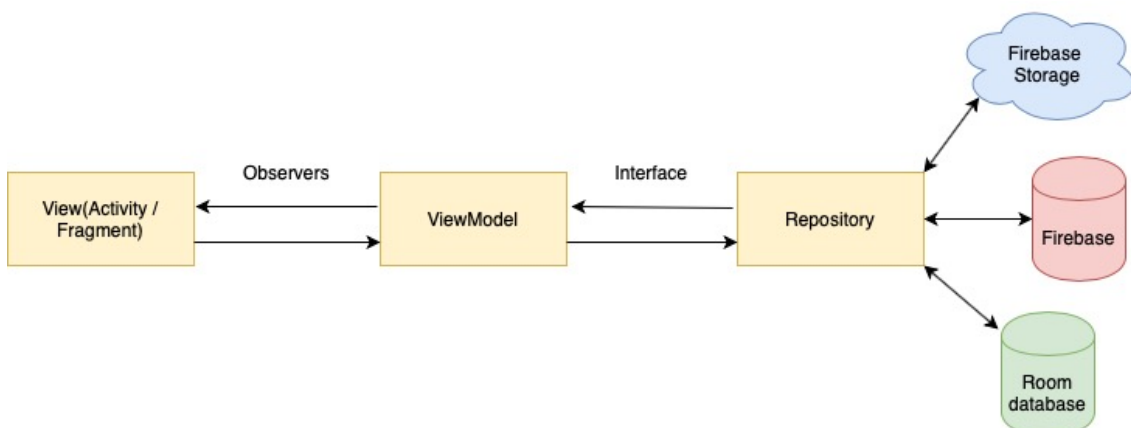


Ilustración 22. Comunicación entre capas

## 3.2 Tecnologías utilizadas

### 3.2.1. Firebase

La aplicación requiere que los usuarios puedan compartir datos en tiempo real siempre que pertenezcan al mismo grupo de trabajo (por ejemplo una empresa). Esto implica que la aplicación debe almacenar esta información en algún sistema de persistencia de datos accesible desde diferentes dispositivos. Para dar solución a esta necesidad se ha escogido utilizar Firebase.

Firebase es una herramienta de Google que facilita la implementación de aplicaciones móviles y permite, entre otras cosas, autenticar usuarios en el sistema, utilizar una base de datos no relacional JSON, almacenamiento de datos en la nube, etc.

#### 3.2.1.1 Realtime Database

La información de activos y estructuras relacionadas de activos con las que han de trabajar los usuarios, deben estar disponibles aún cuando los dispositivos no tengan acceso a internet. Por ello esta información debe estar disponible en la base de datos local del dispositivo (SQLite). Sin embargo, la necesidad de compartir esta información entre usuarios de un mismo grupo de trabajo, hace que sea necesario utilizar un sistema de persistencia de datos global accesible por todos ellos. Para ello se ha utilizado la base de datos JSON Realtime Database de Firebase.

Tanto si el usuario trabaja en modo online como en modo offline (sin acceso a internet), los datos que alimentan a la aplicación y aquellos que se generan, se leen y almacenan directamente en la base de datos local SQLite. Sin embargo, cuando el modo de trabajo del usuario es online (el usuario tiene acceso a internet), la base de datos local del dispositivo se alimenta a su vez de los datos que existen en la base de datos de Firebase. Firebase tiene mecanismos que permiten a la aplicación obtener los nuevos datos y nuevas modificaciones cuando otros usuarios realizan cambios, de forma que la aplicación los inserta o actualiza en la base de datos local del usuario en tiempo real. Estos mecanismos, junto con el sistema LiveData de Google, hacen que un usuario que trabaja en modo online tenga la información siempre actualizada y disponible en su dispositivo.

Cuando un usuario trabaja en modo online, al generarse nuevos datos o realizarse modificaciones sobre los mismos, estos cambios se graban tanto en la base de datos local SQLite como en la base de datos de Firebase. Sin embargo, cuando se trabaja en modo offline, sin acceso a internet, estos cambios solo se guardan en la base de datos local SQLite, ya que el acceso a la base de datos de Firebase no está disponible. Estos datos, no obstante, se graban con una marca que indica que, en el momento en que el usuario trabaje nuevamente en modo online en su dispositivo, se sincronizarán con la base de datos de Firebase.

Al iniciar la sesión de usuario en la aplicación en modo online, se activa el proceso de sincronización de datos, tanto para generar nuevos datos y modificaciones desde la base de datos local a Firebase, como a la inversa, es decir, para descargar los nuevos datos de la base de datos de Firebase a la local. Este proceso de sincronización se mantiene activo durante toda la sesión del usuario gracias a la tecnología que proporciona Firebase para trabajar con su base de datos en tiempo real.

### 3.2.1.2 Storage

La aplicación ha de permitir adjuntar documentos asociados a los activos, como fotografías, manuales, garantías, etc. Estos documentos deben estar disponibles para todos los usuarios que tengan acceso a los activos de un mismo grupo de trabajo. Firebase proporciona un servicio de almacenamiento en la nube de nombre Firebase Storage.

Se ha implementado un sistema de documentación en la aplicación que permite la carga de un archivo, bien mediante la toma de una fotografía o bien adjuntando un documento existente en el dispositivo del usuario, y su descarga en el sistema de almacenamiento en la nube de Firebase, donde queda accesible para otros usuarios que han accedido a la aplicación desde diferentes dispositivos. Los datos para el acceso y la descripción de este archivo se graban en un registro de la entidad documentos, que lo relaciona con el activo correspondiente y el grupo de trabajo al que pertenece.

El acceso a la documentación de un activo es una funcionalidad que solo está disponible en el modo de trabajo online, es decir, con acceso a internet, ya que los ficheros que forman parte de la documentación están almacenados, como ya se ha expuesto, en el sistema de almacenamiento en la nube de Firebase. Firebase proporciona la adecuada tecnología para la carga y descarga de ficheros en la nube.

### 3.2.1.3 Firebase Authentication

Para acceder al servicio de almacenamiento en la nube de Firebase (Firebase Storage), así como acceder a la base de datos en tiempo real de Firebase (Realtime Database) de forma segura, ha sido necesario implementar el servicio de autenticación de Firebase. Firebase proporciona un mecanismo de autenticación de usuarios mediante varias combinaciones de parámetros, que, mediante la configuración de unas reglas de acceso, permite acceder de forma segura a la base de datos que proporciona o al servicio de almacenamiento en la nube, entre otros.

En el caso particular de la aplicación implementada, se ha escogido la autenticación mediante correo electrónico y contraseña (encriptada por la aplicación mediante el algoritmo AES utilizando una clave privada particular). De este modo, tanto para acceder a la base de datos de Firebase, como para



acceder al servicio de almacenamiento en la nube de Firebase, se requiere estar autenticado en Firebase. Cuando el usuario se autentica correctamente en la aplicación en modo online contra los datos existentes en la base de datos en la entidad usuarios, su usuario y su password se envían a Firebase para realizar una segunda autenticación en el sistema de Firebase. Esta autenticación se mantendrá durante toda la sesión de trabajo del usuario que ha accedido a la aplicación en el dispositivo.

En el caso del trabajo en modo offline, al no disponer de acceso a internet, y por tanto, no poder acceder a la base de datos de Firebase (Realtime Database) ni a su servicio de almacenamiento en la nube (Storage), no es necesaria esta autenticación, de igual forma que tampoco podría realizarse. En este caso, la autenticación de usuarios solo se realiza contra la base de datos local (SQLite) del dispositivo, que, siempre que el usuario haya realizado una autenticación online previamente, tiene almacenados los datos del usuario para permitirle autenticarse en modo offline cuantas veces necesite posteriormente.

### 3.2.2 Room

El posible trabajo en modo offline por parte de los usuarios de la aplicación requiere del uso de una base de datos local. Android permite crear una base de datos relacional SQL llamada SQLite para almacenar datos de forma persistente.

Google, a su vez, dispone y recomienda, Room, una capa de abstracción sobre la base de datos SQLite, que permite acceder y gestionar fácilmente el trabajo con SQLite a la vez que aprovecha toda su potencia.

Se han implementado todas las entidades necesarias del modelo de datos y se han etiquetado con la directiva `@Entity` de Room, así como se han definido las consultas necesarias de acceso en interfaces Dao. Las entidades implementadas se utilizan tanto en la base de datos local (SQLite) como para generar las estructuras que forman la base de datos global (Firebase).

### 3.2.3 Play services

En su paquete Play Services, Google ofrece varias librerías de utilidad. Algunas de estas librerías, se han utilizado para la implementación de la geolocalización de activos, tanto para la obtención de los datos en tiempo real, como para la ubicación de los mismos en un mapa.

### 3.2.4 Lectura de etiquetas de activos

Una de las características principales de la aplicación es la posibilidad de acceder fácilmente a los activos escaneando un código de barras o QR identificativo, así como poder hacer uso del lector NFC del dispositivo con el mismo fin. Para ello se ha implementado un lector propio que hace uso de la cámara del dispositivo para leer etiquetas de código de barras y QR, y por otro lado, se ha implementado la tecnología necesaria para realizar lecturas de etiquetas NFC, siempre y cuando el dispositivo disponga de un lector adecuado para ello.

Las etiquetas NFC pueden ser de varios tipos. Tras un estudio detallado sobre ellas se ha determinado dos posibles modos de lectura de dichas etiquetas que han sido implementados:

- Lectura de etiquetas MifareUltraligh
- Lectura de etiquetas MirafareClassic y otros modelos.

Estos tipos de etiquetas han sido escritos de forma diferente, lo que requiere que el sistema que codifica/descodifica el dato en hexadecimal una vez leído (dato impreso en la etiqueta), según el tipo de etiqueta sea inverso.

### 3.2.5 Material Design

Hace ya tiempo Google presentó Material Design como una serie de pautas de implementación del estilo y diseño, un concepto, una filosofía, para entre otras, implementar aplicaciones Android.

En el caso de GESTA, se han implementado la mayoría de vistas de la aplicación con elementos proporcionados por Material Design, haciendo uso de los archivos de estilos (style.xml) y de colores (colors.xml).

### 3.2.6. Android Jetpack

Android Jetpack es un conjunto de bibliotecas y herramientas que facilitan la creación de aplicaciones Android de forma sencilla.

Se han utilizado componentes de arquitectura para la realización de una aplicación sólida como LiveData o ViewModel, que permiten el uso de observadores y mantienen los datos actualizados, por ejemplo el listado de activos, con nuevos datos alimentados por el sistema de sincronización en el trabajo en modo online que pueden incorporarse a la base de datos local del dispositivo en cualquier momento.

### 3.3. Problemáticas detectadas durante la implantación y soluciones aplicadas

#### 3.3.1 Claves primarias

La necesidad de disponer de dos bases de datos, una local (SQLite) y otra global (Firebase Realtime Database) cuyos datos deben mantenerse sincronizados entre ambas no compartiendo la totalidad de los mismos (cada base de datos local solo debe disponer de los datos visibles para el usuario logrado en la aplicación), plantea la problemática de la generación y mantenimiento de las claves primarias de cada tabla o representación de la entidad de la base de datos.

Inicialmente se realizó el diseño de la aplicación para que cada base de datos tuviese sus propios identificadores, existiendo por cada tabla o representación de la entidad dos claves, la de la base de datos local autogenerada (SQLite) y la de la base de datos global, común para todos los dispositivos. Durante la fase de implementación se ha visto la dificultad y la problemática para mantener la coherencia de datos y trabajar con ambas claves, agravado por el posible modo de trabajo offline.

La solución a este problema ha sido utilizar una única clave primaria para las tablas o estructuras que representan las entidades de la base de datos. Para ello, esta clave, que necesariamente ha de generarse desde los dispositivos de los usuarios, debe ser única y no poder duplicarse en los diferentes dispositivos. La solución es el uso de UUID o identificadores únicos universales. Cuando un registro se crea por primera vez, la aplicación genera su clave UUID, garantía de que este identificador es único y no puede generarse otro igual que lleve a una inconsistencia de datos.

#### 3.3.2 Visualización de documentos

Durante la fase de implementación, una vez más, se ha detectado un problema que ha llevado a una ligera modificación del diseño propuesto inicialmente. En este caso la necesidad llega a través de la visualización de los documentos asociados a los activos.

Se ha implementado en la aplicación un visor para mostrar las imágenes adjuntas o tomadas mediante la cámara del dispositivo. Este visor, sin embargo, no permite visualizar otro tipo de documentos como un fichero de texto, un fichero pdf, un fichero excel, etc. Es por este motivo que se ha realizado una modificación en la correspondiente entidad de la base de datos (Documento) para añadir un atributo no definido en el diseño inicial que almacena el tipo de documento.

También se ha eliminado el campo código, definido inicialmente, dejando la identificación del documento por parte del usuario, a través del campo descripción, pues se ha considerado que el campo código no aportaba

información relevante para el usuario, y facilitando al usuario la generación del documento en cualquier caso.

### 3.3.3. Autenticación

El problema detectado durante la fase de implementación relativo a la autenticación de Firebase necesaria para acceder a la base de datos global (Firebase Realtime Database) ha sido uno de los más complicados de resolver.

Para poder trabajar con la aplicación se requiere realizar una autenticación o acceso indicando un usuario y un password, tanto para el trabajo en modo online como en modo offline. Para ello, el usuario debe haberse registrado previamente. La aplicación dispone de una funcionalidad, disponible desde la propia pantalla de acceso, para generar usuarios. Estos usuarios deben quedar registrados en la base de datos global de Firebase, por lo que el registro debe hacerse en modo online, es decir, con acceso a internet.

El problema se presenta cuando, para acceder a la base de datos de Firebase se requiere de una autenticación previa en Firebase mediante su servicio Authentication. Al no existir el usuario, aún, no es posible realizar esta autenticación en Firebase, y por tanto no es posible acceder a la base de datos de Firebase.

Para solucionar el problema anterior se decide crear un usuario "administrador" en Firebase. El usuario administrador, único, se genera directamente en Firebase, una sola vez, y sus datos se almacenan en la aplicación de forma interna de modo que no sean accesibles por ningún usuario. Estos datos son, los que la aplicación, siempre que tenga acceso a internet, utiliza, una vez se ejecuta, para realizar una autenticación mediante la cual se podrá dar de alta usuarios. Esta autenticación, no obstante, no sirve para trabajar con la aplicación y acceder, una vez realizado el acceso mediante usuario y contraseña, a la base de datos de Firebase o al almacenamiento en la nube de Firebase.

La autenticación con el usuario administrador que se realiza de forma automática y transparente para el usuario al iniciar la aplicación, solo se utiliza para crear usuarios.

### 3.3.4 Grupos de trabajo

Otro problema detectado en la fase de implementación es la creación de grupos.

Este problema está estrechamente relacionado con el problema detectado en el punto anterior, pero al que hay que sumar, que para poder generar un usuario, debe haberse generado el grupo de trabajo al que debe pertenecer previamente, ya que será este el que establezca la visibilidad que el usuario tendrá sobre los activos y sus propiedades.

Estudiando diferentes opciones, se ha escogido la siguiente: los grupos de trabajo los debe generar el administrador de la aplicación bajo petición de los usuarios.

Este es un método, además, de establecer un posible modelo de negocio, ya que puede ser una forma de gestionar los pagos por el uso de la aplicación. La aplicación podría estar disponible en el mercado de aplicaciones, de forma gratuita para todos los usuarios, pero quienes quisieran utilizarla pasarían por solicitar un grupo de trabajo a cambio de un coste determinado al administrador de la aplicación, por ejemplo, mediante el envío de un correo electrónico.

Para facilitar el trabajo del administrador a la hora de crear estos grupos se ha generado una versión de la aplicación con una opción que permite la creación de estos grupos.

## 4. Plan de pruebas

Con tal de garantizar el correcto funcionamiento de la aplicación, se ha diseñado y realizado el juego de pruebas completo que se detalla a continuación:

Tipo	Modo	Prueba	Resultado
Administración	Online	Visualización de las opciones de administración: creación de grupos	Correcto
Administración	Online	Creación de grupo (Firebase)	Correcto
Crear usuario	Online	Creación de usuario. Comprobación de creación sin datos básicos. Mensajes de error por pantalla	Correcto
Crear usuario	Online	Comprobación de contraseña y verificación de contraseña. Mensajes de error por pantalla	Correcto
Crear usuario	Online	Comprobar que el grupo de usuarios existe en Firebase database al intentar crear un usuario	Correcto
Crear usuario	Online	Comprobar que el usuario se genera en Firebase Database y en Firebase Authentication	Correcto
Crear usuario	Offline	Comprobar funcionamiento al crear el usuario si se pierde la cobertura estando en esta actividad. Mensaje de error por pantalla	Correcto
Login	Online/ Offline	Comprobar conexión y activar el modo online/offline pulsando sobre el icono correspondiente (wifi activado/wifi desactivado) de la aplicación	Correcto
Login	Offline	Comprobar que no se puede acceder en modo offline sin haber accedido previamente en modo online	Correcto
Login	Online	Comprobar que se realiza el login correcto mediante usuario y contraseña	Correcto
Login	Offline	Comprobar que se realiza el login correcto tras haber accedido una vez e el dispositivo con el usuario y contraseña indicadas	Correcto
Tipos de estados	Online	Al crear tipos de estados se añaden correctamente	Correcto
Tipos de estados	Online	Si un usuario del mismo grupo desde un terminal añade o modifica un estado, otro usuario del mismo grupo en otro terminal ve el resultado inmediatamente	Correcto
Tipos de estados	Offline	Se genera el estado en el dispositivo en local correctamente	Correcto

Tipo	Modo	Prueba	Resultado
Tipos de estados	Online/ Offline	Al generar un estado en modo offline, cuando este usuario accede nuevamente al dispositivo en modo online el estado se graba correctamente en la base de datos Firebase y este es visible inmediatamente por otros usuarios del mismo grupo	Correcto
Tipos de estados	Online	Se comprueba la visibilidad por grupos. Un estado solo lo puede ver un usuario del mismo grupo	Correcto
Tipología de activos	Online	Al crear tipologías de activos se añaden correctamente.	Correcto
Tipología de activos	Online	Si un usuario del mismo grupo desde un terminal añade o modifica una tipología, otro usuario del mismo grupo en otro terminal ve el resultado inmediatamente	Correcto
Tipología de activos	Offline	Se genera la tipología en el dispositivo en local correctamente	Correcto
Tipología de activos	Online/ Offline	Al generar una tipología en modo offline, cuando este usuario accede nuevamente al dispositivo en modo online la tipología se graba correctamente en la base de datos Firebase y esta es visible inmediatamente por otros usuarios del mismo grupo	Correcto
Tipología de activos	Online	Se comprueba la visibilidad por grupos. Una tipología solo la puede ver un usuario del mismo grupo	Correcto
Ubicaciones	Online	Al crear ubicaciones de activos se añaden correctamente.	Correcto
Ubicaciones	Online	Si un usuario del mismo grupo desde un terminal añade o modifica una ubicación, otro usuario del mismo grupo en otro terminal ve el resultado inmediatamente	Correcto
Ubicaciones	Offline	Se genera la tipología en el dispositivo en local correctamente	Correcto
Ubicaciones	Online/ Offline	Al generar una ubicación en modo offline, cuando este usuario accede nuevamente al dispositivo en modo online la ubicación se graba correctamente en la base de datos Firebase y esta es visible inmediatamente por otros usuarios del mismo grupo	Correcto
Ubicaciones	Online	Se comprueba la visibilidad por grupos. Una ubicación solo la puede ver un usuario del mismo grupo	Correcto
Crear activo	Online/ Offline	Los datos de selección se muestran correctamente según visibilidad de grupo de trabajo al crear o editar un activo	Correcto
Crear activo	Online	Creación de nuevo activo. Comprobación de solicitud de datos básicos	Correcto

Tipo	Modo	Prueba	Resultado
Crear/Editar activo	Online	Cuando un usuario de un grupo determinado crea o edita un activo, este se visualiza rápidamente en otro terminal de otro usuario del mismo grupo	Correcto
Crear/Editar activo	Offline	Creación de activo correcta	Correcto
Crear/Editar activo	Online/Offline	Al generar un activo en modo offline, cuando este usuario accede nuevamente al dispositivo en modo online el activo se graba correctamente en la base de datos Firebase y este es visible inmediatamente por otros usuarios del mismo grupo	Correcto
Documentación del activo	Offline	Se comprueba que en modo offline no se puede generar un documento para un activo ni acceder a un documento existente	Correcto
Documentación del activo	Online/Offline	Se comprueba que sin permisos no se puede generar un documento para un activo ni acceder a un documento existente	Correcto
Documentación del activo	Online	Se genera un documento con una imagen tomada con el dispositivo	Correcto
Documentación del activo	Online	Se genera un documento con un fichero seleccionado desde el dispositivo	Correcto
Documentación del activo	Online	Descarga y visualización de imagen asociada a un activo	Correcto
Documentación del activo	Online	Descarga y visualización de un fichero asociado a un activo	Correcto
Historial	Online/Offline	El historial del activo se genera correctamente para la creación del activo y los cambios posteriores de estado, ubicación y geoposicionamiento	Correcto
Historial	Online/Offline	El historial se sincroniza correctamente en los dispositivos del mismo grupo de trabajo al crearse desde otros, tanto en modo online, como en modo offline al pasar a modo online	Correcto
Filtro de activos	Online/Offline	Funcionamiento correcto del filtro de activos	Correcto
Filtro de activos	Online/Offline	Guardado del filtro dentro de la misma sesión	Correcto
Filtro de activos	Online/Offline	Limpieza del filtro mediante botón desde listado	Correcto
Filtro de activos	Online/Offline	Limpieza del filtro desde la edición del filtro	Correcto
NFC	Online/Offline	Filtro del listado al leer una etiqueta NFC	Correcto



Tipo	Modo	Prueba	Resultado
NFC	Online/ Offline	Ofrecer generar un nuevo activo al leer una etiqueta NFC no existente	Correcto
Escaner	Online/ Offline	Filtro del listado al leer un código de barras/ QR	Correcto
Escaner	Online/ Offline	Ofrecer generar un nuevo activo al leer una etiqueta de código de barras o QR no existente	Correcto
Orden de activos	Online/ Offline	Funcionamiento del filtro de ordenación, activando y desactivando opciones, y cambiando de ascendiente a descendiente	Correcto

## 5. Conclusiones

La lección más importante que he aprendido en el desarrollo de este trabajo ha sido la importancia de un buen análisis en cuanto a las necesidades de los usuarios a los que va dirigida la aplicación, y una buena planificación de las fases y su duración. Todo ello me ha servido como guía principal para poder desarrollar el producto en su totalidad y no perderme en el camino.

Otro de los aspectos claves ha sido la fase de diseño. Con tan poco tiempo para implementar una aplicación funcional completa, la fase de diseño ha sido muy importante. Muchas veces se empieza a implementar una aplicación, o simplemente una funcionalidad de una aplicación, sin tener una idea clara del resultado final que se pretende conseguir y esto suele llevar a implementar y re-implementar código con la consecuente pérdida de tiempo.

En este proyecto, al ir trabajando concienzudamente todas las fases, al iniciar la fase de implementación tenía una idea clara de lo que debía hacer. Cómo es lógico surgieron problemas técnicos, pero teniendo siempre en mente el objetivo, pude darles solución rápidamente. Sin embargo, he de decir que no he seguido la planificación en esta fase tal y como estaba planteada en el plan de trabajo. En cuanto inicié la implementación, me di cuenta que el orden de los trabajos a realizar planteado no era del todo correcto, más que correcto, no era el más eficaz, así que realicé ajustes. La implementación de las actividades que había planificado para una fase posterior, la inicié desde el primer momento, y en vez de ir secuencialmente punto a punto en algunos de los trabajos como lo tenía planificado, los fui haciendo en paralelo.

Como resultado final he podido llevar a cabo el proyecto cumpliendo con los objetivos planteados en un inicio en el tiempo requerido. Entiendo que llevar varios años programando software, aunque no se trate específicamente de aplicaciones móviles, me ha permitido ajustar los objetivos a los tiempos de

desarrollo del proyecto. Aún así, han surgido problemas, pero los he podido solventar en el margen de tiempo previsto.

GESTA es un proyecto que podría escalar con relativa facilidad y que, en un futuro, podría ampliar añadiendo funcionalidades como, por ejemplo, crear perfiles de usuarios y permitir al administrador de cada grupo eliminar datos generados. Este es un proceso muy complejo que requiere tiempo de diseño y de desarrollo, motivo por el cual no se ha incluido inicialmente. No obstante, la funcionalidad actual de GESTA permite su uso de forma satisfactoria en la mayoría de los casos.

## 7. Bibliografía

[1] - Fractal - Gestión de Activos y Mantenimiento - Apps en Google Play. [Consulta Febrero 2020] <[https://play.google.com/store/apps/details?id=com.fracttalmobile.app&hl=es\\_419](https://play.google.com/store/apps/details?id=com.fracttalmobile.app&hl=es_419)>

[2] - Gestión de Activos. [Consulta Febrero 2020] <<https://play.google.com/store/apps/details?id=com.gestionactivos.isbancantabria.gestionactivos>>

[3] - Gestión de Activos IT. [Consulta Febrero 2020] <<https://play.google.com/store/apps/details?id=com.ginstr.ItAssetManager>>

[4] - Aplicaciones para Android de NonZeroApps en Google Play. [Consulta Febrero 2020] <<https://play.google.com/store/apps/dev?id=7120341127614413624>>

[5] - Gestión de recursos móviles y gestión de activos. [Consulta Febrero 2020] <<https://www.kyroslbs.com/gestion-de-recursos-y-localizacion-activos-conectados-a-kyros/>>.

[6] - Mobile operating system market share - Worldwide- Statcounter global stats. [Consulta: Febrero 2020] <<https://gs.statcounter.com/os-market-share/mobile/worldwide>>

[7] - Desarrollo en cascada - Wikipedia, la enciclopedia libre [Consulta Marzo 2020] <[https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)>

[8] - Fotos gratis en Pixabay - [Consulta Marzo 2020] <<https://pixabay.com/es/photos/negocio-lady-mujer-chica-ordenador-3560935/>>

[9] - Fotos gratis en Pixabay - [Consulta Marzo 2020] <<https://pixabay.com/es/photos/negocio-empresario-silla-equipo-1839191/>>

[10] - Fotos gratis en Pixabay - [Consulta Marzo 2020] - <<https://pixabay.com/es/photos/mujer-negro-mujer-de-negocios-young-868522/>>