



Escuela de Ingeniería Informática de Valladolid

TRABAJO FIN DE GRADO

Grado en Ingeniería informática

Mención de Ingeniería de Software

Showroom - Web de descarga y visualización de aplicaciones

Autor: David Martín Terán

Tutor: Joaquín Adiego Rodríguez

Agradecimientos

Quería agradecer en primer lugar a mis padres y mi novia por ayudarme en todo lo que han podido durante estos años, querer saber cómo iban los avances con la carrera y proyecto y creer en mí sin crear tensiones o presiones adicionales.

También agradezco a mis amigos que siempre están ahí y a toda la gente que se ha preocupado en distintas maneras por mí a lo largo de este periodo.

Por último, me gustaría mencionar a los profesores de la carrera de Ingeniería Informática con los cuales he aprendido valiosos conceptos para el futuro y en especial a Joaquín por acceder a la tutoría de este proyecto.

Tabla de contenido

1	Introducción	8
1.1	Motivación	8
1.1.1	Objetivos	8
1.2	Conocimientos previos.....	8
1.2.1	Contenido de cada aplicación	8
1.2.2	Nombre y explicación.....	9
1.2.3	Organización del proyecto	9
2	Contexto.....	10
2.1.1	Conocimiento de las herramientas utilizadas	10
2.2	Conocimiento de las herramientas utilizadas	13
2.2.1	Google Play.....	13
2.2.2	Microsoft Store.....	14
2.2.3	Apple Store.....	15
2.2.4	Aptoide.....	16
2.2.5	Amazon AppStore.....	17
3	Desarrollo del proyecto.....	18
3.1	Especificaciones iniciales.....	18
3.2	Proceso de desarrollo.....	18
3.3	Roles en el proyecto.....	20
3.4	Gestión de riesgos.....	20
3.5	Seguimiento del proyecto	24
3.5.1	Fase de Inicio	25
3.5.2	Fase de elaboración.....	25
3.5.3	Fase de construcción	25
3.5.4	Fase de transición.....	26
3.6	Etapas del desarrollo	26
4	Análisis.....	26
4.1	Requisitos	26
4.1.1	Requisitos funcionales.....	27
4.1.2	Requisitos no funcionales	28
4.1.3	Requisitos de información.....	29
4.2	Modelo de casos de uso	29
4.3	Especificaciones de caso de uso.....	29
4.4	Modelo de dominio	37
4.5	Diagramas de secuencia.....	38

5	Diseño.....	45
5.1	Arquitectura	45
5.2	Diagrama de diseño.....	48
5.3	Diseño de la base de datos.....	50
5.4	Diccionario de la base de datos.....	50
6	Diseño de interfaz	53
6.1	Diseño centrado en usuario	53
6.2	Análisis de usuarios	53
6.3	Tareas de usuario	54
6.4	Bocetaje.....	54
6.5	Evaluación de la interfaz	64
7	Implementación	65
8	Pruebas.....	67
9	Despliegue.....	74
10	Conclusiones y posibles mejoras.....	75
10.1	Conclusiones.....	75
10.2	Posibles mejoras.....	75
11	Bibliografía	76
12	ANEXOS	80
12.1	Manual de usuario	80
12.2	Manual de usuario	82
12.3	Manual de administrador.....	84
12.4	Manual de súper administrador.....	86

Lista de figuras

Figura 1: Angular Framework.....	11
Figura 2: Firebase Database	12
Figura 3: Astah UML.....	12
Figura 4: Google Play.....	13
Figura 5: Microsoft Store.....	14
Figura 6: Apple Store.....	15
Figura 7: Aptoide	16
Figura 8: Amazon AppStore.....	17
Figura 9: Gráfico de flujos de trabajo sobre iteraciones.....	19
Figura 10: Modelo de casos de uso.....	29
Figura 11: Modelo de dominio.....	37
Figura 12: Diagrama de secuencia de CU001.....	38
Figura 13: Diagrama de secuencia de CU002.....	38
Figura 14: Diagrama de secuencia de CU003.....	39
Figura 15: Diagrama de secuencia de CU004.....	39
Figura 16: Diagrama de secuencia de CU006.....	40
Figura 17: Diagrama de secuencia de CU005.....	39
Figura 18: Diagrama de secuencia de CU007.....	40
Figura 19: Diagrama de secuencia de CU008.....	41
Figura 20: Diagrama de secuencia de CU009.....	42
Figura 21: Diagrama de secuencia de CU010.....	42
Figura 22: Diagrama de secuencia de CU011.....	43
Figura 23: Diagrama de secuencia de CU012.....	43
Figura 24: Diagrama de secuencia de CU013.....	44
Figura 25: Diagrama de secuencia de CU014.....	44
Figura 26: Diagrama de secuencia de CU015.....	44
Figura 27: Diagrama de secuencia de CU016.....	45
Figura 28: Ejemplos de módulos Angular.....	46
Figura 29: Ejemplo de componente Angular.....	46
Figura 30: Ejemplo de servicios Angular	47
Figura 31: Ejemplo de routing Angular	47
Figura 32: Diagrama de clases de la base de datos.....	50
Figura 33: Bocetaje del flujo de la aplicación.....	54
Figura 34: Boceto de la pantalla de inicio de sesión	55
Figura 35: Boceto de pantalla de registro de usuario	55
Figura 36: Boceto de pantalla Home de usuario.....	56
Figura 37: Boceto de pantalla de detalle de aplicación de usuario	57
Figura 38: Boceto de pantalla de Home de súper administrador	57
Figura 39: Boceto de pantalla de detalle de aplicación de administrador.....	58
Figura 40: Boceto de pantalla de selección de súper administrador	59
Figura 41: Boceto de pantalla de añadir administrador	59
Figura 42: Boceto de pantalla de edición de grupos.....	60
Figura 43: Boceto de pantalla de edición de categorías	60
Figura 44: Boceto móvil inicio de sesión.....	61
Figura 45: Boceto móvil registro	61

Figura 46: Boceto móvil home usuario	62
Figura 47: Boceto móvil home administrador	62
Figura 48: Boceto móvil detalle app usuario.....	62
Figura 49: Boceto móvil detalle app administrador.....	62
Figura 50: Boceto móvil selección súper administrador	63
Figura 51: Boceto móvil añadir administrador.....	63
Figura 52: Boceto móvil de edición de grupos.....	63
Figura 53: Boceto móvil edición de categorías	63
Figura 54: MU – Pantalla de inicio de sesión	80
Figura 55: MU – Pantalla de registro de usuario.....	81
Figura 56: MU - Pantalla home de usuario.....	82
Figura 57: MU - Pantalla de detalla de aplicación de usuario.....	82
Figura 58: MU - Pantalla de detalle de imagen.....	83
Figura 59: MU - Pantalla de comentario	84
Figura 60: MU - Pantalla home de administrador.....	84
Figura 61: MU - Pantalla de detalle de aplicación de administrador.....	85
Figura 62: MU - Pantalla de selección de súper administrador	86
Figura 63: MU - Pantalla home de súper administrador.....	86
Figura 64: MU - Pantalla de detalle de aplicación de súper administrador.....	87
Figura 65: MU - Pantalla de añadir administradores	87
Figura 66: MU - Pantalla de edición de grupos	87
Figura 67: MU - Pantalla de edición de categorías.....	87

Lista de tablas

Tabla 1: Características destacables de Google Play.....	13
Tabla 2: Características destacables de Microsoft Store	14
Tabla 3: Características destacables de Apple Store.....	15
Tabla 4: Características destacables de Aptoide.....	16
Tabla 5: Características destacables de Amazon AppStore	17
Tabla 6: Roles del proyecto	20
Tabla 7: Colores asignados al riesgo según impacto y probabilidad.....	21
Tabla 8: Riesgo-001	22
Tabla 9: Riesgo-002	22
Tabla 10: Riesgo-003	22
Tabla 11: Riesgo-004.....	23
Tabla 12: Riesgo-005	23
Tabla 13: Riesgo-006	23
Tabla 14: Riesgo-007	24
Tabla 15: Riesgo-008	24
Tabla 16: Fases del proyecto con fechas estimadas y reales	25
Tabla 17: Requisitos funcionales.....	28
Tabla 18: Requisitos no funcionales.....	28
Tabla 19: Requisitos de información.....	29
Tabla 20: CU-001	30

Tabla 21: CU-002	31
Tabla 22: CU-003	31
Tabla 23: CU-004	32
Tabla 24: CU-005	32
Tabla 25: CU-006	32
Tabla 26: CU-007	33
Tabla 27: CU-008	33
Tabla 28: CU-009	34
Tabla 29: CU-010	34
Tabla 30: CU-011	34
Tabla 31: CU-012	35
Tabla 32: CU-013	35
Tabla 33: CU-014	35
Tabla 34: CU-015	36
Tabla 35: CU-016	36
Tabla 36: Contenido de tabla Application en la BD.....	51
Tabla 37: Contenido de tabla User en la BD.....	51
Tabla 38: Contenido de tabla Category en la BD	51
Tabla 39: Contenido de la tabla Comment en la BD	51
Tabla 40: Contenido de la tabla ImagesByApp en la BD	52
Tabla 41: Contenido de la tabla Company en la BD	52
Tabla 42: PCN-001	67
Tabla 43: PCN-002	67
Tabla 44: PCN-003	68
Tabla 45: PCN-004	69
Tabla 46: PCN-005	69
Tabla 47: PCN-006	70
Tabla 48: PCN-007	70
Tabla 49: PCN-008	70
Tabla 50: PCN-009	71
Tabla 51: PCN-010	71
Tabla 52: PCN-011	72
Tabla 53: PCN-012	73
Tabla 54: PCN-013	73

1 Introducción

1.1 Motivación

Con la ejecución de esta aplicación web se busca añadir una herramienta para poder visualizar de forma conjunta las distintas aplicaciones que han realizado los alumnos en la asignatura de Sistemas Móviles de la Ingeniería Informática de Valladolid. En este proyecto únicamente estarán aquellas aplicaciones pertenecientes a los últimos 2 años lectivos.

Además de esta visualización también contará con una herramienta para poder añadir más aplicaciones para proyectos de años futuros.

De esta forma se busca una visualización rápida y cómoda de cualquier usuario a dichas aplicaciones utilizando interfaces similares a algunas de las más extendidas como son Play Store o Microsoft Store.

1.1.1 Objetivos

En primer lugar, se fijarán una serie de objetivos los cuales marcarán la construcción y realización del proyecto:

- Poner en práctica los métodos de exploración e investigación para el conocimiento de las herramientas de trabajo.
- Administrar la toma de decisiones en las diferentes fases de vida del proyecto.
- Conocer las características y prioridades de la muestra de información de las aplicaciones.
- Construir una aplicación web desde la fase de bocetaje hasta su publicación.

1.2 Conocimientos previos

Una vez expuestos estos objetivos se van a detallar algunos puntos importantes del desarrollo de la aplicación como el contenido expuesto de las aplicaciones, las nociones anteriores a la práctica sobre las tecnologías utilizadas y la explicación del nombre.

1.2.1 Contenido de cada aplicación

Para cada aplicación se van a tener en cuenta una serie de puntos que van a aportar información sobre su funcionamiento.

- Nombre de la aplicación
- Logo de la aplicación

- Descripción de su funcionamiento a grandes rasgos
- Imágenes de uso de la aplicación
- Subtítulo genérico para asociarla a un tipo (Productividad, Entretenimiento, Finanzas...)
- Enlace que redirigirá al punto de descarga de la aplicación.

1.2.2 Nombre y explicación

La aplicación se llama Showroom puesto que es el nombre derivado del término relativo a su significado. Un showroom es un espacio abierto donde diseñadores y fabricantes de moda exponen sus nuevas colecciones a los posibles compradores. Debido a que la naturaleza del proyecto tenía una estrategia similar en cuanto a la muestra de las nuevas aplicaciones hechas por los alumnos se ha escogido este nombre.

1.2.3 Organización del proyecto

En primer lugar, vamos a repasar de manera breve los capítulos los diferentes puntos de los que constará el trabajo de modo que se facilite la lectura y entendimiento.

- **Introducción:** Apartado inicial con el que se expondrán las bases del proyecto, así como las tecnologías, objetivos y origen de este.
- **Contexto:** Se expondrán varias herramientas o aplicaciones de fin semejante con el fin de resaltar aquellas características que las hacen únicas y así entender cuál es el camino del diseño del proyecto. También se mencionarán las tecnologías que servirán de apoyo para el proyecto y de mi experiencia previa con cada una de ellas.
- **Desarrollo del proyecto:** Explicación de los requisitos iniciales propuestos, de la propuesta de desarrollo posterior del proyecto y exposición de este a los diferentes tipos de riesgos.
- **Análisis:** Requisitos funcionales, no funcionales y de información del proyecto, casos de uso y sus especificaciones, modelo de dominio y los distintos diagramas de secuencia asociados a cada uno de los casos de uso mencionados.
- **Diseño:** Explicación de la arquitectura angular y de los puntos que se han intentado llevar a cabo en el proyecto, diseño de los componentes Angular escogidos y de las clases de la base de datos.
- **Diseño de interfaz:** Diseño y bocetaje de la interfaz de la web, análisis de los usuarios objetivo y retroalimentación sobre el bocetaje escogido.
- **Implementación:** Explicación del proceso de puesta a punto de la web hasta su despliegue final.
- **Pruebas:** Exposición de las pruebas finales para verificar la calidad del producto.
- **Despliegue:** Proceso de despliegue de la aplicación.
- **Manual:** Pequeña guía de funcionamiento para cada una de las interfaces del sistema.
- **Conclusiones:** Resumen de las conclusiones extraídas y algunas mejoras futuras posibles al proyecto.
- **Bibliografía**

2 Contexto

Para la realización de la aplicación se ha observado el comportamiento de aplicaciones que tienen una funcionalidad u objetivos similares. En la actualidad existen muchos tipos de aplicaciones las cuales sirven para mostrar información de otras aplicaciones y permitir su descarga. Si bien es cierto que este proyecto carece de los conocimientos y tiempo necesarios para hacer algo que se asemeje a estas aplicaciones sí se pueden extraer ciertas características que ayudarían en su desarrollo. Las aplicaciones móviles han ido en aumento desde los inicios de los teléfonos inteligentes y se han hecho necesarias este tipo de aplicaciones para tener un punto global de acceso a muchas de ellas. También se hará un repaso de conocimientos previos sobre las herramientas que se usarán en el proyecto para poner un punto de partida.

2.1.1 Conocimiento de las herramientas utilizadas

Para complementar la información del conocimiento previo de las herramientas se expondrá de forma breve su funcionamiento y posteriormente se expondrá la toma de contacto que se ha tenido en épocas anteriores.

Angular: Es un *framework* de JavaScript de código abierto utilizado para la creación de aplicaciones web apoyado en el lenguaje TypeScript. Es mantenido por Google y tiene ciertas características que sirven de ayuda a la hora de empezar a construir aplicaciones o webs:

- Un framework basado en componentes para la construcción de aplicaciones web escalables.
- Una colección de librerías bien integradas que cubren gran cantidad de características como enrutamiento, manejo de formularios, facilitan la comunicación cliente-servidor, y más.
- Un gran set de herramientas de desarrollo que ayudan al usuario a desarrollar, construir, probar y actualizar su código.

Angular tiene la ventaja de ser apto tanto para aplicaciones de nivel empresarial como aquellas hechas por un solo desarrollador. Tiene actualizaciones que siguen la línea del producto para que acomodarse a nuevas versiones no se haga tedioso y cuenta con una gran comunidad de desarrolladores, librerías y creadores de contenido que hacen que el desarrollo en el framework sea menos duro para aquellos que acaben de comenzar su etapa de desarrollador. Por estas razones son por las que comencé con la idea de utilizar este framework ya que quería probar a hacer un proyecto relativamente grande para lo que yo estaba acostumbrado con una herramienta que puede que me ayude en un posible futuro laboral.

Mis conocimientos sobre Angular eran bastante limitados en un inicio. Solamente había probado el framework en una asignatura del último año durante un mes así que partía de una base poco sólida. Tenía conocimiento de la comunicación entre distintos componentes y de los templates de estos, pero carecía de información sobre otras muchas temáticas. Con ayuda de la documentación empecé a comprender el funcionamiento y a lo largo del desarrollo he ido entendiendo más sobre el uso de cada una de las herramientas que este framework ofrece.

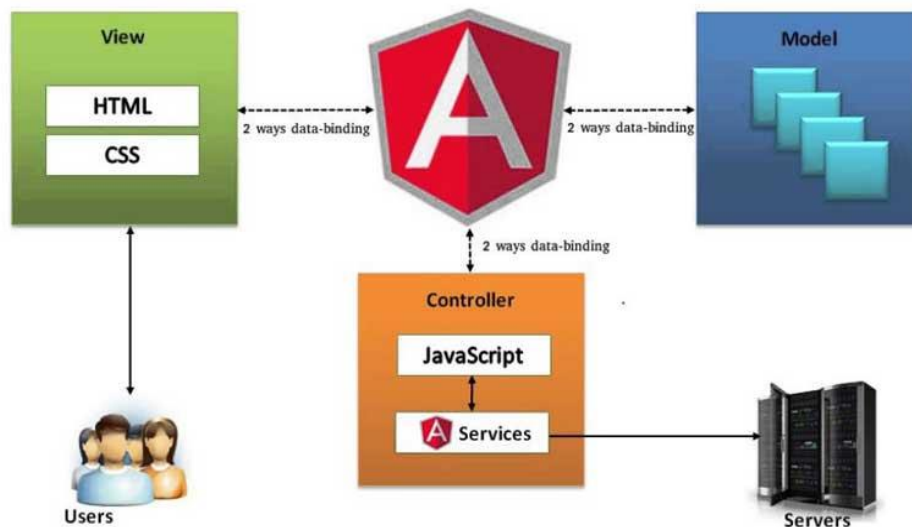


Figura 1: Angular Framework

Firebase: Plataforma usada para el desarrollo de aplicaciones web ubicada en la nube integrada con Google Cloud Platform. Fue creada en 2011 y adquirida por Google posteriormente en 2014. Usa un conjunto de herramientas para la creación y sincronización de proyectos en la nube. Estos proyectos serán dotados de escalabilidad en el número de usuarios y también se hará posible una monetización de estos.

Esta plataforma ofrece una serie de ventajas que hace que resulte atractivo su uso para los desarrolladores:

- Sincronizar de forma sencilla los datos de sus proyectos sin tener que administrar conexiones o lógica compleja.
- Ofrece herramientas multiplataforma que permiten su integración tanto para plataformas web como para aplicaciones móviles. Compatible con grandes plataformas como IOS, Android, aplicaciones web, Unity y C++.
- Usa la infraestructura de Google y escala de manera rápida y sencilla cualquier tipo de aplicación.
- Creación de proyectos sin necesidad de un servidor. Las herramientas se incluyen en los SDK para los dispositivos móviles y web.

Aparte de estas ventajas también hay que tener en cuenta la documentación extensa y la gran comunidad de desarrolladores que pueden ayudar a resolver cualquier problema mediante foros. Además, cuenta con más características en forma de servicios que encajaban en este proyecto:

- **Firebase Auth:** Servicio de autenticación de usuarios usando código cliente. Cuenta con varios proveedores como Facebook, GitHub, Twitter, Google, Yahoo y Microsoft así como los usuales con correo y contraseña. Este servicio proporciona de forma rápida y concreta lo que el proyecto necesitaba en el registro y autenticación de los usuarios.

- Firebase Storage: Servicio de carga y descargas de archivos. Permitirá tener centralizadas y bien distribuidas las imágenes de la aplicación.
- Firebase Cloud Firestore: Servicio de almacenamiento de datos. Bas de datos NoSQL en la que los datos toman forma de colecciones en las que se agruparán los documentos que son cada una de las entradas en la base de datos. Si bien es cierto que tiene limitaciones como la búsqueda por nombre o como múltiples condiciones en la búsqueda como por ejemplo se puede en otros lenguajes SQL para este proyecto se acomodó la base de datos para que fuera totalmente funcional.

Sobre esta herramienta mis conocimientos eran prácticamente nulos. Si bien es cierto que me había informado de las diferencias entre una base de datos SQL y una NoSQL nunca había trabajado con una. Firebase tiene un ecosistema propio que fue difícil de entender al principio, pero gracias a la búsqueda en documentación y foros fui capaz de resolver cada una de las dudas que se me plantearon durante el desarrollo del proyecto.

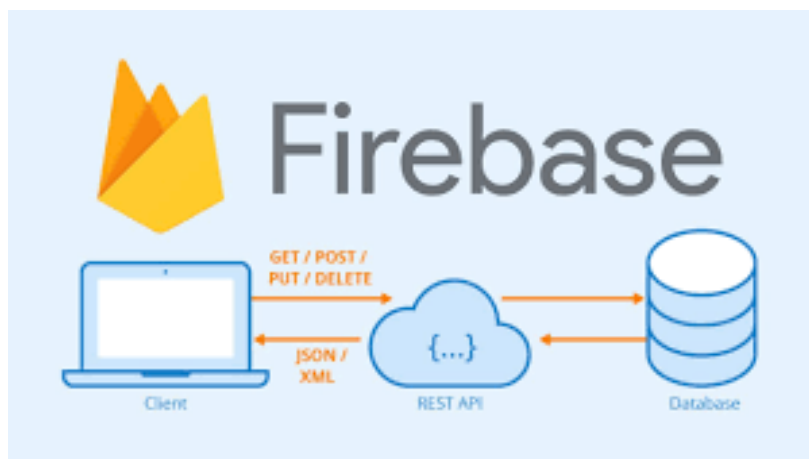


Figura 2: Firebase Database

Astah: Herramienta de modelado UML (Lenguaje Unificado de Modelado) y mapas conceptuales. A la hora de desarrollar un proyecto como este se necesita una herramienta que proporcione las características necesarias para crear los diferentes diagramas que constituirán el modelado de la aplicación web. Debido a la experiencia con este programa en los últimos años de carrera se optó por ella ya que no supondría mucho tiempo de aclimatación y produce resultados óptimos. Si bien es cierto que pueden hacerse múltiple cantidad de diagramas el uso de esta aplicación se centrará en hacer diagramas de clase, base de datos, casos de uso y de secuencia.

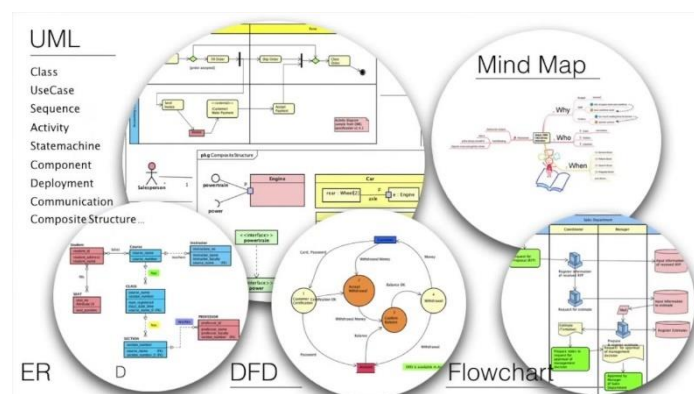


Figura 3: Astah UML

2.2 Conocimiento de las herramientas utilizadas

2.2.1 Google Play

Sin duda la plataforma de distribución digital más globalizada en la descarga de aplicaciones. Ofrece aplicaciones para sistema operativo Android, así como una tienda en línea operada por Google. Permite a los usuarios adquirir juegos, música, libros, revistas y películas además de las propias aplicaciones. Viene prácticamente preinstalada en todos los dispositivos Android y alberga más de 3 millones de aplicaciones.

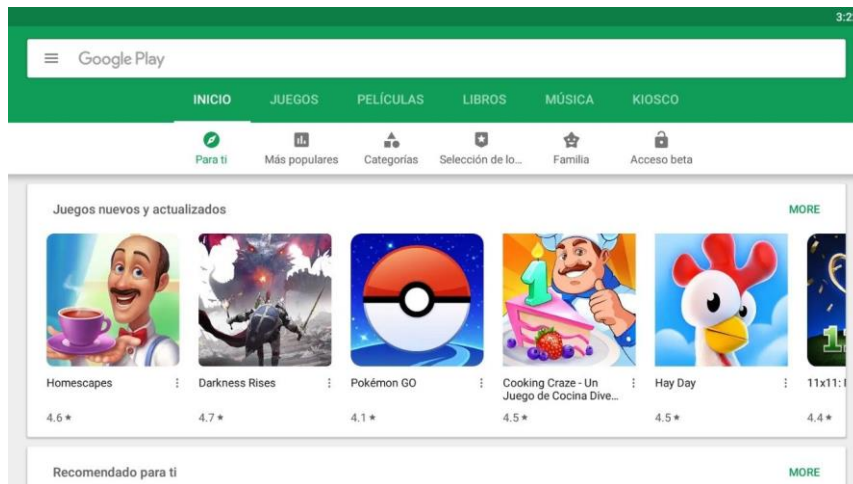


Figura 4: Google Play

Características destacables
Distribución tanto de aplicaciones, juegos, películas y libros.
Separación por categorías (mejor valorados, juegos casuales...).
Sugerencias personalizadas según tus descargas.
Poder de búsqueda por nombre y categorías.
Comentarios interactivos y visuales.
Estadísticas de aplicación fácilmente visibles.
Imágenes de aplicaciones interactivas.

Tabla 1: Características destacables de Google Play

2.2.2 Microsoft Store

Plataforma de distribución digital creada por Microsoft Corporation. Presente a partir del software Windows 8 provee de un catálogo de aplicaciones centralizado y seguro para este software. Al igual que en el caso de Google Play también trae consigo juegos, música, películas y demás añadidos a las propias aplicaciones.

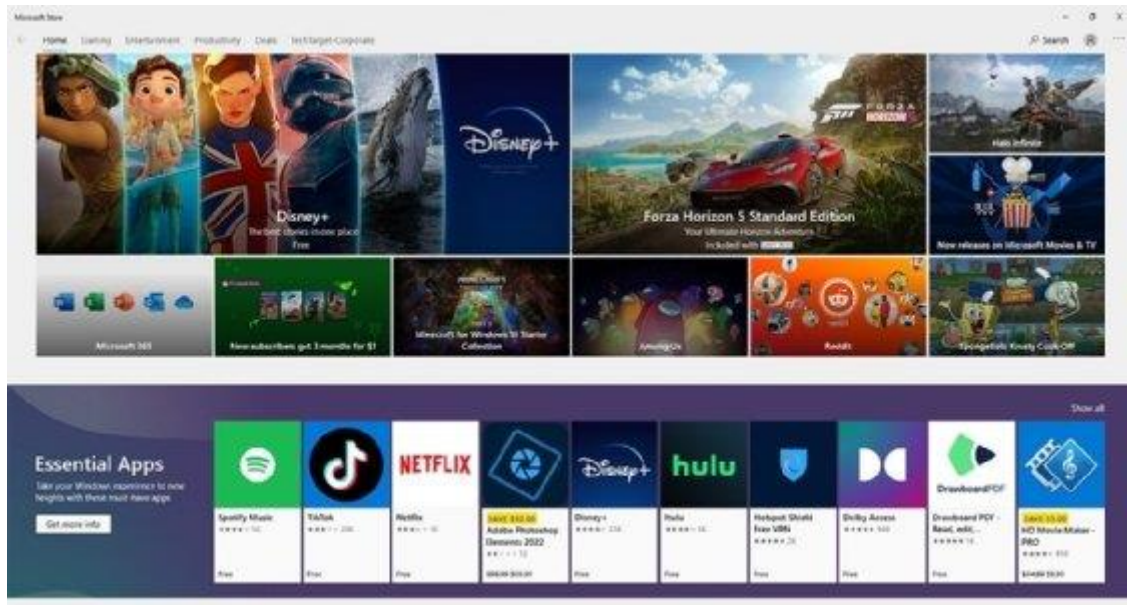


Figura 5: Microsoft Store

Características destacables
Distribución tanto de aplicaciones, juegos, películas y series de TV.
Separación por categorías (mejor valorados, juegos casuales...).
Rápido acceso a diferentes colecciones de aplicaciones similares.
Poder de búsqueda por nombre y categorías.
Comentarios interactivos y visuales.
Estadísticas de aplicación fácilmente visibles.
Imágenes de aplicaciones interactivas.
Carrusel de imágenes de aplicaciones o películas cambiante que activa el colorido y sensación interactiva del usuario.

Tabla 2: Características destacables de Microsoft Store

2.2.3 Apple Store

Servicio de búsqueda y descarga de aplicaciones para usuarios de iPhone, iPod Touch y iPad creado por la compañía Apple Inc. Las aplicaciones presentes son todas desarrolladas en el iOS SDK y publicadas por apple.



Figura 6: Apple Store

Características destacables
Distribución tanto de aplicaciones y juegos.
Separación por categorías (mejor valorados, juegos casuales...).
Rápido acceso a diferentes colecciones de aplicaciones similares.
Poder de búsqueda por nombre y categorías.
Comentarios interactivos y visuales.
Estadísticas de aplicación fácilmente visibles.
Imágenes de aplicaciones interactivas.
Exclusividad en muchas aplicaciones.
Prueba gratuita de ciertas aplicaciones de pago.

Tabla 3: Características destacables de Apple Store

2.2.4 Aptoide

Plataforma de descarga de aplicaciones que en los últimos años se ha convertido en uno de los repositorios con más aplicaciones Android del mercado. Aptoide no es un almacén único y centralizado ya que cada usuario administra su propia selección y almacén. La comunicación se basa en un protocolo abierto basado en XML y su gestión se parece a la de APT (administrador de paquetes de Debian) donde se trabajan con múltiples repositorios. Cuando un usuario requiere de un software utiliza el cliente para buscar fuentes donde se almacena la aplicación.

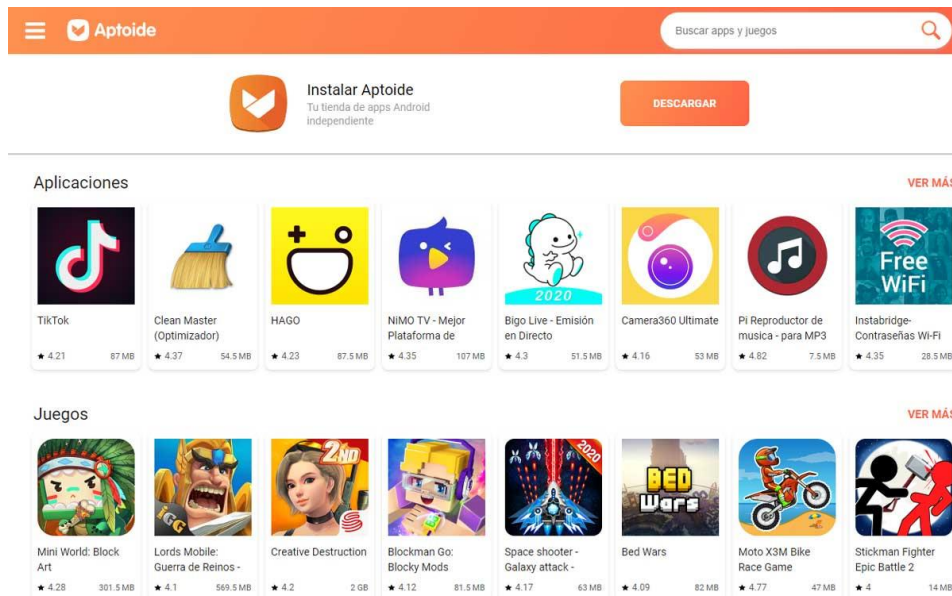


Figura 7: Aptoide

Características destacables
Distribución tanto de aplicaciones y juegos.
Separación por categorías (mejor valorados, juegos casuales...).
Poder de búsqueda por nombre y categorías.
Comentarios interactivos y visuales.
Aplicaciones que no se encuentran en Play Store para Android.
Imágenes de aplicaciones interactivas.
Sistema de donaciones y monedas propio.
Rebaja en aplicaciones y juegos de pago.
Sistema de tiendas para exponer a creadores.

Tabla 4: Características destacables de Aptoide

2.2.5 Amazon AppStore

Tienda de aplicaciones y juegos Android. En su inicio fue creada para dispositivos Fire de la marca, pero es accesible desde cualquier dispositivo Android. Su adición a Microsoft Store en el 2021 hizo que tuviera un número mayor de usuarios siendo posible su redirección desde la Store de Microsoft para descargar aplicaciones Android.

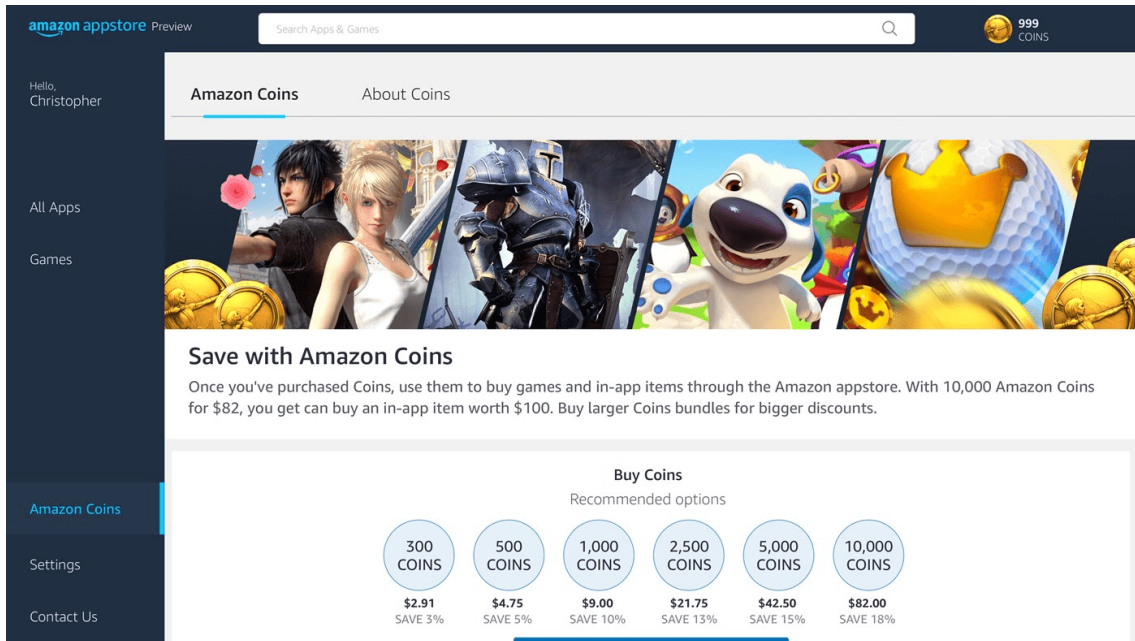


Figura 8: Amazon AppStore

Características destacables
Distribución tanto de aplicaciones y juegos.
Poder de búsqueda por nombre y categorías.
Comentarios interactivos y visuales.
Imágenes de aplicaciones interactivas.
Sistema monedas propio para adquirir productos.
Ofertas en aplicaciones y precios.
Aplicaciones y juegos son desinstalados al desinstalar la aplicación.

Tabla 5: Características destacables de Amazon AppStore

3 Desarrollo del proyecto

3.1 Especificaciones iniciales

Para un correcto desarrollo de un proyecto hay que delimitar inicialmente unas especificaciones iniciales que guiarán todas las decisiones posteriores sobre el mismo. La idea de proyecto fue proporcionada por mi tutor Joaquín tras deliberar algunas propuestas propias que quizás excedían o no ajustaban a lo necesario para un proyecto de fin de carrera. La propuesta inicial era la de construir una aplicación web responsiva (tanto para dispositivos móviles como ordenadores) que sirviese como punto intermedio de acceso y edición a las diferentes aplicaciones creadas por los alumnos de la escuela, particularmente en la asignatura de Sistemas Móviles. La idea inicial solo contaba de ese prerrequisito así que la parte de diseño y desarrollo de la aplicación estaba a mi libre disposición.

3.2 Proceso de desarrollo

Para el progreso en el desarrollo de la aplicación se ha utilizado el llamado Proceso Racional Unificado. Este es un proceso propiedad de IBM que consiste en una metodología estándar utilizada para el análisis, diseño, implementación y documentación con la ayuda del Lenguaje Unificado de Modelado (UML).

Las características principales del RUP asociadas al proyecto son:

- **Interacción continua con el cliente:** En mi caso como el cliente va a ser el alumnado de la escuela seleccioné una muestra pequeña de alumnos para poder ir probando la aplicación web ya que me resultaba difícil y costoso probarlo con una muestra más grande. Coincidió que yo mismo y compañeros de clase somos parte del target de la web así que por esa parte resultó sencillo.
- **Valor iterativo:** A lo largo del proyecto se elaboraron una serie de etapas de desarrollo que en su mayoría comprendían el diseño y desarrollo de cada una de las páginas principales, ya que, si bien es cierto que tienen cierta conexión se podía hacer una clara diferencia entre sus funcionalidades.
- **Enfocarse en la calidad:** En lugar de aplicar un esqueleto e ir mejorándolo y añadiendo detalles se enfocó de manera iterativa e incremental, siempre queriendo obtener la mayor calidad posible antes de pasar a la siguiente iteración. De esta forma primero se mejoraron detalles particulares, si bien no hasta nivel de producción, sí hasta un nivel avanzado quedando mínimos retoques y con toda la funcionalidad.
- **Equilibrio de prioridades:** Inicialmente se detallaron ciertos aspectos de la aplicación que a la hora del desarrollo tenían que ser cumplidos evitando de esta manera conflictos en la idea del proyecto.

A parte de las características de un proyecto guiado por la metodología RUP hay que destacar el modelo de ciclo de vida que acompaña a esta idea. Comprende el trabajo en secuencias semi ordenadas y las tareas en fases e iteraciones. Existen 4 fases:

- **Fase de inicio:** Consta de iteraciones que hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

- **Fase de elaboración:** Iteraciones orientadas al desarrollo de la arquitectura, flujos de requisitos, modelo de negocio, análisis y diseño y parte de implementación de la arquitectura.
- **Fase de construcción:** Construcción del producto con el diseño elaborado previamente. Van sucediéndose iteraciones que modelan los casos de uso definidos.
- **Fase transición:** Fase final que sirve para probar y garantizar que el producto está preparado para la entrega al usuario.

El siguiente diagrama dibuja cómo varía el esfuerzo en cada una de las actividades por cada iteración sucesiva.

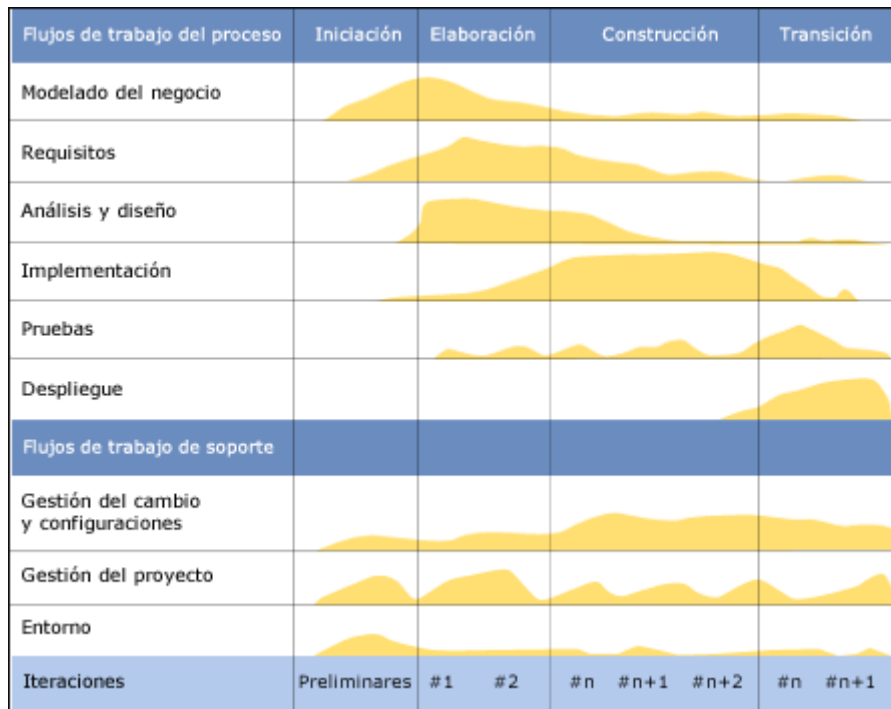


Figura 9: Gráfico de flujos de trabajo sobre iteraciones

3.3 Roles en el proyecto

En este apartado van a ser expuestos los roles de las personas que efectuaron este proyecto.

Miembro	Rol	Responsabilidad
Joaquín Adiego Rodríguez	Jefe de proyecto	Encargado de definir las metas del proyecto e interpretar sus necesidades para que se ejecute de manera adecuada.
David Martín Terán	Analista	Encargado de obtener los requisitos y elaborar un modelo de análisis y diseño teniendo en cuenta los objetivos marcados.
	Diseñador	Encargado del diseño del sistema tanto para la arquitectura del código como de la base de datos que busque dar soluciones a los requisitos obtenidos.
	Desarrollador	Encargado de implementar el código y las soluciones previamente diseñadas.
	Probador	Encargado de realizar las pruebas convenientes para comprobar que la elaboración de la aplicación no muestra fallos en su uso.

Tabla 6: Roles del proyecto

3.4 Gestión de riesgos

Antes de implementar cualquier desarrollo hay que tener en cuenta que puede que haya una serie de riesgos.

El riesgo es una medida de magnitud de los daños frente a una situación peligrosa. Es por esto por lo que cada situación que conlleve cierto riesgo sea tomada en cuenta, tanto su probabilidad de ocurrencia como sus consecuencias derivadas.

Los riesgos en un proyecto pueden tener dos naturalezas:

- **Riesgos conocidos:** Aquellos que son identificados y analizados. En un proyecto se intenta que sea el mayor número de los riesgos que lo abordan puesto que reconociendo un riesgo determinado es posible establecer un plan de acción y contingencia en caso de que termine sucediendo.
- **Riesgos desconocidos:** Como su nombre indica son aquellos riesgos que no han sido identificados en la fase de identificación de riesgos y no pueden ser tratados de forma proactiva.

Podemos clasificar los riesgos de varias maneras, pero en nuestro caso distinguiremos tres tipos:

- **Riesgos de proyecto:** aquellos relacionados con la planificación del proyecto, ya sea la parte de personal, requisitos, recursos...
- **Riesgos de producto:** son los relacionados con la falta de experiencia en el ámbito o con el desconocimiento de las tecnologías utilizadas. Hacen que la calidad del producto se vea afectada.
- **Riesgos de Proceso:** atienden a riesgos de diseño y estrategia, así como mala documentación.

Una vez visto los tipos vamos a diferenciar los riesgos por la probabilidad de aparición y por el impacto que producen en el proyecto.

Diferenciaremos el impacto generado en el proyecto como:

- Bajo
- Moderado
- Grave
- Extremo

Las probabilidades serán medidas en porcentaje teniendo esta jerarquía:

- Baja (<25%)
- Media (25%-50%)
- Alta (50%-75%)
- Muy alta (75%-100%)

		IMPACTO			
		Bajo	Moderado	Grave	Extremo
PROBABILIDAD	Baja				
	Media				
	Alta				
	Muy Alta				

Tabla 7: Colores asignados al riesgo según impacto y probabilidad

Una vez expuestos los diferentes tipos de impacto y probabilidad se expondrán ciertos riesgos elaborando un plan de contingencia para reducir el posible impacto de estos sobre el proyecto:

Riesgo-001		Escasez de conocimientos sobre las herramientas			
Descripción		Escasa experiencia con el Framework Angular y con la administración de bases de datos vía Firebase.			
Categoría		Producto			
Probabilidad	Alta	Impacto	Grave	Exposición	
Consecuencias		Puede suponer retraso en el desarrollo del proyecto. Debido a que las fechas de planificación y entrega no son críticas no supone un impacto mayor.			
Plan de contingencia		Repasar conceptos de desarrollo en las herramientas utilizadas y consultar a personas con mayor experiencia.			

Tabla 8: Riesgo-001

Riesgo-002		No disponibilidad de personal			
Descripción		El desarrollador sufre una baja temporal debido a motivos personales o de salud.			
Categoría		Proyecto			
Probabilidad	Baja	Impacto	Extremo	Exposición	
Consecuencias		No se cumplen los plazos del trabajo y, dependiendo de la gravedad de la situación puede peligrar la entrega final de este.			
Plan de contingencia		Reorganización del plan de trabajo y objetivos para ajustarse a los nuevos plazos.			

Tabla 9: Riesgo-002

Riesgo-003		No disponibilidad de equipo			
Descripción		El desarrollador tiene un problema con su equipo de trabajo.			
Categoría		Proyecto			
Probabilidad	Media	Impacto	Moderado	Exposición	
Consecuencias		Conllevaría una pérdida del trabajo instantáneo no guardado, el cual no debería ser mucho y tiempo para la búsqueda y puesta a punto de otro equipo, así como de arreglo del primero.			
Plan de contingencia		Realizar guardados y subidas a la nube continuas para si se da el caso poder proseguir desde otro equipo.			

Tabla 10: Riesgo-003

Riesgo-004		Modificación de requisitos iniciales			
Descripción		El proyecto sufre un cambio de requisitos en una parte tardía del proceso de elaboración.			
Categoría		Proceso			
Probabilidad	Baja	Impacto	Extremo	Exposición	
Consecuencias		Cambios en la estructura del proyecto y revisión de la elaboración de este.			
Plan de contingencia		Partir con los objetivos claros desde un inicio y atenuar el problema lo antes posible. Sería necesaria una reunión con el tutor del proyecto para reenfocar los pasos esenciales posteriores.			

Tabla 11: Riesgo-004

Riesgo-005		Fallo en diseño			
Descripción		El diseño expuesto puede no estar acorde con las especificaciones iniciales del trabajo.			
Categoría		Proceso			
Probabilidad	Media	Impacto	Grave	Exposición	
Consecuencias		Reconstrucción del diseño del producto.			
Plan de contingencia		Dejar claros cuáles tienen que ser los requisitos en el diseño del proyecto y aclarar posibles dudas con revisiones parciales del producto.			

Tabla 12: Riesgo-005

Riesgo-006		Falta de experiencia en gestión de proyectos			
Descripción		Debido a que nunca se ha llevado la gestión de un proyecto de esta magnitud, la planificación puede presentar un factor de riesgo.			
Categoría		Producto			
Probabilidad	Media	Impacto	Grave	Exposición	
Consecuencias		Fallos en la planificación y consecución de hitos. En un caso extremo puede dañar la entrega final del producto.			
Plan de contingencia		Revisar proyectos con longitudes y temas similares para dar estimaciones reales de los hitos.			

Tabla 13: Riesgo-006

Riesgo-007		Mala interpretación de requisitos			
Descripción		Una vez listados los requisitos se realiza un desarrollo incorrecto de los mismos.			
Categoría		Proceso			
Probabilidad	Media	Impacto	Grave	Exposición	
Consecuencias		Readaptación del plan de trabajo para la nueva estimación con los cambios necesarios para encauzar los requisitos a los reales.			
Plan de contingencia		Revisar los requisitos de manera habitual antes de incluir una funcionalidad.			

Tabla 14: Riesgo-007

Riesgo-008		Calidad baja de software			
Descripción		El producto final cuenta con una calidad baja en el código.			
Categoría		Producto			
Probabilidad	Media	Impacto	Grave	Exposición	
Consecuencias		Puede hacer que haya que remodelar cierta parte del código y afectar a la planificación del proyecto.			
Plan de contingencia		Conocimiento de las herramientas y sus prácticas habituales.			

Tabla 15: Riesgo-008

3.5 Seguimiento del proyecto

En este apartado se expondrá la planificación del proyecto comentando el tiempo estimado para cada una de las tareas y su contrastación con el tiempo final.

Los meses de septiembre y octubre estuve teniendo conversaciones con Joaquín para determinar la que sería la idea del trabajo. En un principio tenía alguna que otra idea pensada, pero escogí la idea proporcionada por Joaquín de la creación de una web para la administración de las aplicaciones creadas en la asignatura de sistemas móviles ya que era una idea en la que podría utilizar tecnologías similares a las que quería en las mías.

Debido a que hasta enero tenía prácticas y escasez de tiempo centré mis planes una vez acabadas intentando ir aprendiendo y haciendo pequeños avances en el diseño y la elaboración de requisitos de la aplicación. En un principio hubo una etapa inicial que duró todo el mes de noviembre y parte del mes de diciembre donde me centré en aprender Angular y Firebase para reducir el posible Riesgo-001 (escasez de conocimiento de las herramientas) y el Riesgo-008 (calidad baja de software).

A continuación, se muestra una tabla que expone las fechas reales y estimadas de las diferentes fases del proyecto.

Fase	Inicio estimado	Inicio Real	Fin estimado	Fin real
Inicio	4/1/2022	7/1/2022	20/1/2022	23/1/2022
Elaboración	20/1/2022	24/1/2022	20/2/2022	19/2/2022
Construcción	21/2/2022	20/2/2022	14/4/2022	3/5/2022
Transición	15/4/2022	4/5/2022	15/5/2022	1/6/2022

Tabla 16: Fases del proyecto con fechas estimadas y reales

Una vez expuestas las fechas se mostrarán los detalles de cada una de ellas.

3.5.1 Fase de Inicio

Una parte de esta etapa puede verse como el tiempo de búsqueda y adaptación sobre el framework que había hecho anteriormente. No lo he incluido en esta etapa debido a que no formaba parte de la planificación del proyecto. Se obtuvo una visión general del sistema propuesto y se estipularon los requisitos que debería tener la aplicación. La estimación fue más o menos correcta ya que empecé tres días más tarde de lo estimado, pero finalicé también tres días después.

3.5.2 Fase de elaboración

En esta segunda fase se dividieron tareas debido a que comprendía el análisis y diseño de la aplicación. Se hizo un prototipado de la aplicación, el diseño de las interfaces para incluir la información deseada y también un modelado del sistema a gran escala. Para esta primera etapa de la elaboración se estipuló en primera instancia un margen de 18 días que acabaron siendo 20 ya que el modelado me llevó más tiempo del pensado y las interfaces también me llevaron bastante tiempo.

Hubo posteriormente una segunda etapa dentro de la fase que comprendía la propia creación del proyecto, así como la instalación de las herramientas de trabajo. También se acomodó el área de trabajo y creó el primer esqueleto de la aplicación. El tiempo estimado fue de 12 días, pero debido a la experiencia previa con las herramientas la puesta a punto del entorno de trabajo fue relativamente rápida y puede acortar lo bastante para finalizar esta segunda fase antes de lo estimado en un principio.

3.5.3 Fase de construcción

Para esta tercera fase se empezó con la creación del producto final. Se elaboraron los casos de uso con detalle para tener una mejor idea de cómo avanzar en lo que a funcionalidad se refiere. Se completó el análisis y diseño empezados en la fase anterior. Poco a poco fueron construidos los diferentes elementos de la aplicación y se detallaron cuáles serían las pruebas que se realizarían una vez acabado este proceso. Por último, se creó el Manual de usuario para los diferentes roles de la web. Los tiempos estimados en esta fase son siempre los más complicados de cumplir ya que siempre suelen salir imprevistos a la hora de construir una web y ajustar la funcionalidad para que se consiga satisfacer todos los prerrequisitos acordados. Es

por esto por lo que hay una variación de algo más de quince días desde la estimación a la fecha de finalización real.

3.5.4 Fase de transición

En esta última fase se realizaron las pruebas sobre el proyecto. Se pulieron los últimos detalles de diseño e implementación que quedaron sueltos en anteriores fases. Se validó el manual de usuario y se hicieron pequeños ajustes. En última instancia la aplicación fue subida al servidor para su uso posterior y se procedió a comprobar su funcionamiento desde su lugar de uso. El tiempo estimado fue más o menos correcto teniendo en cuenta que tenía retraso aún de la fase anterior

3.6 Etapas del desarrollo

Durante las diferentes etapas y fases del desarrollo del proyecto se ha aplicado la metodología Kanban. El proceso Kanban consiste en la creación de una serie de tarjetas o post-its llamadas Kanban que se mueven siguiendo el proceso conocido como principio a fin. El objetivo de esto es mantener un flujo constante de actividades y representar de forma sencilla cuales se encuentran en desarrollo.

Las diferentes calificaciones que puede tener una tarea en un tablero Kanban son:

- **Backlog:** Se colocan las tareas que están pensadas por el sistema. Sirve de proceso de hibernación y son susceptibles a una posible revisión puesto que no se ha hecho ningún tipo de desarrollo en ellas.
- **To do:** Tareas pendientes en el sistema. Da una idea de cuáles tienen que ser las siguientes en las que el desarrollador tiene que enfocarse.
- **In progress:** Tareas que se encuentran en estado de desarrollo. Son sacadas de las tareas pendientes de la etapa "To do". Suelen ser limitadas puesto que el foco es ir terminándolas y así ir avanzando tareas por el Kanban.
- **Testing:** Última etapa por la que pasa una tarea. Una vez terminada tiene que pasar ciertas baterías de pruebas que validen el trabajo realizado.
- **Done:** Una vez testadas las tareas podemos darlas como finalizadas y almacenarlas aquí para tener el control cuáles ya no están en el foco de atención del desarrollador.

4 Análisis

4.1 Requisitos

Los requisitos son propiedades y características del producto en su estado final. Su posible incumplimiento puede hacer que proyectos se vean destinados al fracaso, es por esto, que su gestión debe verse en cualquier sistema como una prioridad.

Unos requisitos pobres pueden afectar que la calidad de un sistema caiga en picado o hasta genere defectos en su versión final.

A la hora de generar posibles requisitos sobre un sistema se pueden observar dos tipos diferenciados:

- Requisitos de usuario: declaraciones en lenguaje natural y percepción de los límites del sistema.

- Requisitos de sistema: Documentación detallada de los servicios ofrecidos por el sistema en desarrollo. Esta especificación debe ser completa puesto que lo que busca es ajustar al máximo todas las funcionalidades de cierto sistema.

Los requisitos en este proyecto estarán subdivididos de la manera habitual en proyectos de este tipo: requisitos funcionales, requisitos no funcionales y requisitos de información. Se va a ver cada uno de ellos con más detalle en los siguientes puntos.

4.1.1 Requisitos funcionales

Los requisitos funcionales son aquellos que definen las funciones de un sistema o subsistemas añadidos. En cuanto a requisitos de usuario en este apartado tendríamos frases generales sobre el comportamiento de entrada y salida del sistema. Por otro lado, los requisitos de sistema serían detallados y constarían de cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone que la versión final del sistema tendría que incluir.

ID	Descripción
RF-001	El sistema deberá permitir la existencia de varios roles de usuarios: administrador, usuario y súper administrador.
RF-002	El sistema deberá permitir que los usuarios puedan registrarse con un rol usuario.
RF-003	El sistema deberá permitir que los súper administradores puedan registrar nuevos administradores.
RF-004	El sistema permitirá borrar la cuenta a cualquier usuario o administrador del sistema.
RF-004	El sistema deberá permitir visualizar todas las aplicaciones presentes en el sistema con sus diferentes logos y nombres identificativos.
RF-005	El sistema deberá permitir al usuario filtrar las aplicaciones por nombre.
RF-006	El sistema deberá permitir al usuario visualizar el detalle de cada aplicación.
RF-007	El sistema deberá permitir al usuario visualizar en detalle las imágenes de cada aplicación.
RF-007	El sistema deberá permitir al usuario calificar y comentar cada una de las aplicaciones.
RF-008	El sistema deberá permitir al usuario redirigirse a la página de descarga de cualquier aplicación.
RF-009	El sistema deberá permitir al usuario ver cada uno de los comentarios que tiene una aplicación.
RF-010	El sistema deberá permitir al administrador visualizar todas las aplicaciones presentes en el sistema que correspondan a su empresa.
RF-011	El sistema deberá permitir al administrador modificar cada una de las aplicaciones que supervise (nombre, tipo, logo, descripción e imágenes).

RF-012	El sistema deberá permitir al administrador modificar la visibilidad de una aplicación en el entorno del usuario.
RF-013	El sistema deberá permitir al administrador borrar una aplicación que supervise.
RF-014	El sistema deberá permitir al administrador añadir imágenes para una determinada aplicación.
RF-015	El sistema deberá permitir al administrador añadir una nueva aplicación al sistema.
RF-016	El sistema deberá permitir al súper administrador modificar cada una de las aplicaciones (nombre, tipo, logo, descripción e imágenes).
RF-017	El sistema deberá permitir al súper administrador añadir administradores asociados a diferentes grupos.
RF-018	El sistema deberá permitir al súper administrador cambiar el número y nombre de los grupos presentes a la hora de añadir nuevos administradores que serán responsables de la edición de las aplicaciones de ese grupo.
RF-019	El sistema deberá permitir al súper administrador cambiar el número y nombre de categorías presentes a la hora de almacenar una aplicación nueva en el sistema por un administrador.

Tabla 17: Requisitos funcionales

4.1.2 Requisitos no funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que estos corresponden a los requisitos funcionales.

Estos son, por tanto, referidos a aquellos requisitos que no responden a la descripción de guarda o realización de una acción sino a sus características de funcionamiento.

ID	Descripción
RNF-001	El sistema podrá ser ejecutado tanto en un dispositivo móvil como en un ordenador.
RNF-002	El sistema deberá contener vistas sencillas e intuitivas para facilitar la usabilidad al usuario.
RNF-003	La respuesta del sistema no podrá superar los 5 segundos (si hay conexión estable).
RNF-004	Se mostrarán los cambios actualizados en todo momento.
RNF-005	El funcionamiento será fácil de recordar por el usuario.
RNF-006	La aplicación tendrá que contar con un grado de robustez que permita avisar al usuario de cualquier error.

Tabla 18: Requisitos no funcionales

4.1.3 Requisitos de información

ID	Descripción
RI-001	El sistema deberá almacenar información de usuario.
RI-002	El sistema deberá almacenar el rol del usuario.
RI-003	El sistema podrá almacenar la cuenta asociada de Google de un usuario.

Tabla 19: Requisitos de información

4.2 Modelo de casos de uso

En esta sección se va a mostrar el diagrama principal de casos de uso. Este se definiría tal que así:

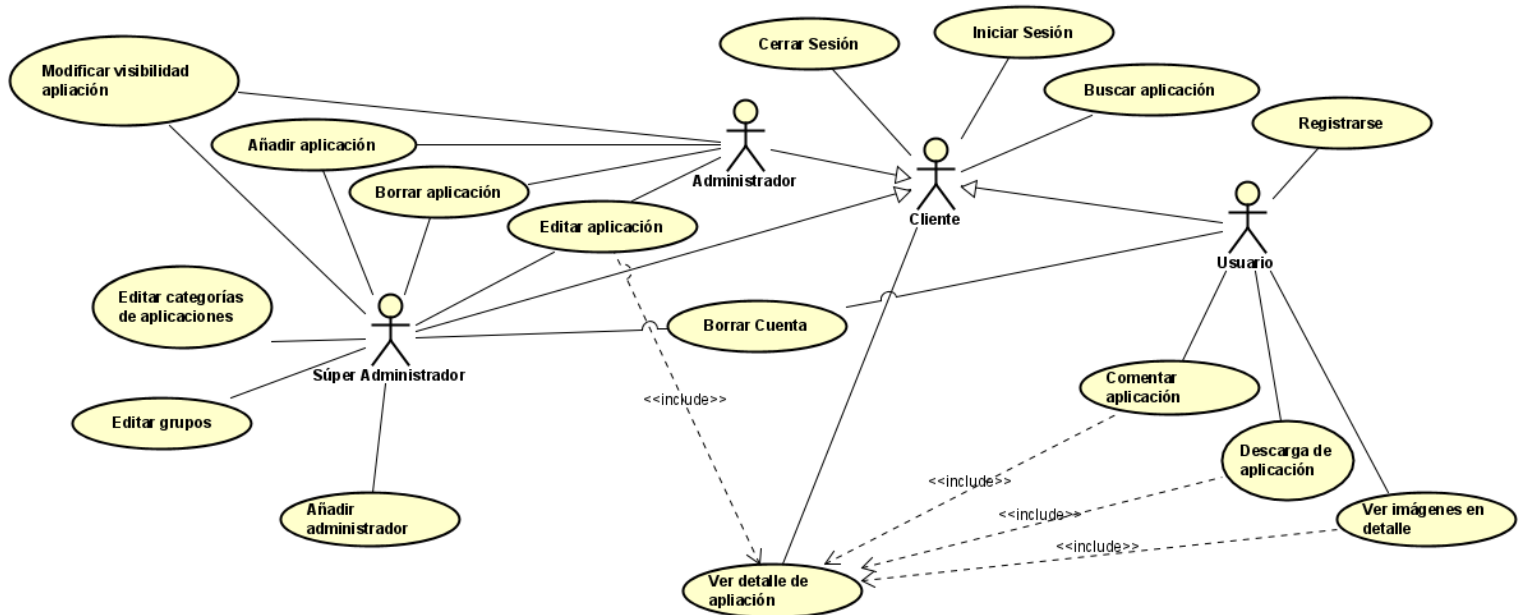


Figura 10: Modelo de casos de uso

4.3 Especificaciones de caso de uso

En este punto veremos estos casos de uso explicados individualmente. En ellos estarán detallados las posibles acciones y excepciones del sistema.

CU-001	Iniciar sesión en el sistema	
Actor	Cliente	
Descripción	El sistema deberá permitir al usuario iniciar sesión con su cuenta en el sistema.	
Precondición	El usuario debe estar situado en la pantalla de login del sistema y haberse registrado previamente (si es login con Google el registro previo no será necesario).	
Secuencia normal	Paso	Acción
	1	El usuario introduce su correo en el campo apropiado.
	2	El usuario introduce su contraseña en el campo apropiado.

	3	El usuario pulsa el botón Iniciar Sesión para entrar en el sistema.
	4	El sistema comprueba si los datos introducidos son correctos.
Postcondición	El usuario queda identificado en el sistema y es redirigido a la pestaña home del tipo de usuario que corresponda su rol.	
Excepciones	Datos incorrectos.	
	1	El sistema avisa de que los datos introducidos no son correctos.
	2	El usuario introduce datos válidos y el caso de uso continúa.
	Datos incompletos.	
	1	El sistema avisa de que los campos están incompletos.
	2	El usuario completa los datos y el caso de uso continúa.

Tabla 20: CU-001

CU-002	Registrarse en el sistema	
Actor	Usuario	
Descripción	El sistema deberá permitir al usuario registrarse si no tiene una cuenta adherida al correo.	
Precondición	El usuario debe estar situado en la pantalla de registro del sistema.	
Secuencia normal	Paso	Acción
	1	El usuario introduce su nombre de usuario en el campo apropiado.
	2	El usuario introduce su correo en el campo apropiado.
	3	El usuario introduce su contraseña en el campo apropiado.
	4	El usuario introduce su contraseña de nuevo idéntica a la anterior.
	5	El usuario pulsa el botón Registrarse.
	6	El sistema guarda los datos del nuevo usuario.
Postcondición	El usuario queda identificado en el sistema y es redirigido a la pestaña home del tipo de usuario que corresponda su rol.	
Excepciones	El correo ya está en uso.	
	1	El sistema avisa de que ya existe alguien con el mismo correo en el sistema.
	2	El usuario escribe un correo distinto y continua el caso de uso.
	El correo no es válido.	
	1	El sistema avisa de que el correo no es válido.
	2	El usuario cambia el correo por uno válido y el caso de uso continúa.
	La contraseña no es válida.	
	1	El sistema avisa de que la contraseña es demasiado corta.
	2	El usuario pone una contraseña mayor de 5 caracteres y el caso de uso continúa.
Campos incompletos.		

	1	Los campos de nombre de usuario, correo y contraseña no están completos.
	2	El usuario completa los campos y el caso de uso continúa.
	Usuario vuelve a la pantalla de iniciar sesión.	
	1	El caso de uso queda cancelado.

Tabla 21: CU-002

CU-003	Cerrar sesión en el sistema	
Actor	Cliente	
Descripción	El sistema deberá permitir a cualquier tipo de usuario cerrar sesión.	
Precondición	El usuario debe estar identificado en el sistema y en la pantalla de inicio de cada uno de los usuarios independientemente del rol	
Secuencia normal	Paso	Acción
	1	El usuario aprieta el botón Cerrar Sesión.
Postcondición	El usuario sale de su sesión y es redirigido a la pantalla de login.	

Tabla 22: CU-003

CU-004	Añadir administrador	
Actor	Súper Administrador	
Descripción	El sistema deberá permitir al súper administrador añadir administradores.	
Precondición	El súper administrador debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El súper administrador introduce un nombre válido.
	2	El súper administrador selecciona la empresa a la que estará asociado el administrador.
	3	El súper administrador introduce el email del nuevo administrador.
	4	El súper administrador introduce la contraseña del nuevo administrador
	5	El súper administrador hace click en Registrar.
	6	El sistema guarda los datos del nuevo administrador
Postcondición	El administrador queda identificado en el sistema.	
Excepciones	El correo ya está en uso.	
	1	El usuario escribe un correo distinto y continua el caso de uso.
	2	El usuario completa los datos y el caso de uso continúa.
	Correo no válido.	
	1	El sistema avisa de que el correo no es válido.
	2	El usuario cambia el correo por uno válido y el caso de uso continúa.
	Contraseña no es válida.	
	1	El sistema avisa de que la contraseña es demasiado corta.
	2	El usuario pone una contraseña mayor de 5 caracteres y el caso de uso continúa.
	Campos incompletos.	
1	Los campos de nombre de usuario, correo y contraseña no están completos.	
2	El usuario completa los campos y el caso de uso continúa.	

	Usuario vuelve a la pantalla de iniciar sesión.	
1	El caso de uso queda cancelado.	

Tabla 23: CU-004

CU-005	Borrar cuenta	
Actor	Usuario - Administrador	
Descripción	El sistema deberá permitir a cualquier usuario o administrador borrar su cuenta del sistema	
Precondición	El usuario o administrador debe estar registrado en el sistema y en la pantalla principal.	
Secuencia normal	Paso	Acción
	1	El usuario o administrador pulsa el botón de borrar cuenta.
	2	El sistema alerta del borrado y pregunta por segunda vez para comprobar con seguridad la decisión.
	3	El usuario o administrador confirma el borrado
	4	El sistema borra al usuario o administrador del sistema.
Postcondición	El administrador queda borrado del sistema impidiendo su futura identificación.	
Excepciones	Vuelve a pantalla principal.	
	1	El caso de uso finaliza

Tabla 24: CU-005

CU-006	Editar categorías de aplicaciones	
Actor	Súper Administrador	
Descripción	El sistema deberá permitir al súper administrador cambiar el tipo de categorías con las que se catalogará una aplicación.	
Precondición	El súper administrador debe estar registrado en el sistema y en la pantalla de edición de categorías.	
Secuencia normal	Paso	Acción
	1	El súper administrador añade, edita o elimina a su criterio categorías.
	2	El súper administrador da al botón de Guardar.
	3	El sistema guarda las nuevas categorías.
Postcondición	Las categorías son actualizadas.	
Excepciones	Datos incompletos.	
	1	El sistema avisa de que los campos están incompletos.
	2	El súper administrador completa los datos y el caso de uso continúa.
	Vuelve a pantalla de súper administrador	
	1	El caso de uso finaliza.

Tabla 25: CU-006

CU-007	Borrar aplicación	
Actor	Administrador – Súper Administrador	
Descripción	El sistema deberá permitir al administrador o súper administrador borrar cualquier aplicación a la que tenga acceso.	
Precondición	El administrador o súper administrador debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción

	1	El administrador o súper administrador presiona el botón Borrar de la aplicación que desee.
	2	El sistema pide confirmación para el borrado.
	3	El administrador o súper administrador confirma el borrado de la aplicación.
	4	El sistema borra la aplicación.
Postcondición	La aplicación no se encuentra en el sistema.	
Excepciones	Cancela el borrado.	
	1	El caso de uso finaliza.

Tabla 26: CU-007

CU-008	Añadir aplicación	
Actor	Administrador o súper administrador.	
Descripción	El sistema deberá permitir al administrador o súper administrador añadir una aplicación al sistema.	
Precondición	El administrador o súper administrador debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El administrador o súper administrador presiona el botón Añadir.
	2	El administrador o súper administrador escoge un logo para la aplicación.
	3	El administrador o súper administrador pone un nombre a la aplicación.
	4	El administrador o súper administrador edita la categoría de la aplicación.
	5	El administrador o súper administrador pone un enlace para su descarga.
	6	El administrador o súper administrador pone una descripción para la aplicación.
	7	(Opcional) El administrador introduce imágenes relativas a la aplicación que se mostrarán al ver más detalles de esta.
	8	El administrador presiona el botón Guardar.
	9	El sistema añade la nueva aplicación al sistema.
Postcondición	La aplicación no se encuentra en el sistema.	
Excepciones	El administrador vuelve a pantalla Home.	
	1	El caso de uso finaliza.
	Datos incompletos.	
	1	El sistema avisa de que los campos están incompletos.
	2	El usuario completa los datos y el caso de uso continúa.

Tabla 27: CU-008

CU-009	Editar aplicación	
Actor	Administrador o súper administrador	
Descripción	El sistema deberá permitir al administrador o súper administrador añadir una aplicación al sistema.	
Precondición	El administrador o súper administrador debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción

	1	El administrador o súper administrador presiona el botón Editar de la aplicación que desee.
	2	El administrador o súper administrador edita el logo, nombre, categoría, enlace o descripción de la aplicación.
	3	El administrador o súper administrador presiona el botón de guardar.
	4	El sistema guarda los cambios sobre la aplicación.
Postcondición	La aplicación es actualizada.	
Excepciones	El administrador o súper administrador vuelve a pantalla Home.	
	1	El caso de uso finaliza.
	Datos incompletos.	
	1	El sistema avisa de que los campos están incompletos.
2	El administrador o súper administrador completa los datos y el caso de uso continúa.	

Tabla 28: CU-009

CU-010	Buscar aplicación	
Actor	Cliente	
Descripción	El sistema deberá permitir a cualquier cliente registrado en el sistema la búsqueda por nombre de cualquier aplicación.	
Precondición	El cliente debe estar identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El cliente escribe algo en la entrada de búsqueda.
	2	El cliente presiona el botón Buscar.
	3	El sistema muestra los resultados coincidentes.
Postcondición	Los resultados coincidentes son mostrados por pantalla.	
Excepciones	No existen resultados.	
	1	El caso de uso finaliza.

Tabla 29: CU-010

CU-011	Ver detalle de aplicación.	
Actor	Cliente	
Descripción	El sistema deberá permitir al cliente ver el detalle de cada aplicación que tengan permiso.	
Precondición	El cliente debe estar identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El cliente presiona el botón de editar en caso de ser súper administrador o administrador o pulsa sobre la aplicación si es usuario.
	2	El sistema muestra los datos referentes a la aplicación. En el caso del administrador y súper administrador son editables dando paso al CU-009 y en el caso del usuario meramente visibles.

Tabla 30: CU-011

CU-012	Comentar aplicación	
Actor	Usuario	
Descripción	El sistema deberá permitir al usuario comentar cualquier aplicación.	
Precondición	El usuario debe estar identificado en el sistema y en el detalle de una aplicación.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el botón Calificar.
	2	El sistema muestra una ventana emergente dónde introducirá el comentario (o editará si ya comentó anteriormente).
	3	El usuario añade o edita el comentario.
	4	El usuario pulsa el botón publicar.
5	El sistema actualiza el comentario.	
Postcondición	El comentario y la calificación de la aplicación son actualizados en el sistema.	
Excepciones	El usuario cancela el comentario.	
	1	El usuario vuelve al detalle de aplicación y el caso de uso finaliza.

Tabla 31: CU-012

CU-013	Ver imágenes en detalle	
Actor	Usuario	
Descripción	El sistema deberá permitir al usuario ver imágenes de aplicaciones de en detalle.	
Precondición	El usuario debe estar identificado en el sistema y en el detalle de una aplicación.	
Secuencia normal	Paso	Acción
	1	El usuario presiona en la imagen que quiere ver con más detalle.
	2	El sistema muestra esa imagen únicamente y permite la navegación con las otras.

Tabla 32: CU-013

CU-014	Descarga de aplicación	
Actor	Usuario	
Descripción	El sistema deberá permitir al usuario redirigirse al enlace que tenga anexo cada aplicación.	
Precondición	El usuario debe estar identificado en el sistema y en el detalle de una aplicación.	
Secuencia normal	Paso	Acción
	1	El usuario presiona el botón Obtener.
	2	El sistema redirige al usuario al enlace que tenga puesto dicha aplicación.

Tabla 33: CU-014

CU-015	Modificar visibilidad de aplicación	
Actor	Administrador – Súper administrador	
Descripción	El sistema debe permitir al administrador o súper administrador modificar la visibilidad de las aplicaciones en las que tengan acceso de edición.	
Precondición	El administrador o súper administrador debe estar identificado en el sistema y en la pantalla de exposición de las aplicaciones.	
Secuencia normal	Paso	Acción
	1	El administrador o súper administrador presiona el botón de borrado de una aplicación.
	2	El sistema pregunta de nuevo al usuario si está seguro del borrado de la aplicación.
	3	El administrador o súper administrador confirman el borrado de la aplicación.
	4	El sistema borra la aplicación del sistema.
Postcondición	La aplicación es borrada del sistema y no estará disponible para ningún usuario.	
Excepciones	El administrador o súper administrador cancela el borrado	
	1	El caso de uso finaliza.

Tabla 34: CU-015

CU-016	Editar grupos	
Actor	Súper Administrador	
Descripción	El sistema deberá permitir al súper administrador cambiar los grupos presentes en el sistema.	
Precondición	El súper administrador debe estar registrado en el sistema y en la pantalla de edición de grupos.	
Secuencia normal	Paso	Acción
	1	El súper administrador añade, edita o elimina a su criterio grupos.
	2	El súper administrador da al botón de Guardar.
	3	El sistema guarda los nuevos grupos.
Postcondición	Los grupos son actualizados.	
Excepciones	Datos incompletos.	
	1	El sistema avisa de que los campos están incompletos.
	2	El súper administrador completa los datos y el caso de uso continúa.
	Vuelve a pantalla de súper administrador	
	1	El caso de uso finaliza.

Tabla 35: CU-016

4.4 Modelo de dominio

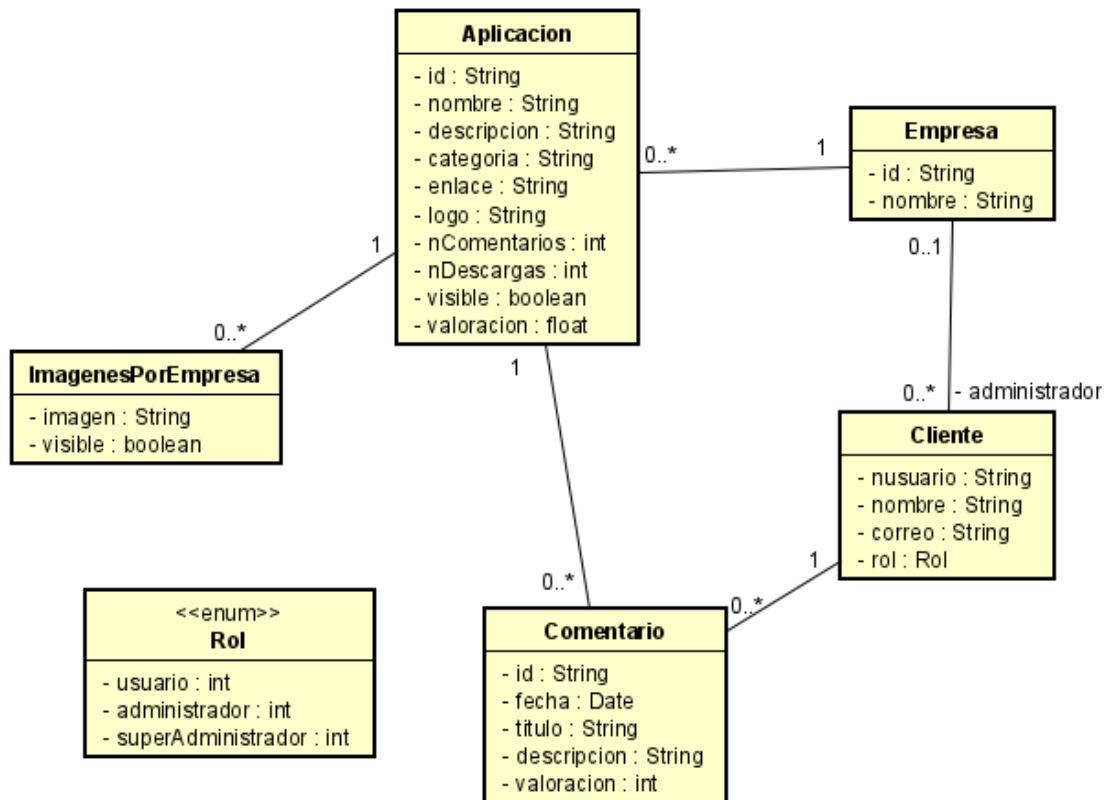


Figura 11: Modelo de dominio

A continuación, se expondrá de forma breve las diferentes clases con las que consta el diagrama:

- **Aplicacion**: Comprime la información relacionada con cada una de las aplicaciones del sistema. Consta de características nombre, descripción, categoría, enlace, logo y visibilidad que podrán ser editadas por un usuario administrador. También cuenta con datos estadísticos como su número de comentarios y descargas y la valoración total obtenida
- **Empresa**: Núcleo que comprende todas las aplicaciones. Distintivo que puede servir para diferenciar las diferentes empresas que pueden tener administradores que añaden, editen o eliminen las aplicaciones.
- **Cliente**: Clientes del sistema. El nombre será útil en caso de que un usuario haga un comentario puesto que el este vendrá acompañado del nombre del usuario que lo escriba y no dejará constancia de su correo para los demás usuarios.
- **Comentario**: Opiniones de los usuarios sobre las aplicaciones. Un usuario puede comentar en todas las aplicaciones, pero una única vez por cada una de ellas, siendo posible editar dicho comentario si la opinión sobre esta cambia.

4.5 Diagramas de secuencia

Los diagramas de secuencia son también importantes a la hora de la construcción de una aplicación web puesto que nos muestran las distintas interacciones de objetos a través del tiempo en cada caso de uso. Esto ayudará a entender ciertos detalles de la implementación de cada escenario de los casos de uso expuestos anteriormente, incluyendo los objetos y clases que se usan para implementar el escenario expuesto, así como los mensajes entre estos mismos objetos y clases. Siguiendo el orden establecido a la hora de detallar cada uno de los casos de uso se van a detallar cada uno de los diagramas de secuencia resultantes.

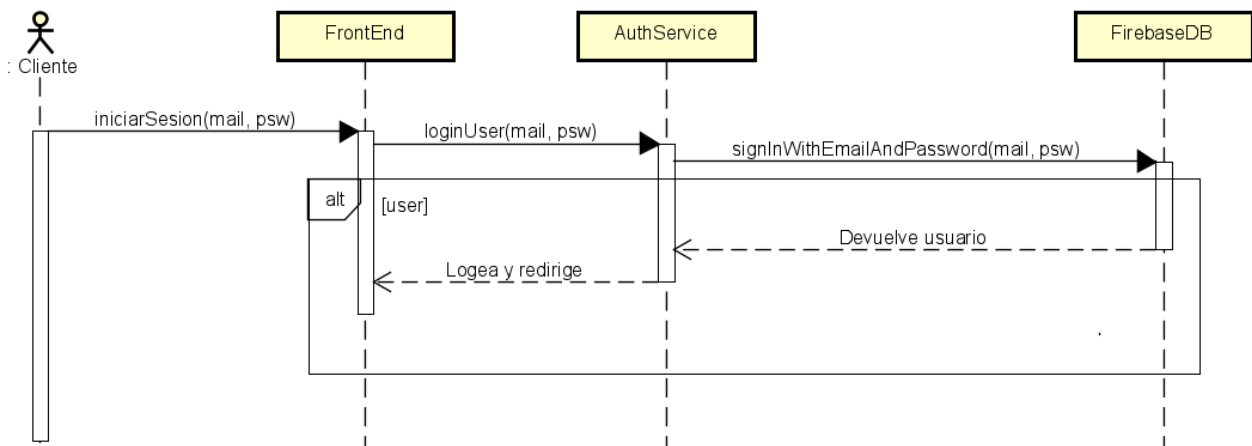


Figura 12: Diagrama de secuencia de CU001

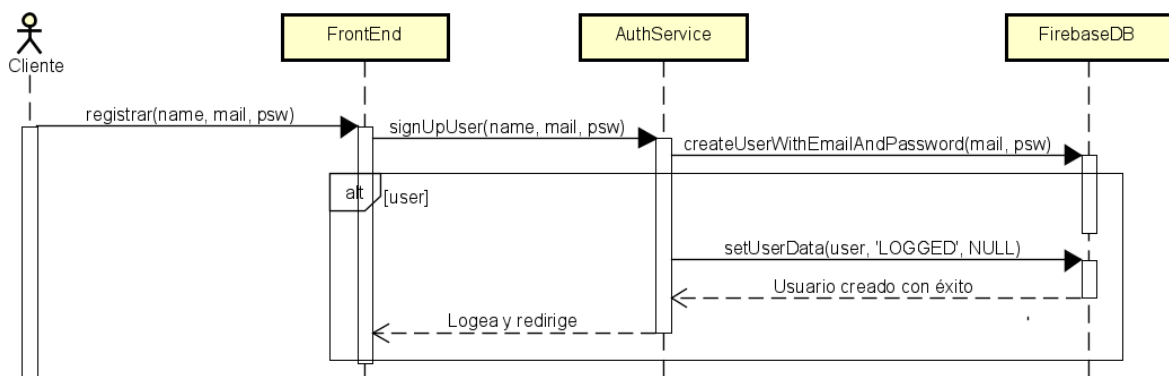


Figura 13: Diagrama de secuencia de CU002

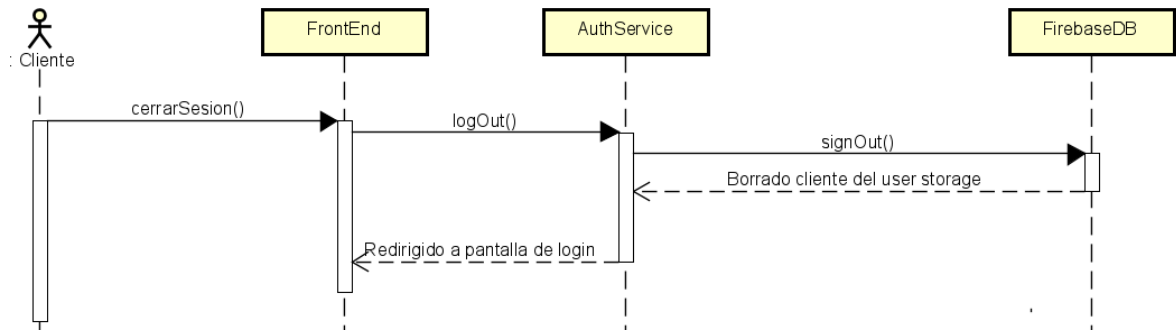


Figura 14: Diagrama de secuencia de CU003

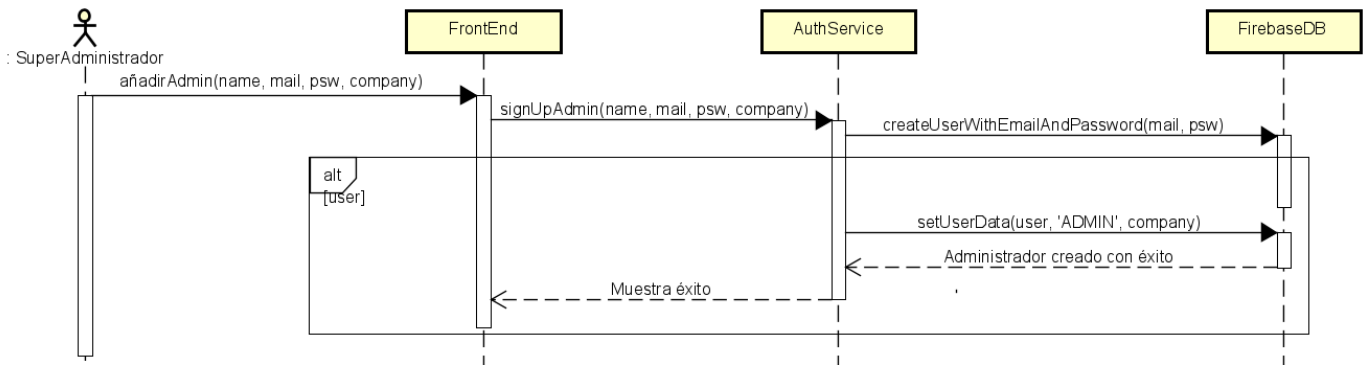


Figura 15: Diagrama de secuencia de CU004

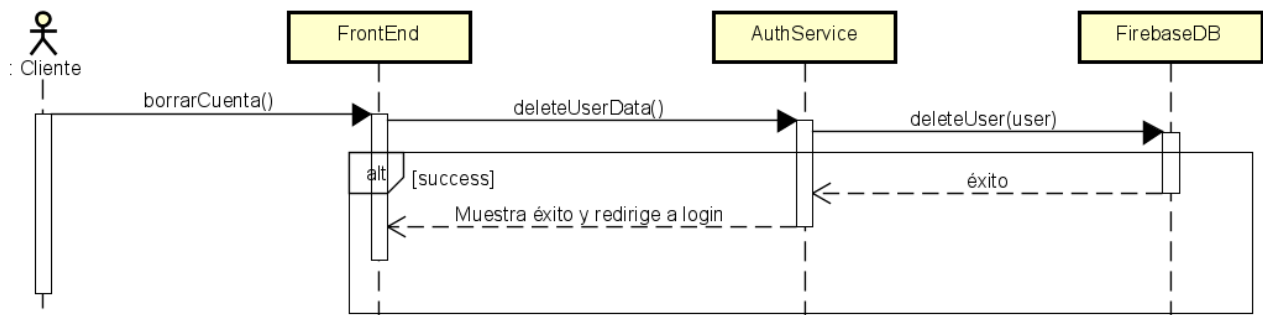


Figura 16: Diagrama de secuencia de CU005

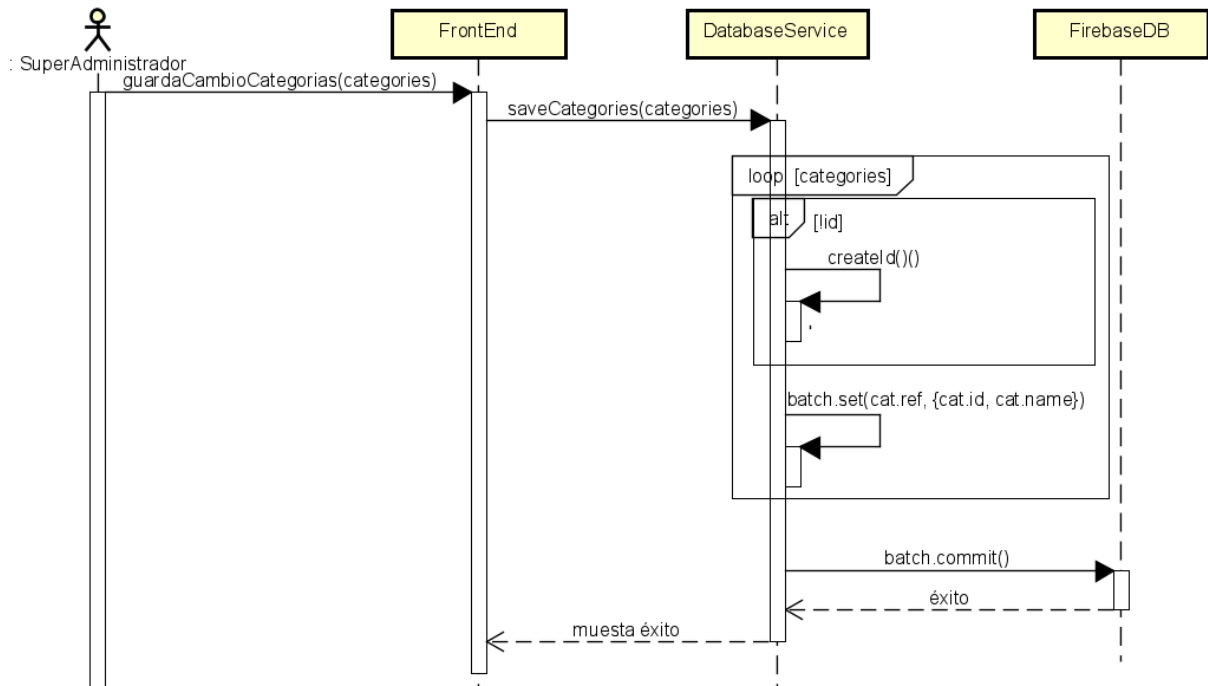


Figura 17: Diagrama de secuencia de CU006

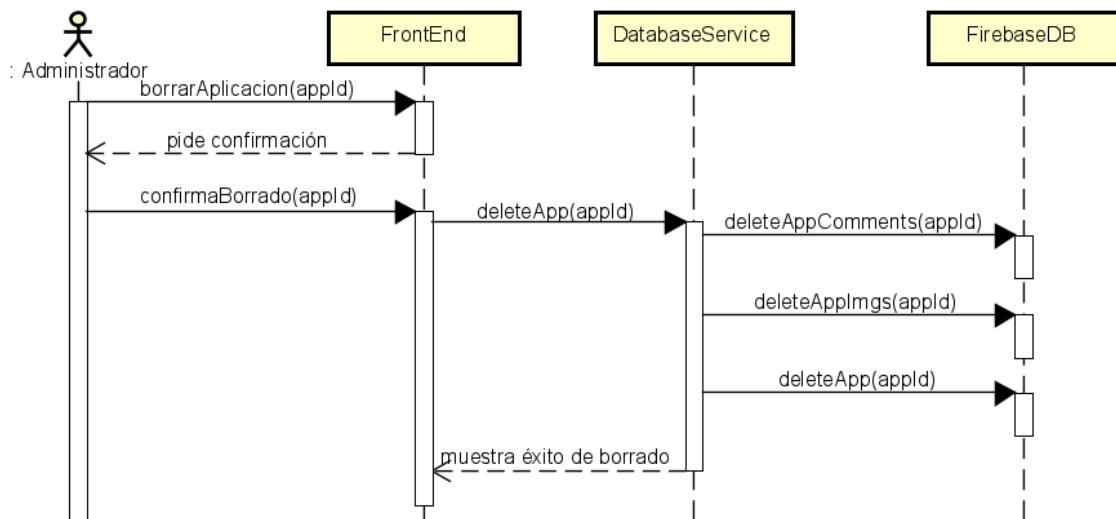


Figura 18: Diagrama de secuencia de CU007

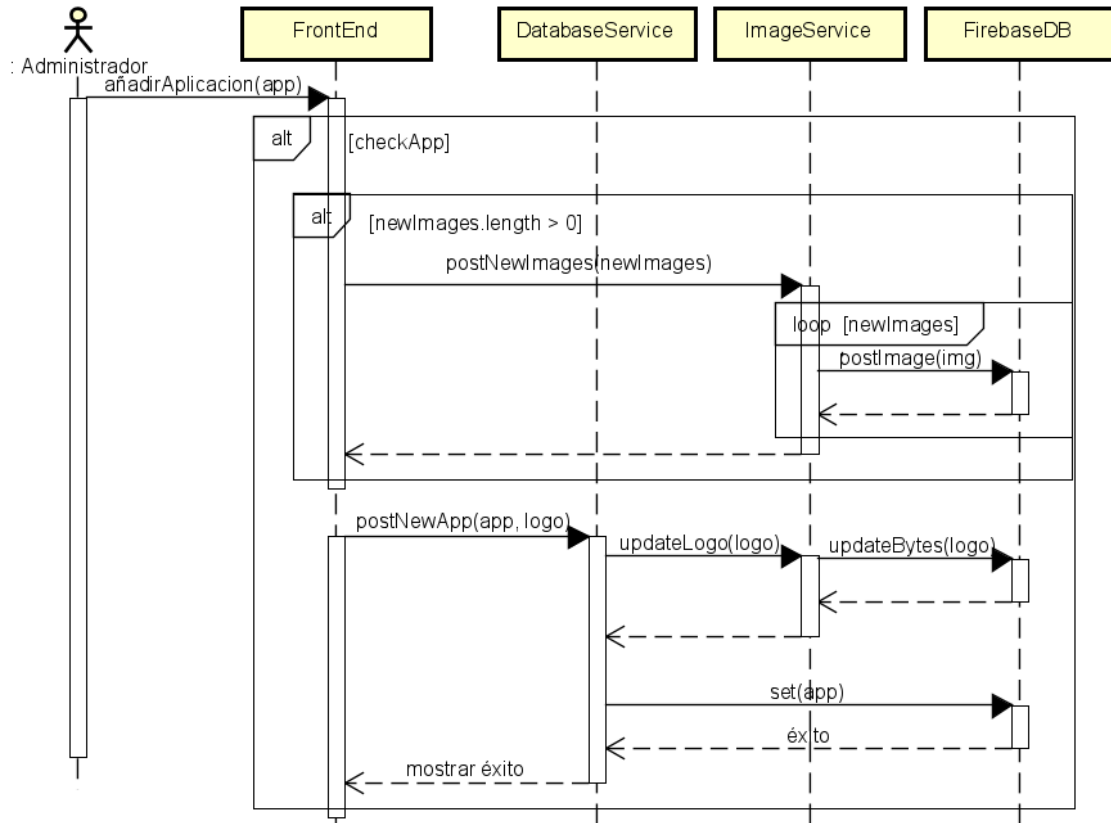


Figura 19: Diagrama de secuencia de CU008

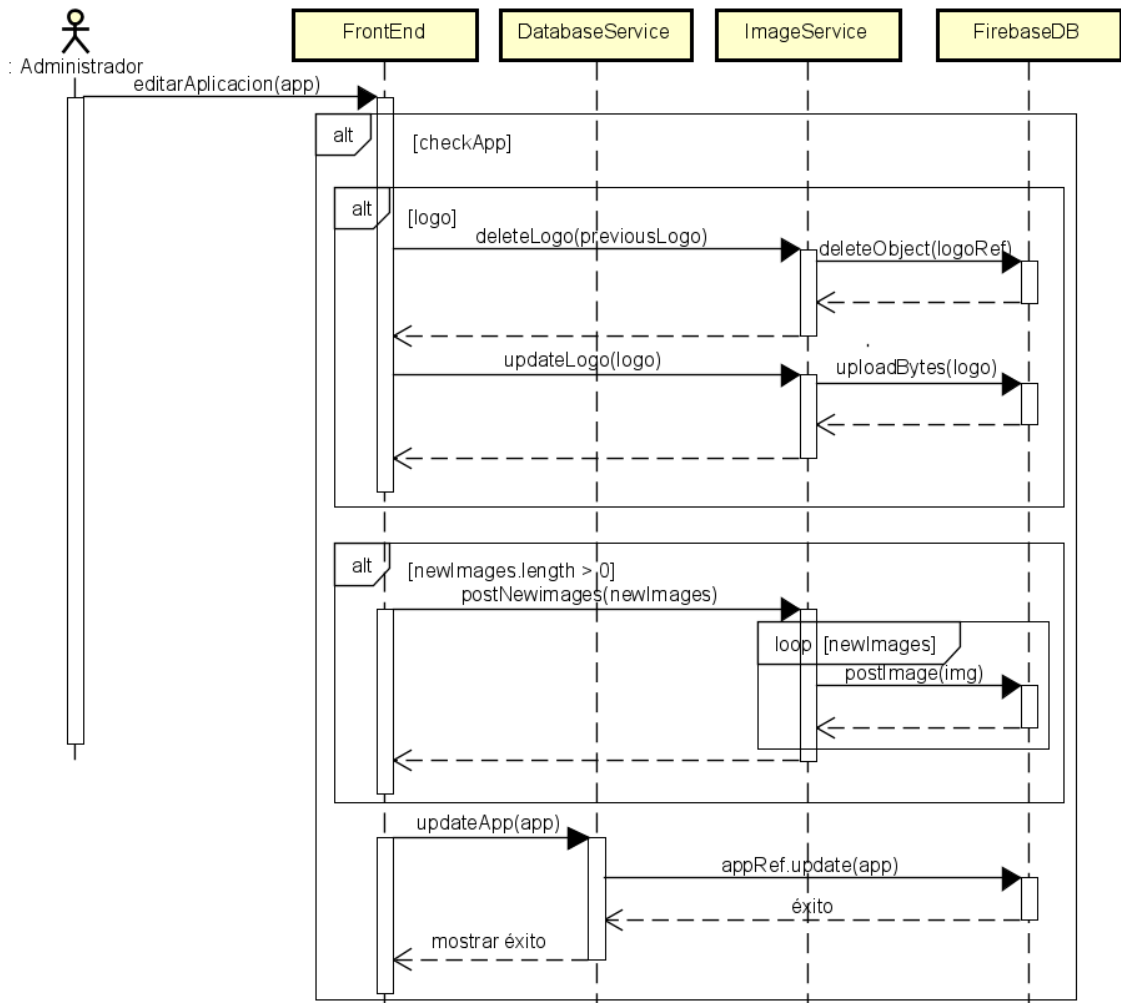


Figura 20: Diagrama de secuencia de CU009

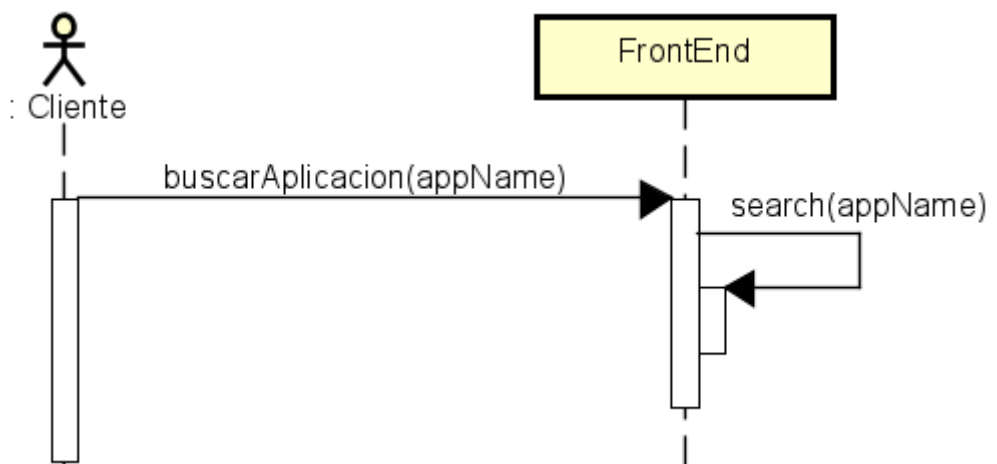


Figura 21: Diagrama de secuencia de CU010

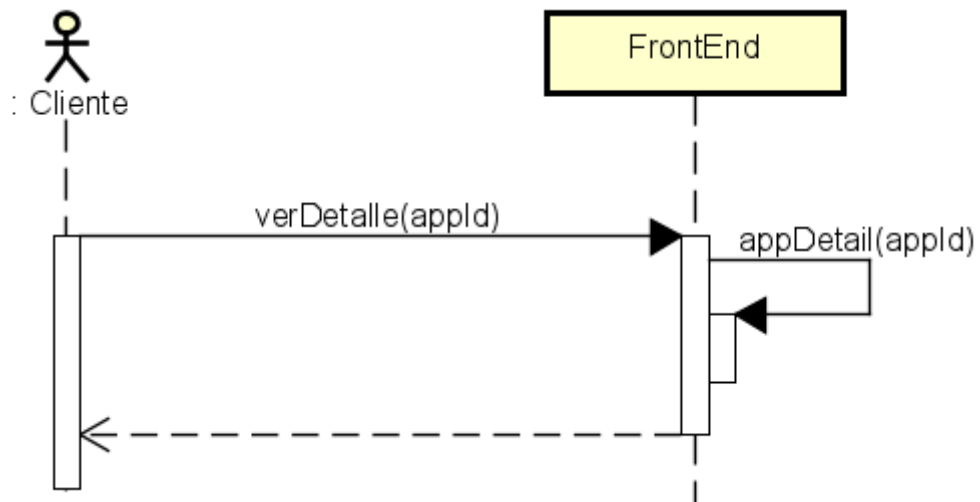


Figura 22: Diagrama de secuencia de CU011

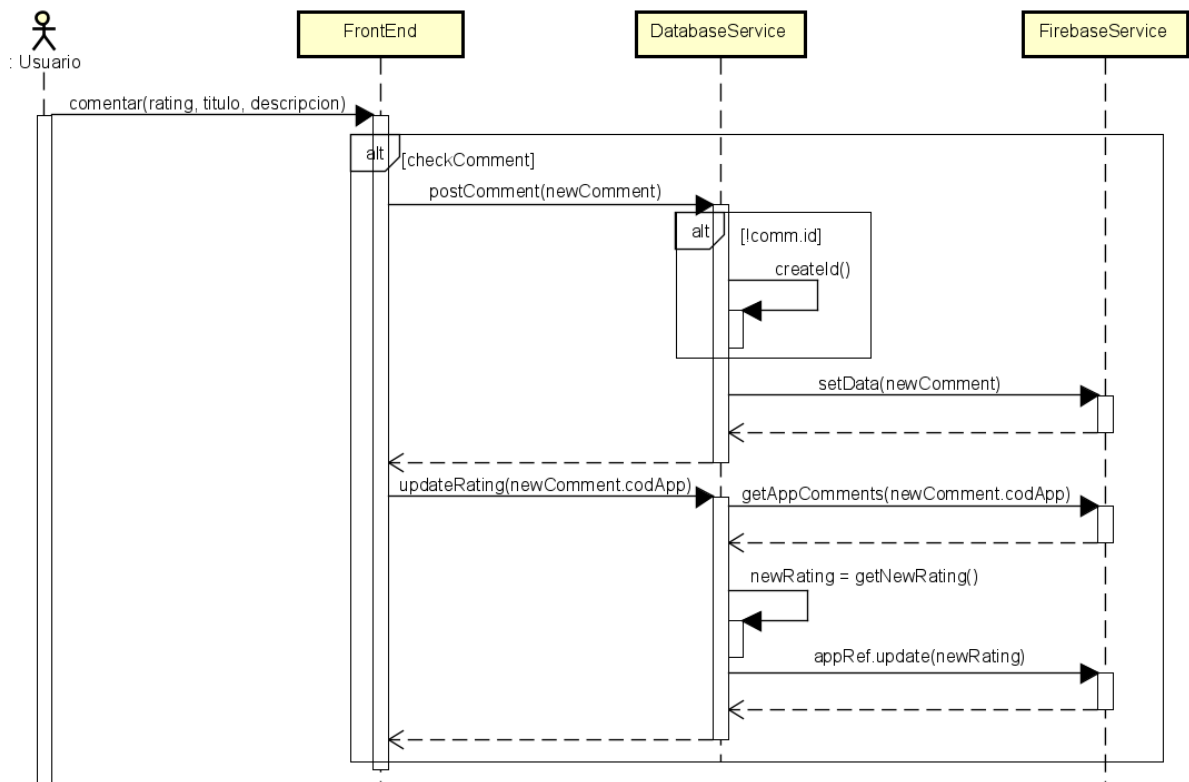


Figura 23: Diagrama de secuencia de CU012

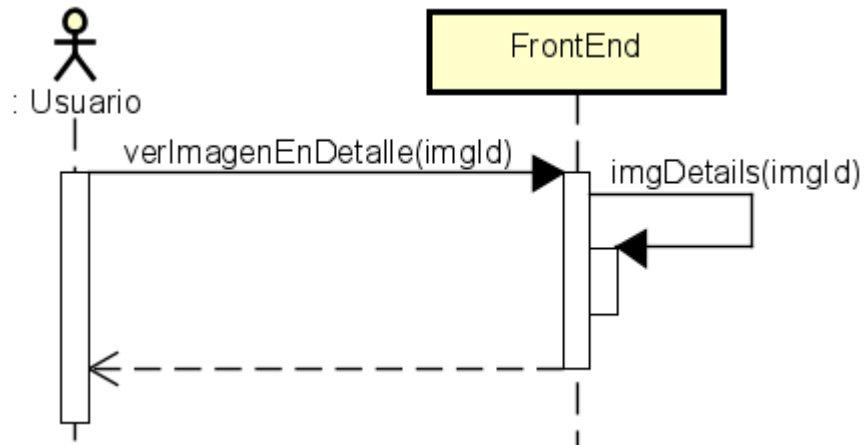


Figura 24: Diagrama de secuencia de CU013

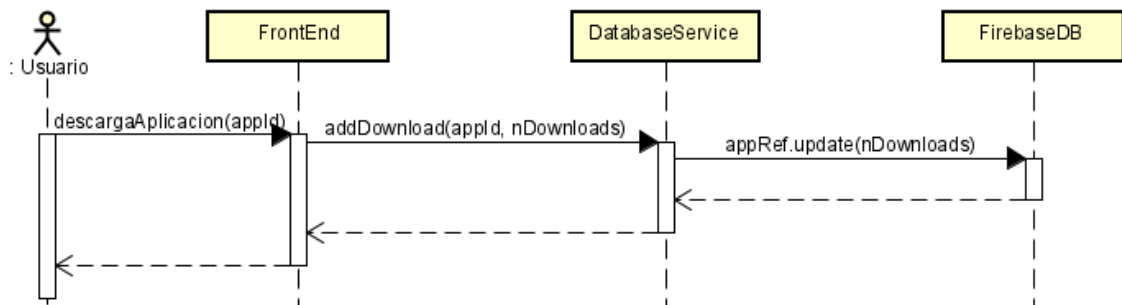


Figura 25: Diagrama de secuencia de CU014

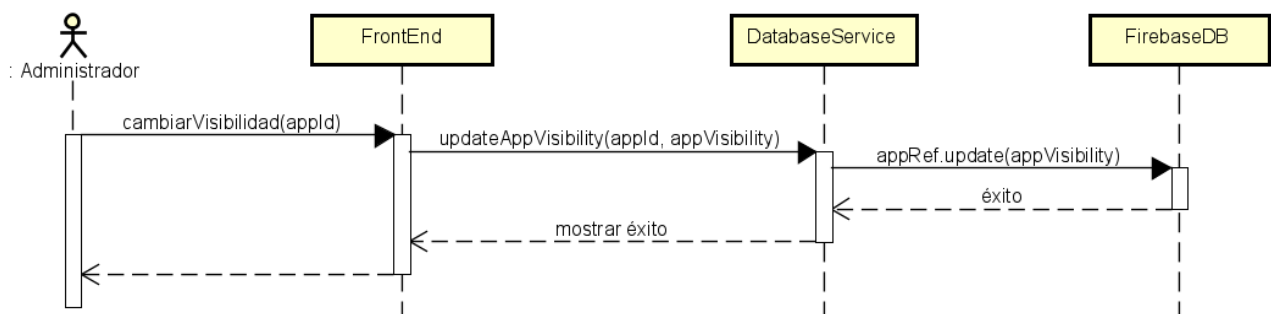


Figura 26: Diagrama de secuencia de CU015

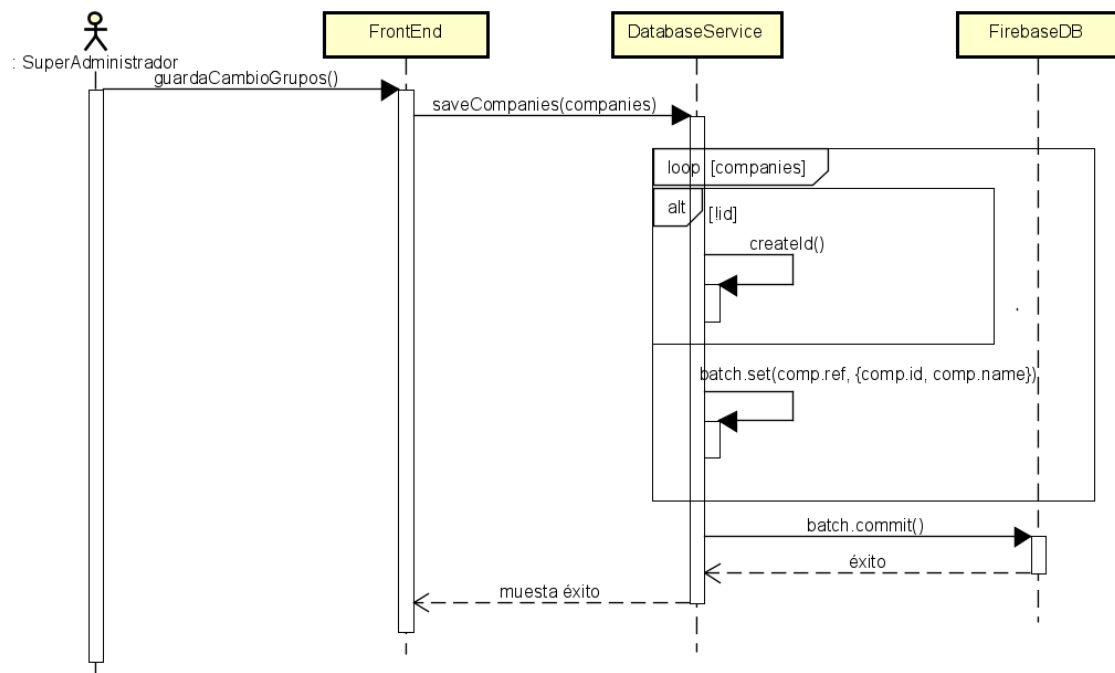


Figura 27: Diagrama de secuencia de CU016

5 Diseño

5.1 Arquitectura

Para la aplicación se pensó en un principio en la utilización del framework Angular puesto que era una herramienta con la que había tenido un breve contacto, me gustó la experiencia y quería aprender más sobre ella.

Según la documentación oficial de Angular, la arquitectura de una aplicación en Angular es basada por ciertos conceptos fundamentales. Los bloques de construcción básicos son componentes angulares organizados en NgModules.

NgModules recopila código relacionado en conjuntos funcionales. Una aplicación de Angular está definida por un conjunto de NgModules. Una aplicación siempre tiene al menos un módulo raíz que permite el arranque y, por lo general, tiene muchos más módulos de funciones. Estos NgModules se definen por una clase decorada con `@NgModule()` que toma un único objeto de metadatos, cuyas propiedades describen el módulo. Estas propiedades son:

- **declarations:** Los componentes, directivas, y pipes que pertenecen a este NgModule.
- **exports:** El subconjunto de declaraciones que deberían ser visibles y utilizables en las plantillas de componentes de otros NgModules.
- **imports:** Otros módulos cuyas clases exportadas son necesarias para las plantillas de componentes declaradas en este NgModule.
- **providers:** Creadores de servicios que este NgModule aporta a la colección global de servicios; se vuelven accesibles en todas las partes de la aplicación. (También puedes especificar proveedores a nivel de componente, que a menudo es preferido).

- **bootstrap**: La vista principal de la aplicación, llamado el componente raíz, que aloja todas las demás vistas de la aplicación. Sólo el NgModule raíz debe establecer la propiedad bootstrap.

Para comprender cómo se construyen las aplicaciones hay que entender primero algunos de los conceptos básicos que conforman el framework:

- **Módulos**: contenedor en el que se declaran los componentes, directivas, pipes y servicios que se van a utilizar en una parte de la aplicación. La utilización de varios módulos favorece la distribución dentro de una aplicación ya que es posible combinar componentes de otro módulos. También define las dependencias con otros módulos los cuales pueden ser importados para usarse en los componentes del mismo módulo.

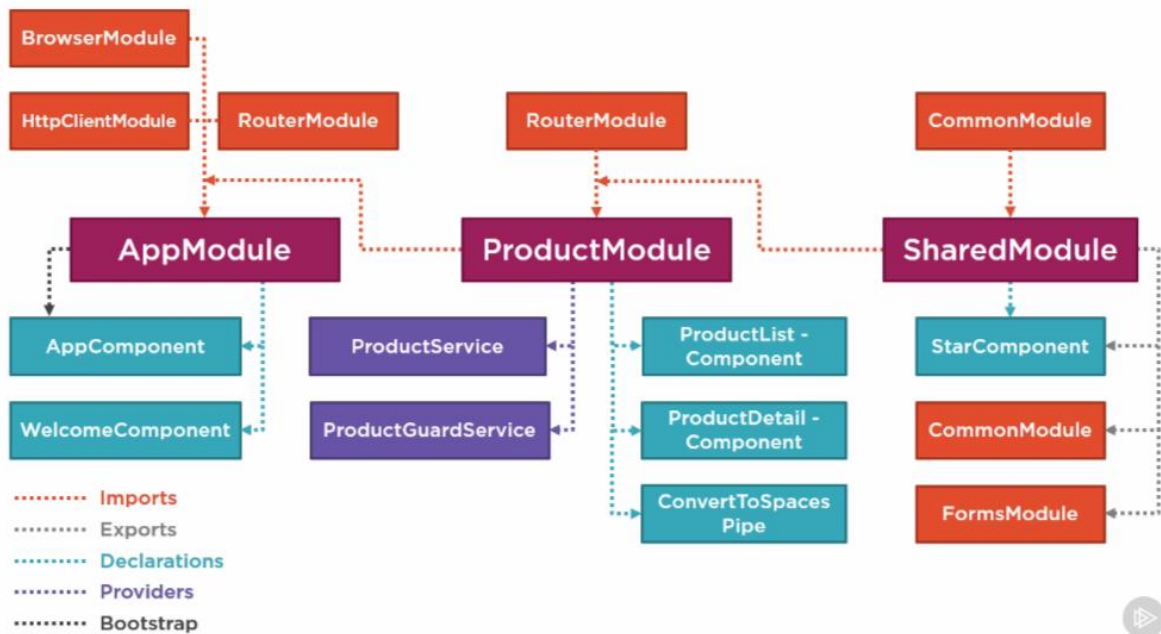


Figura 28: Ejemplos de módulos Angular

- **Componentes**: Elemento generalmente está compuesto por un template (archivo HTML) que proporciona la interfaz que va a ver el usuario, un archivo de estilos para ajustar dicho template y un script o archivo de lógica apoyado en el lenguaje TypeScript que aportará las propiedades y métodos de la vista.

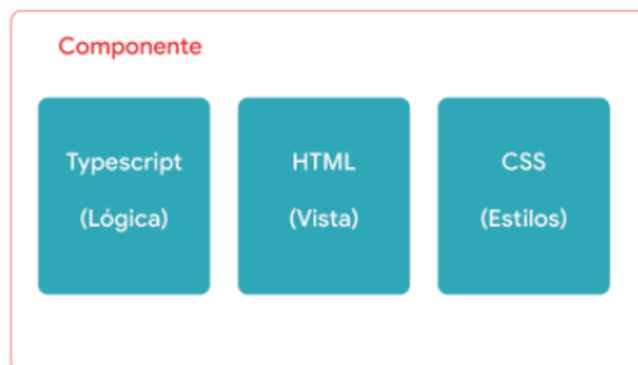


Figura 29: Ejemplo de componente Angular

- **Servicios:** son proveedores de datos, mantienen lógica de acceso a ellos y operativa relacionada con el negocio y las cosas que se hacen con los datos dentro de la aplicación.

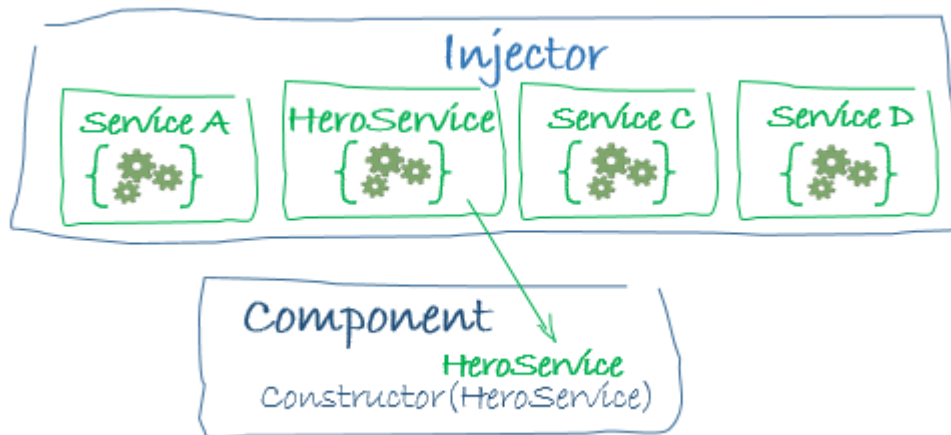


Figura 30: Ejemplo de servicios Angular

- **Routing:** servicio que maneja la navegación entre las diferentes vistas de la aplicación.

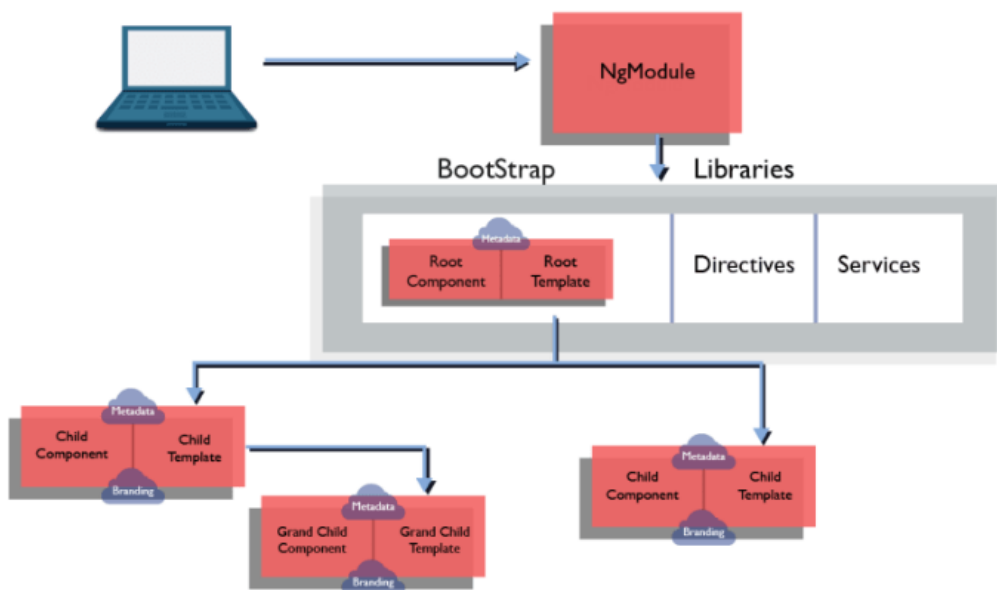


Figura 31: Ejemplo de routing Angular

Una vez vistos estos conceptos básicos de la tecnología repasaremos los atributos clave de la arquitectura:

- **La estructura de módulos:** como se ha explicado a la hora de definir los módulos es una buena práctica en aplicaciones utilizar varios módulos que puedan separar componentes y estructurar mejor la aplicación. En este caso al no tener apartados muy distintivos a la hora de la realización no se ha optado por la estructura multimodular.

- **Organización de los componentes:** Es muy importante en un proyecto angular diferenciar de forma adecuada los componentes. Es por esto por lo que siguiendo un estructurado con carpetas los componentes han sido separados por su tipo de cliente, quedando así diferenciados los componentes usados para usuarios, administradores y súper administradores, de esta manera los componentes de registro e identificación tienen su propia carpeta puesto que afectan a diferentes roles de usuario.
- **Librerías personalizadas:** es una práctica recomendable rehusar ciertas librerías o servicios para evitar duplicados. En el caso de esta aplicación existe tanto un servicio que accede a la base de datos como otro para las diferentes imágenes que son compartidos por varios componentes.
- **Técnicas de comunicación:** normalmente en una aplicación es necesario reciclar información proveniente de otra vista. Para hacer esto en una comunicación entre componentes padres e hijos se utilizan las etiquetas de @Input y @Output. Como en esta aplicación no existe tal relación padre e hijo dónde un componente es el que contiene al otro la información es pasada por argumentos del enrutamiento entre dichos componentes.

Además de estos atributos arquitectónicos hay que seguir algunas prácticas recomendadas:

- Mantener los métodos y clases pequeñas
- Nombres claros en variables y métodos
- No reutilizar variables
- Dividir el problema en partes más pequeñas

5.2 Diagrama de diseño

En esta aplicación se han usado las tecnologías de Angular para la parte de frontend y Firebase para el almacenamiento. Para la parte backend se han utilizado las herramientas proporcionadas por el propio Firebase a través de varios servicios creados en la parte de Angular.

Vamos a repasar la estructura de directorios creada para el proyecto.

En primer lugar, tenemos la carpeta model dónde se encuentran las clases que modelan los objetos de la base de datos:

- **Application:** Cada una de las aplicaciones ubicadas en el sistema.
- **Category:** Las diferentes categorías que pueden ser otorgadas a una aplicación.
- **Comment:** Los comentarios de las aplicaciones.
- **Company:** Las diferentes entidades o grupos a las que pueden pertenecer los administradores y que diferencian las aplicaciones a las cuales tienen acceso.
- **Image:** Cada una de las imágenes que son subidas a Firebase Storage.
- **State:** Objeto interno para modelar el comportamiento de la pantalla Home.
- **Users:** Cada usuario que pertenezca al sistema, ya sea administrador o cliente.

Por otra parte, tenemos las carpetas dentro del directorio pages dónde están situados los diferentes componentes y que diferenciaremos en:

- **Login:** Componentes referentes a la autenticación de los usuarios.

- **User-identification:** Componente correspondiente a la identificación de los usuarios.
- **User-sign-up:** Componente correspondiente al registro de nuevos usuarios.
- **Client:** Componentes referentes al uso de la aplicación por parte de los clientes.
 - **App-page:** Componente correspondiente a la página de inspección de cada una de las aplicaciones por el cliente.
 - **Delete-user:** Componente con ventana emergente que sirve de confirmación al cliente para borrar su usuario del sistema.
 - **Pop-up:** Componente con ventana emergente para añadir o editar comentarios hechos por el cliente.
 - **User-home:** Componente referente a la página de inicio del cliente dónde son mostradas todas las aplicaciones.
- **Company:** Componentes referentes al uso de la aplicación por parte de los administradores.
 - **Admin-home:** Componente referente a la página de inicio del administrador dónde puede editar las aplicaciones.
 - **Admin-app:** Componente de detalle de edición de la aplicación.
 - **Delete-pop-up:** Componente con ventana emergente para asegurar que el borrado de una aplicación no sucede de forma trivial.
- **Super-user:** Componentes referentes al uso de la aplicación por parte del súper administrador.
 - **Super-user:** Componente dónde se mostrará el menú en el que el súper usuario navegará entre las diferentes funcionalidades.
 - **Super-user-home:** Componente por el cual los administradores pueden dar de alta nuevos administradores y modificar las categorías de las aplicaciones que se dan de alta
 - **Super-user-app:** Componente de detalle de edición de la aplicación.
 - **Super-user-admin:** Componente usado para añadir nuevos administradores al sistema.
 - **Super-user-company:** Componente para editar, añadir y borrar grupos al sistema.
 - **Super-user-category:** Componente para editar, añadir y borrar categorías de aplicaciones al sistema.

También es destacable la carpeta dónde estarán los servicios que nos ayudarán tanto a la toma y entrega de información de la base de datos como a gestiones internas comunes que no tienen que ser repetidas por cada componente. Estos serán:

- **Auth (guard):** Guarda que nos asegurará que todo usuario identificado en el sistema es redirigido a su página principal y no pueda redirigirse a la de identificación y registro para evitar problemas de autenticación.
- **Auth (service):** Servicio que consta de todas las funciones relativas a la autenticación vía Firebase. Estos métodos han sido sacados de la documentación inicial y sirven para comprobar que todo el proceso de identificación se hace de forma correcta.
- **Bd:** Servicio que comprime las funciones de traspaso de datos entre el frontend y el entorno de firestore.
- **Error:** Servicio que se encarga de avisar al usuario en caso de que este esté sufriendo un error de cualquier tipo.

- **Image:** Servicio para el tratado de las imágenes en el sistema con el Storage proporcionado por Firebase.

5.3 Diseño de la base de datos

La base de datos como se ha expuesto anteriormente se ha construido con la tecnología Firebase, en particular con su extensión Cloud Firestore. Esta base de datos NoSql es alojada en la nube y puede ser accedida para apps en Apple, Android y Web directamente desde los SDKs nativos.

Aquí se va a mostrar un diagrama con las clases creadas para la aplicación:

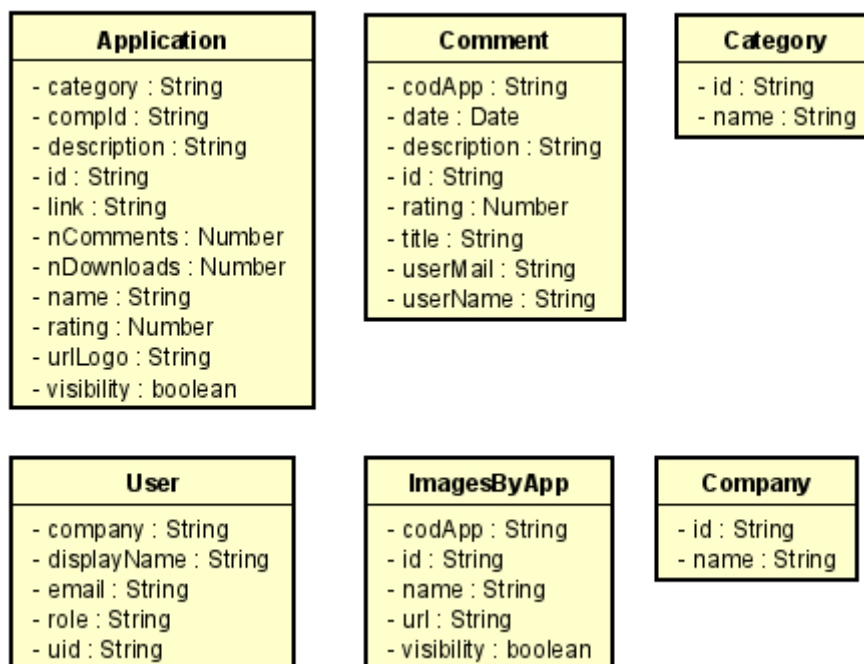


Figura 32: Diagrama de clases de la base de datos

5.4 Diccionario de la base de datos

A continuación, serán expuestos los nombres con los significados para cada uno de ellos:

Application	
Columna	Descripción
category	Categoría a la que pertenece la aplicación.
compld	Id de la empresa a la que pertenece la aplicación.
description	Descripción de la aplicación.

link	Enlace para la descarga de la aplicación.
nComments	Número de comentarios totales que la aplicación ha recibido.
nDownloads	Número de veces que alguien ha pulsado el botón de descarga.
name	Nombre de la aplicación.
rating	Valoración media de la aplicación.
urlLogo	Enlace del logo de la aplicación alojado en Firebase.
visibility	Visibilidad para el usuario cliente de la aplicación.

Tabla 36: Contenido de tabla Application en la BD

User	
Columna	Descripción
company	Empresa a la que pertenece el usuario.
displayName	Nombre del usuario. Este nombre será utilizado para mostrarse en los comentarios de las aplicaciones y así no hacer público su correo.
email	Correo del usuario.
role	Tipo de usuario: súper administrador, administrador o cliente.
uid	Id identificativo del usuario.

Tabla 37: Contenido de tabla User en la BD

Category	
Columna	Descripción
Id	Id identificativo de la categoría.
Name	Nombre de la categoría.

Tabla 38: Contenido de tabla Category en la BD

Comment	
Columna	Descripción
codApp	Id de la aplicación a la que pertenece el comentario.
date	Fecha de publicación del comentario.
description	Descripción del comentario.
id	Id identificativo del comentario.
rating	Valoración del cliente sobre la aplicación.
title	Título del comentario.
userMail	Correo del usuario que escribe el comentario.
userName	Nombre de usuario que escribe el comentario.

Tabla 39: Contenido de la tabla Comment en la BD

ImagesByApp	
Columna	Descripción
codApp	Id de la aplicación de la aplicación a la que pertenece la imagen.
id	Id de la imagen.
name	Nombre de la imagen.
url	Url de la imagen.
visibility	Visibilidad de la imagen.

Tabla 40: Contenido de la tabla ImagesByApp en la BD

Company	
Columna	Descripción
id	Id de la empresa.
name	Nombre de la empresa.

Tabla 41: Contenido de la tabla Company en la BD

6 Diseño de interfaz

6.1 Diseño centrado en usuario

Para esta práctica se ha seguido el diseño centrado en el usuario (DCU). Se trata de una metodología o filosofía que trata de dar crear productos o plantearlos para que den solución a necesidades específicas de los usuarios.

En este caso la necesidad era la de centralizar el acceso a las aplicaciones de la asignatura de Sistemas Móviles y las instrucciones y el apoyo han sido brindados tanto por el tutor Joaquín como de algunos exalumnos de la escuela. Con este enfoque se busca incluir a los usuarios lo máximo posible a través del proceso de diseño y desarrollo. A parte de lo conocido como usabilidad en el grado de fluidez e interacción con el sistema, el usuario tiene que estar satisfecho con las necesidades que aproxima el producto. Esta ideología ha sido implantada ya en muchos proyectos de diseño responsive, web o publicitario.

6.2 Análisis de usuarios

En la práctica del diseño de interfaces es esencial la búsqueda del colectivo de usuarios al que está destinada la aplicación para poder ceñirse a sus necesidades en la mayor medida de lo posible. Para el análisis del colectivo se utilizan ciertas características que engloban al usuario promedio de la aplicación. Estas características son tanto físicas (edad, sexo, morfología...) como mentales (nivel de educación, experiencia con sistemas semejantes...). Aquí se tendrá en cuenta que la mayoría de las personas que manipulen la aplicación sean alumnos de la escuela por lo que el perfil de usuario quedaría tal que así.

Perfil de usuario:

- **Edad:** Entre 18 y 30 años
- **Sexo:** Ambos
- **Experiencia tecnológica:** Alta
- **Experiencia con aplicaciones similares:** Alta

Motivación

La principal motivación del proyecto es tener un punto de acceso centralizado al trabajo de los alumnos de la asignatura de Sistemas Móviles dónde podrán visualizar el trabajo de alumnos que han pasado por la escuela de manera cómoda y simple. Además, podrán subir su propia aplicación dónde expondrán de manera breve el contenido e imágenes de sus interfaces.

6.3 Tareas de usuario

Las tareas que realizará un usuario en la aplicación dependerán del rol que tenga este usuario. Los roles van a dividirse en 3:

- **Usuario:** Este será el usuario promedio de la aplicación. Solo tendrá acceso a la visualización de las aplicaciones. Podrá acceder a la información individual de cada aplicación y comentar su opinión dando una nota entre 0 y 5 estrellas. También tendrá acceso al proceso de descarga de cualquier aplicación.
- **Administrador:** Usuario que tendrá permisos para editar y borrar la información sobre las aplicaciones que pertenezcan a su grupo. También tendrá la posibilidad de añadir una nueva aplicación al sistema, así como editar la visibilidad de las ya presentes.
- **Súper administrador:** Como administrador puede editar de igual forma a lo visto en el punto anterior solo que sin restricciones de grupo ya que tiene acceso a cada una de las aplicaciones alojadas en el sistema. Tendrá la potestad de añadir a administradores al sistema. Este puede modificar a su vez las posibles categorías que engloban a cada tipo de aplicación (entretenimiento, música, finanzas...), así como los grupos que se repartirán las aplicaciones.

6.4 Bocetaje

Para la parte de diseño de la interfaz primero se mostrará un diagrama de flujo entre las diferentes pantallas de la aplicación:

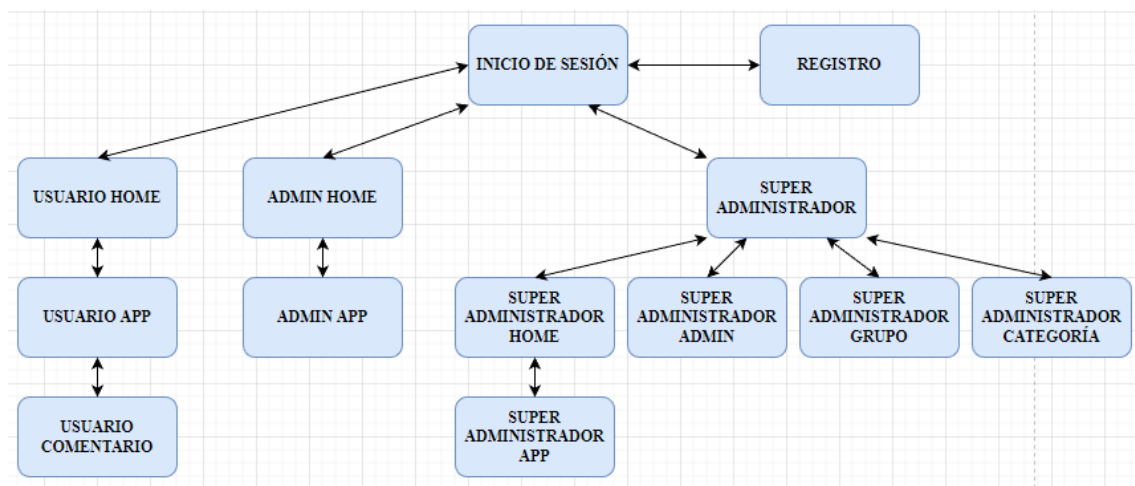


Figura 33: Bocetaje del flujo de la aplicación

Ahora se expondrán los primeros diseños completos de la interfaz completos. Estos diseños se han hecho gracias a la aplicación de uizard.

Inicio de Sesión

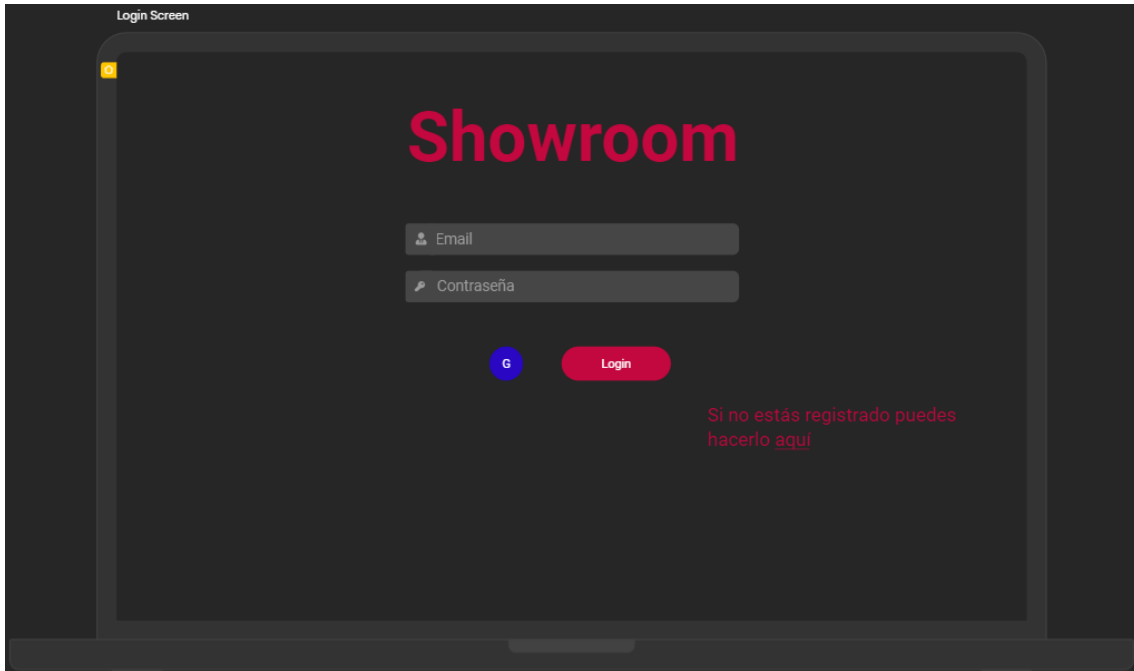


Figura 34: Boceto de la pantalla de inicio de sesión

Esta será la página de inicio de sesión para todo tipo de usuario. Contará con un campo para mail y contraseña. También podrá identificarse mediante Google (no será necesario que se registre en este caso pues se hará de forma automática al iniciar sesión).

Registro

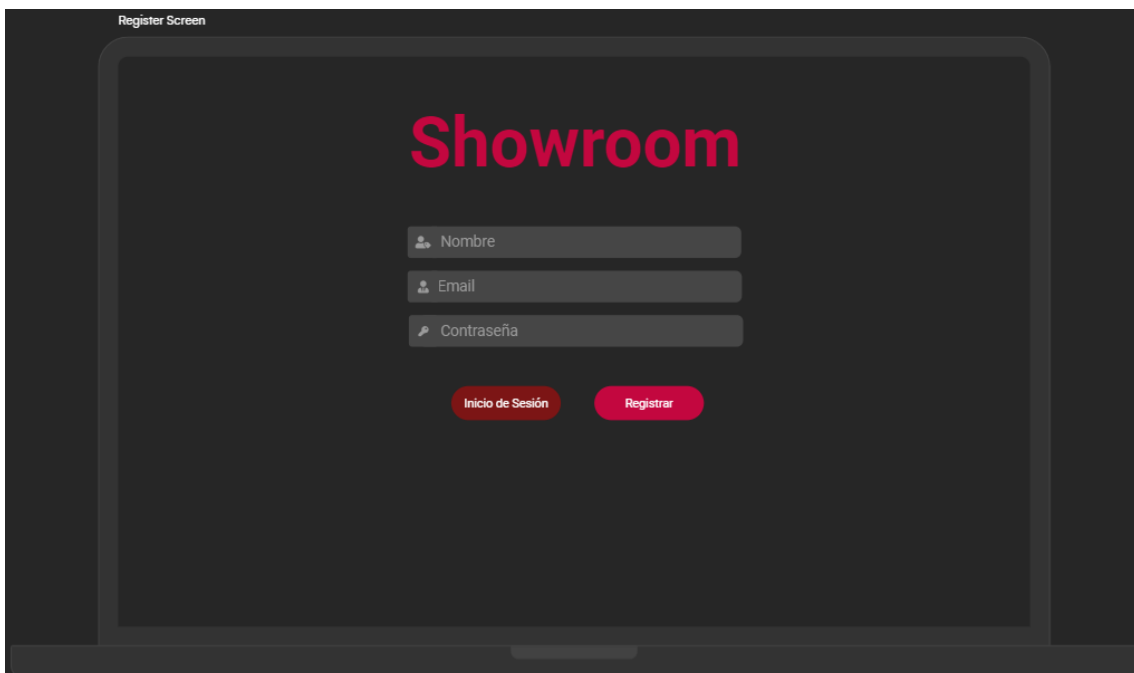


Figura 35: Boceto de pantalla de registro de usuario

Página que servirá a los usuarios para crear una cuenta con el fin de poder acceder a la pantalla dónde se mostrarán las imágenes. Se pide el nombre para no mostrar el correo públicamente en caso de que se haga un comentario en una aplicación.

Usuario Home

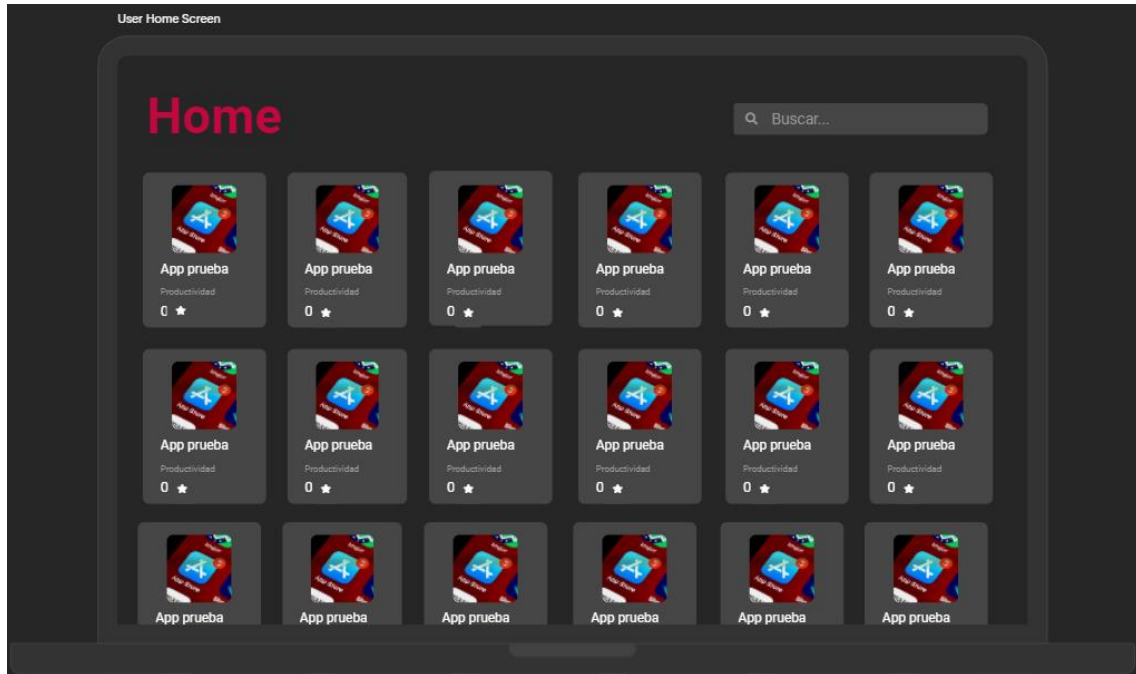


Figura 36: Boceto de pantalla Home de usuario

Página en la que se mostrarán las diferentes aplicaciones. Se proporcionará una barra de búsqueda para que sea más cómodo y rápido en caso de que existan muchas aplicaciones. También se expondrá la cantidad de estrellas media dada por los usuarios y la categoría de la aplicación.

Usuario App

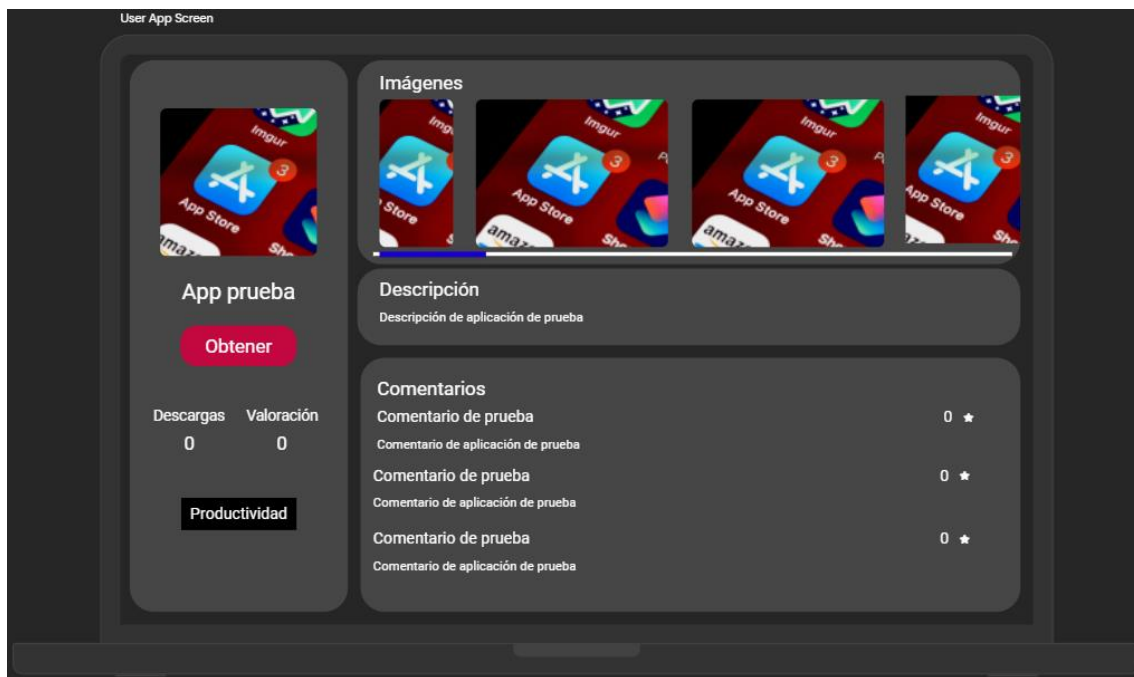


Figura 37: Boceto de pantalla de detalle de aplicación de usuario

En esta página se expondrá con detalle la aplicación que seleccione el usuario, así como sus imágenes y comentarios.

Admin Home – Super Administrador Home

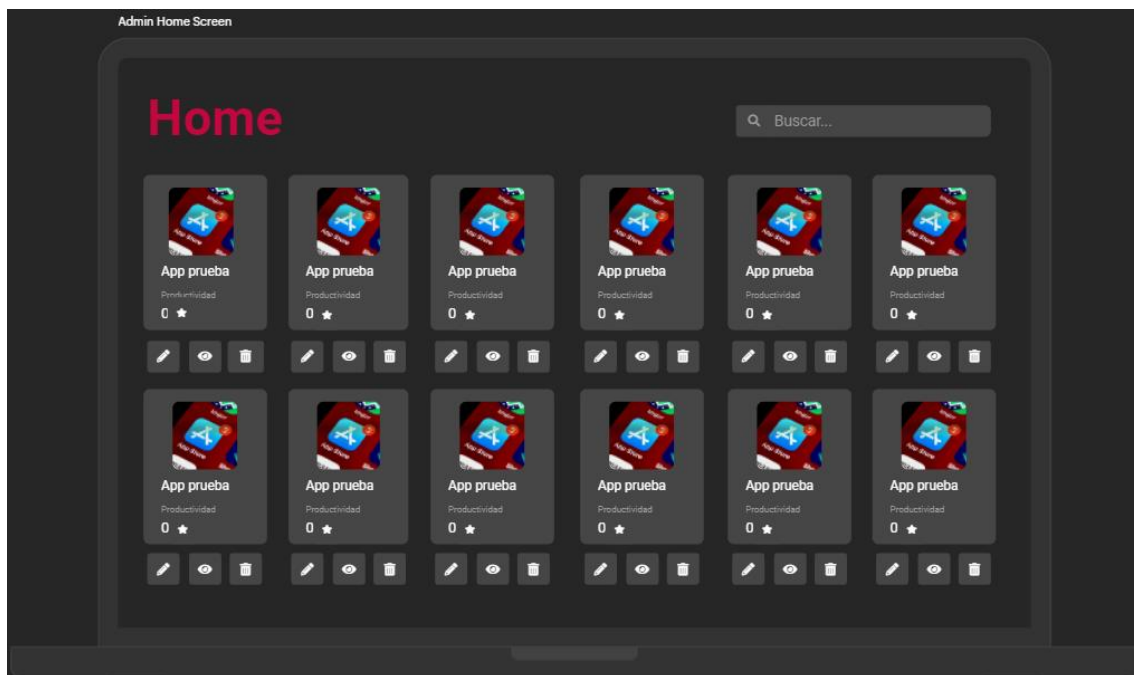


Figura 38: Boceto de pantalla de Home de súper administrador

Se mostrarán las aplicaciones como en el caso de los clientes. Los administradores tendrán acceso a ellas para su edición, podrán cambiar su visibilidad y borrarlas. La única diferencia entre

las pantallas de administradores y súper administradores es que los primeros solo tendrán acceso a aquellas aplicaciones pertenecientes a su grupo mientras que el súper administrador será capaz de acceder a cada una de ellas.

Admin App – Super Administrador App

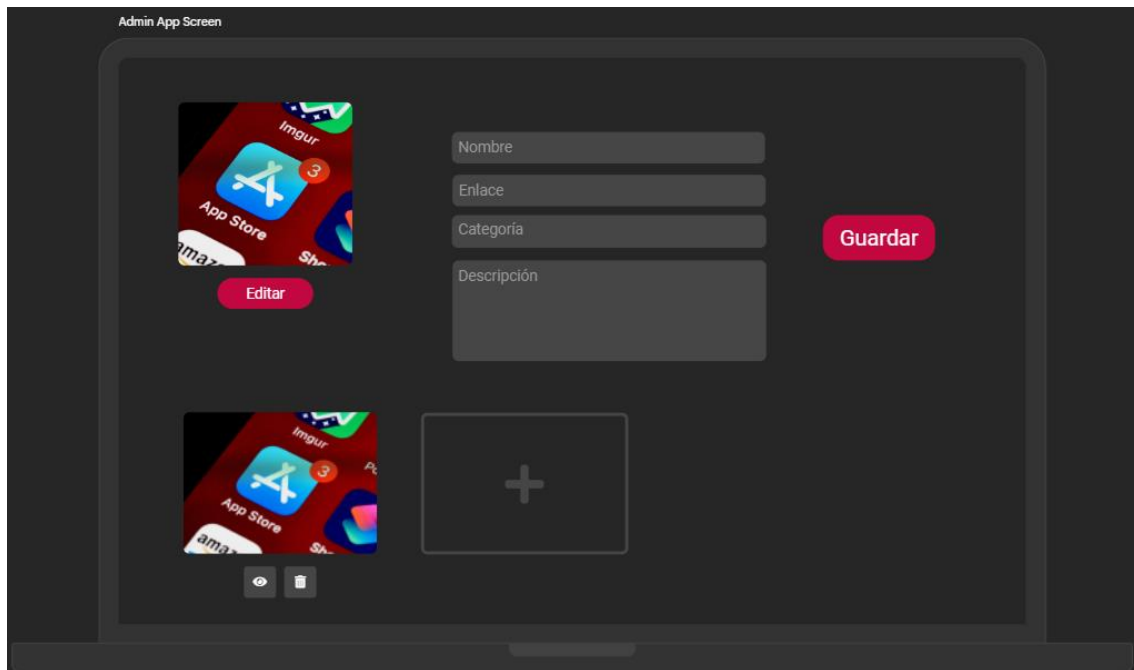


Figura 39: Boceto de pantalla de detalle de aplicación de administrador

Esta pantalla expondrá los detalles de la aplicación en los que se podrán hacer cambios. También se presentarán las imágenes que podrán hacerse o no visibles y editar de igual forma el logo de la aplicación. En caso de ser una aplicación nueva estos registros vendrán vacíos y será obligatorio su relleno antes de guardar. Si se modifica cualquier característica, pero no se pulsa el botón de guardado no se efectuará el cambio.

Súper Administrador

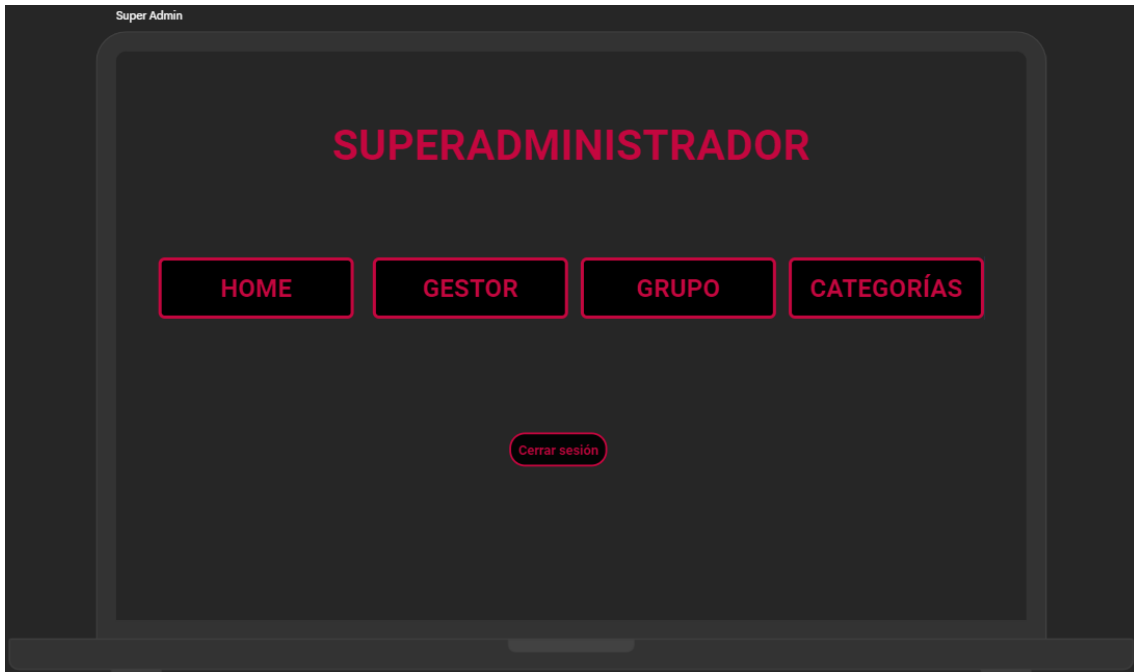


Figura 40: Boceto de pantalla de selección de súper administrador

Pantalla de selección del súper dónde podrá navegar entre las diferentes opciones.

Súper Administrador Admin

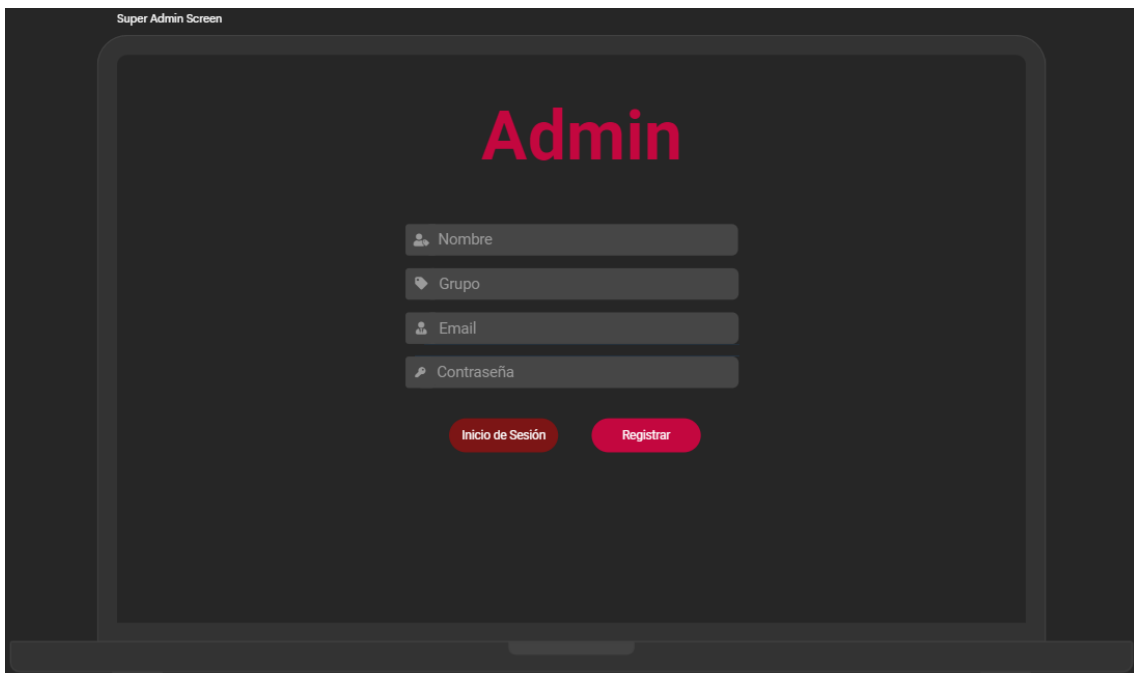


Figura 41: Boceto de pantalla de añadir administrador

Pantalla en la cual el súper administrador puede añadir un nuevo administrador en el sistema.

Super Administrador Grupo

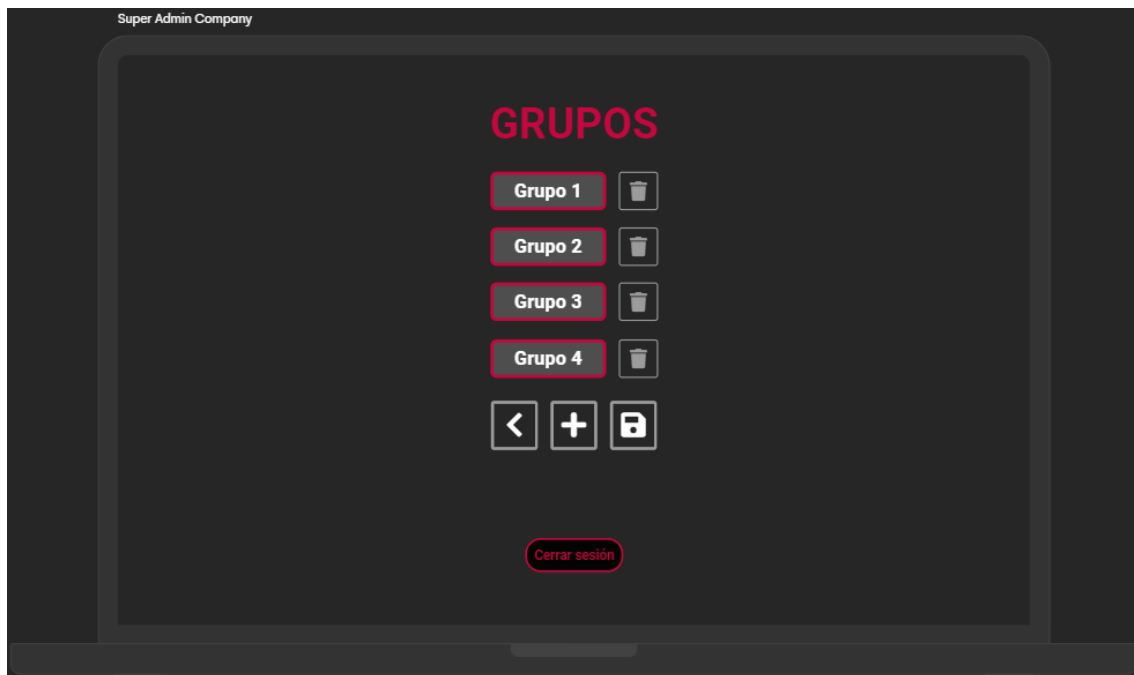


Figura 42: Boceto de pantalla de edición de grupos

Pantalla de control de grupos. Se permitirá su adición al resto y borrar los ya presentes. Carecerán de valor los cambios si no se da al botón de guardado.

Super Administrador Categorías

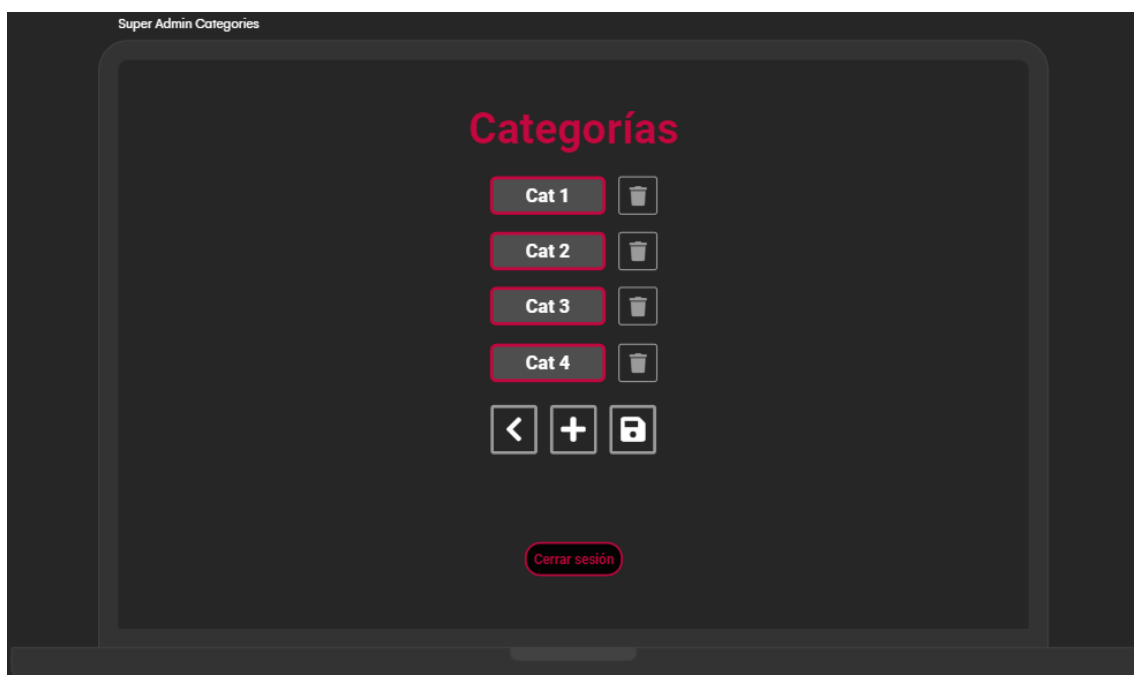


Figura 43: Boceto de pantalla de edición de categorías

Pantalla de control de categorías. El súper administrador en este caso podrá efectuar los cambios pertinentes a las categorías como en la pantalla referente a los grupos.

También fueron realizados bocetos móviles de estas interfaces para así poder maquetar cómo se haría el decremento de manera responsiva de cada una de las páginas. Se van a exponer de manera breve y menos explicativa que sus homónimas de tamaño mayor.

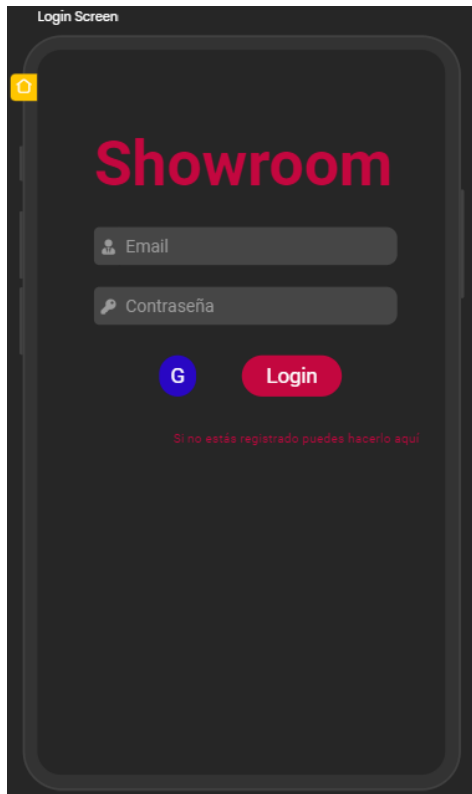


Figura 44: Boceto móvil inicio de sesión

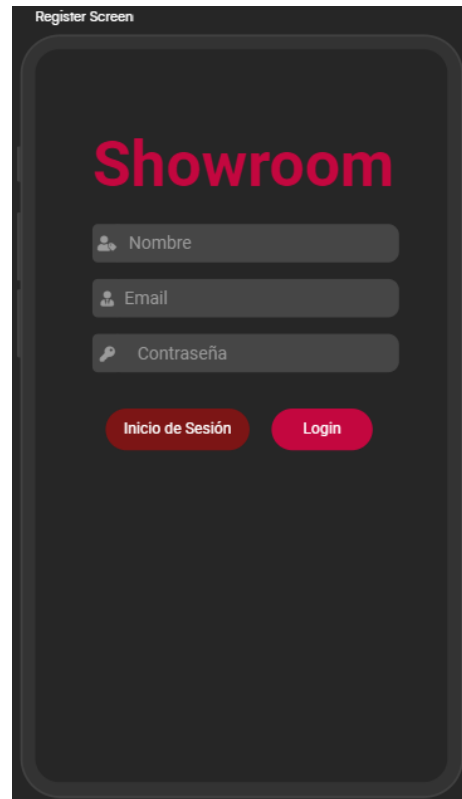


Figura 45: Boceto móvil registro

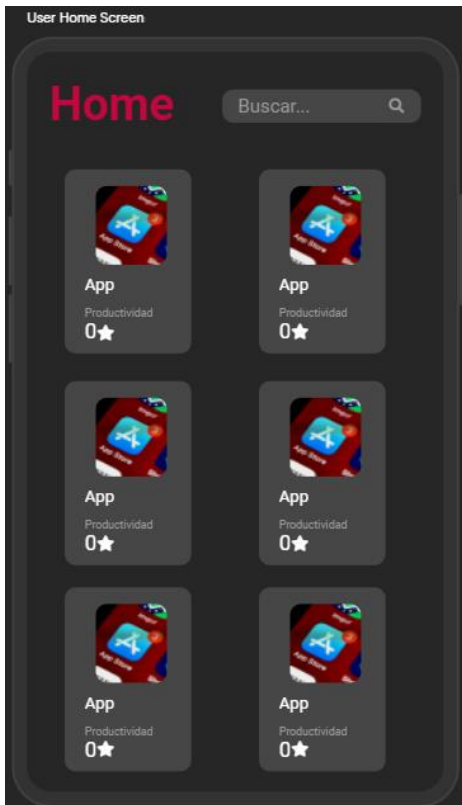


Figura 46: Boceto móvil home usuario

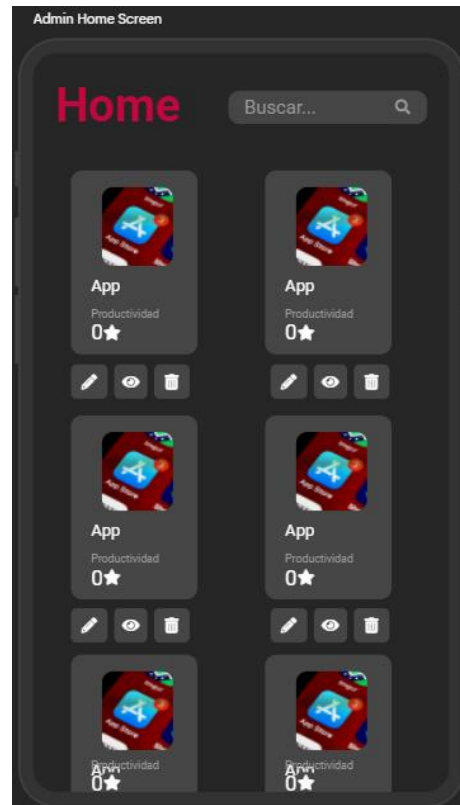


Figura 47: Boceto móvil home administrador



Figura 48: Boceto móvil detalle app usuario



Figura 49: Boceto móvil detalle app administrador



Figura 50: Boceto móvil selección súper administrador

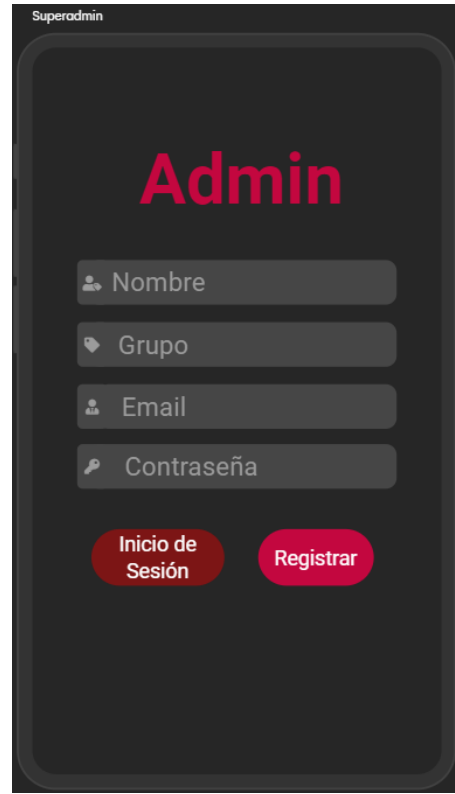


Figura 51: Boceto móvil añadir administrador

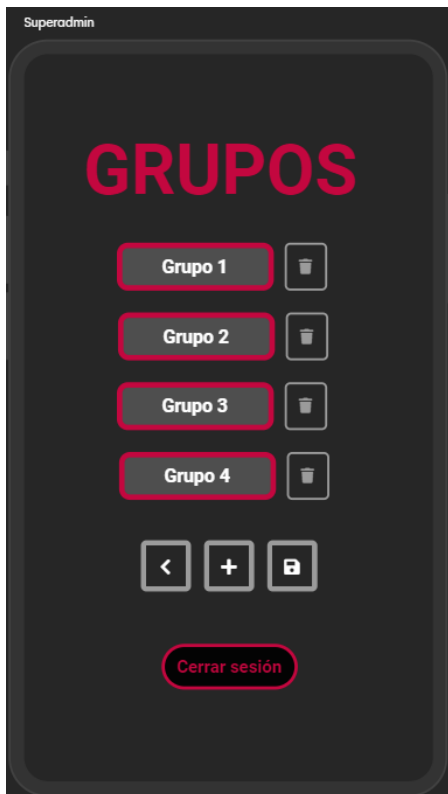


Figura 52: Boceto móvil de edición de grupos



Figura 53: Boceto móvil edición de categorías

6.5 Evaluación de la interfaz

Los diseños de interfaz iniciales fueron enseñados a varias personas. Estas personas tienen entre 22 y 24 años y son antiguos alumnos de la escuela así que su opinión puede acercarse a la de los posibles futuros usuarios de la aplicación. También hay algunos comentarios que fueron brindados por el profesor a la hora de la propia construcción de la aplicación. Los comentarios fueron en su mayoría positivos, aunque hubo una serie de cambios que propusieron:

- Agregar alguna distinción por colores en los botones de edición del usuario administrador. Ya que las acciones están guiadas por iconos únicamente estaría bien que hubiese también códigos por colores (rojo para eliminar...) para estar más seguro de lo que se está haciendo.
- Hacer desplazables las imágenes en la pantalla móvil.
- Poder observar imágenes más de cerca y singularmente.
- Expandir y contraer comentarios. Ya que puede darse la situación de que haya un gran grupo de comentarios que el usuario no quiere leer en ese momento.
- Botón de navegación hacia atrás. Es algo en lo que no había caído en primera instancia debido a los botones de navegación de ordenador y móviles, pero siempre hay que tener en la propia interfaz el botón de regreso.
- Filtrado de aplicaciones para el usuario administrador o súper administrador. De esta forma podría modificar las aplicaciones en el orden que desee y encontrar cuáles de ellas tienen por ejemplo más comentarios o descargas.
- Color amarillo para la estrella de la calificación de la aplicación de manera que destaque más.
- Repetición de contraseña en el registro para que el usuario esté seguro de que ha tecleado la contraseña correctamente. Con esta medida se lograría un aumento de la seguridad a la hora de teclear su contraseña.
- Selección de grupo con menú de despliegue de los disponibles. Esto haría que no hubiera erratas al escribir un nombre o confusión por varios parecidos.

7 Implementación

En esta apartado se van a exponer los distintos pasos seguidos a la hora de la realización del proyecto.

En un principio, tras haber contemplado la idea y las tecnologías que iban a ser utilizadas comencé con la creación del proyecto en Firebase. Al estar en una etapa temprana del proyecto y no haber decidido aún el nombre se puso “Showroom” debido a la naturaleza de la aplicación.

La creación del proyecto Firebase es importante para desde un inicio poder optar a las opciones de conexión a la base de datos en la creación del proyecto angular y las opciones de Authentication (Autenticación) y Storage (Imágenes).

Una vez creado el proyecto en Firebase se pasó a la creación del proyecto Angular en utilizando Visual Studio. La versión utilizada de Angular es la 12. Para la creación del proyecto se utilizará el comando “ng new”.

Una vez creado el proyecto Angular se utilizará la key que proporciona Firebase para conectar ambos proyectos. Esta clave será copiada como una variable dentro del archivo de environment.ts y posteriormente se importará en el archivo module.ts el módulo de AngularFire para inicializar la app con la clave proporcionada desde Firebase.

Cuando los dos proyectos fueron conectados de manera exitosa se pasó a la siguiente fase, la construcción de los componentes los cuales compondrían la aplicación. Estos componentes fueron creados teniendo en cuenta los prerequisites observados y el diseño previo de la web.

Primero se hicieron los correspondientes a la autenticación y registro de los usuarios, para esto se creó el servicio “auth” dónde estarán alojados todos los métodos necesarios para el inicio, registro y abandono de las sesiones de los clientes. Apoyándome en la documentación proporcionada por Firebase y a sus métodos signInWithEmailAndPassword y createUserWithEmailAndPassword construí el primer modelo funcional de registro y autenticación que redirigía dependiendo del rol a los usuarios a diferentes componentes. Para su buen funcionamiento se crearon varios usuarios previos de pruebas a los que se les asignaba diferentes roles en la tabla users en Firestore.

Posteriormente sabiendo que la autenticación funcionaba se pasó a la creación de los componentes relativos al usuario. Lo primero que había que realizar era la pantalla de home dónde se podrían ver las aplicaciones. Es por esto por lo que se crearon aplicaciones de prueba que con la ayuda de del servicio bd creado se consiguió tener la funcionalidad de recoger los datos de las diferentes aplicaciones y repartirlos dependiendo de diferentes categorías. Posteriormente se agregó el sistema de búsqueda y de ampliar el visionado de las diferentes categorías de aplicaciones y solo quedó algún que otro retoque estético respecto al modelado inicial.

Lo siguiente en el calendario de ejecución era la creación del detalle de cada una de las aplicaciones. Una vez pulsada la aplicación el usuario será redirigido a la ventana dónde se hallarán toda la información relativa a la aplicación. En primer lugar, se obtuvo la información que tendría que ser mostrada dentro de dicha pantalla de forma que no hubiera ningún tipo de problemas con ello, para esto también se usó el servicio de base de datos expuesto anteriormente. Una vez traída la información relativa a la aplicación se pasó a la construcción del pop-up con el que el cliente podría calificar la aplicación en cuestión, así como el apartado

dentro del detalle dónde serían vistos dichos comentarios. Para esto se creó un nuevo componente y se apoyó en el selector `ngb-rating` proporcionado por Bootstrap para crear una calificación con sistema de estrellas como viene siendo tendencia en aplicaciones similares. Teniendo los comentarios pudiendo ser sobrescritos y actualizados correctamente se pasó a la creación del apartado de imágenes. Este apartado comenzó siendo algo arcaico en un principio simplemente siendo traídas las imágenes desde el módulo de Storage proporcionado por Firebase con algunas imágenes de prueba. Al necesitar traer las imágenes desde el Storage se creó un nuevo servicio llamado `image` con el que se abordaría este funcionamiento.

Una vez que las funcionalidades del usuario habían sido prácticamente satisfechas se pasó a la parte del administrador. Se comenzó con la pantalla de Home tal y como se había hecho con el cliente. Debido a que la funcionalidad y visionado era similar a la parte del usuario no supuso gran carga de trabajo. Se puso la búsqueda y se añadieron los botones los cuales añadirían las funciones de editar tanto la aplicación como la visibilidad al cliente de esta y su borrado. Se comenzó por el cambio de visibilidad de las aplicaciones ya que era lo más sencillo al solo tener que cambiar un valor booleano en la base de datos.

Posteriormente se creó un componente para detallar y permitir la edición de las aplicaciones. Para esto se tuvieron que crear nuevos métodos de edición en el servicio `bd` para que los clientes pudieran ver los cambios producidos por los administradores. Por último, se añadió la funcionalidad de edición de las imágenes presentes en la aplicación. Con unos botones parecidos a los que se habían puesto en la pantalla home del administrador y nuevos métodos dentro del servicio de `image` se consiguió que las imágenes (incluyendo el logo) pudieran ser manipuladas de forma correcta y salvadas únicamente cuando el administrador pulsara el botón de guardar.

En último lugar tocaba cubrir las pantallas del súper administrador. En primer lugar, se dispuso un menú simple con el fin de poder navegar por ellas. Se creó en primer lugar la pantalla home y el detalle de aplicaciones pues eran los mismos que para el caso de los administradores. Después se pasó a la pantalla para añadir administradores, para ello se dispuso de un formulario de registro para administradores apoyándose nuevamente en el servicio `bd`. Se hicieron pruebas añadiendo nuevos administradores y autenticándose en la pantalla de login. Cuando funcionó de manera correcta se pasó a la creación de los grupos y categorías en la base de datos y a los formularios que tendrían a mano los súper administradores dónde podrían cambiarlas. Se utilizó un sistema similar al utilizado para la edición tanto de aplicaciones como imágenes.

Cuando las funcionalidades de la página estaban prácticamente cubiertas se puso más detalle en la detección de errores y se creó un servicio `error` con el que poder advertir al usuario de los diferentes detalles del error que podría estar teniendo. Para ello se ha incluido un sistema de toast (Angular Toast) mediante la inclusión de la biblioteca lo que avisará mediante etiquetas que serán de diferente color dependiendo de la intención del mensaje al usuario (rojo para error, amarillo para aviso...).

Otro punto de detalle final era la manipulación de imágenes. Se creó un espacio en el que pudieran verse de forma individual pudiendo navegar entre ellas en la parte cliente para así tener una mejor vista de estas. También se añadió la posibilidad de borrar la cuenta a un usuario o administrador.

En última instancia se añadieron pequeños detalles que no habían sido vistos anteriormente o que por decisiones finales tuvieron que ser cambiados y se preparó la web para ser evaluada mediante numerosas pruebas.

8 Pruebas

A la hora de certificar la calidad de un proyecto software es un hábito saludable realizar numerosas pruebas con el fin de evitar posibles erratas que en la fase de desarrollo pueden no ser visibles. Estas pruebas pueden ser de dos tipos:

- Pruebas de caja blanca: centradas en los detalles procedimentales del software con una fuerte ligación al código fuente sirven para comprobar que los procesos y datos sigan los flujos estipulados en su creación.
- Pruebas de caja negra: centradas en apreciar la funcionalidad de un producto software. No se tiene en cuenta el código, datos de su implementación o rutas internas y solo se tienen en cuenta las entradas y sus correspondientes salidas observando que todo funciona según lo planeado.

En este proyecto las pruebas en las que nos enfocaremos serán las de caja negra. Se proporcionarán diferentes situaciones en las que se definirá con antelación el resultado esperado y se comprobará que tras realizar las acciones en el sistema ambos coincidan.

PCN – 001 Iniciar Sesión		
Entrada	Resultado esperado	Resultado obtenido
El cliente previamente registrado introduce sus datos correctamente y pulsa el botón de inicio de sesión.	El cliente es redirigido a la pestaña home de la aplicación.	Correcto
El cliente completa los campos de inicio de sesión sin haberse registrado.	El sistema informa que el cliente no tiene una cuenta activa en el sistema	Correcto
El cliente no rellena el campo de correo y pulsa el botón de inicio de sesión	El sistema informa al usuario que no se han completado todos los campos necesarios para el inicio de sesión.	Correcto
El cliente no rellena el campo de contraseña y pulsa el botón de inicio de sesión	El sistema informa al usuario que no se han completado todos los campos necesarios para el inicio de sesión.	Correcto

Tabla 42: PCN-001

PCN – 002 Cerrar sesión		
Entrada	Resultado esperado	Resultado obtenido
El cliente previamente identificado pulsa el botón de cierre de sesión.	El sistema lo redirige a la pantalla de inicio de sesión impidiendo que este vuelva a entrar sin que se identifique de nuevo	Correcto

Tabla 43: PCN-002

PCN – 003 Registrar usuario		
Entrada	Resultado esperado	Resultado obtenido
El cliente introduce sus datos de forma correcta sin tener coincidencia con el correo de otro usuario	El cliente se registra en el sistema y se le redirige a la pantalla Home de cliente.	Correcto
El cliente no completa el campo de nombre.	El sistema avisa al cliente de que faltan campos por rellenar.	Correcto
El cliente no completa el campo de correo.	El sistema avisa al cliente de que faltan campos por rellenar.	Correcto
El cliente no completa el campo contraseña.	El sistema avisa al cliente de que faltan campos por rellenar.	Correcto
El cliente no completa el campo de repetición de contraseña.	El sistema avisa al cliente de que faltan campos por rellenar.	Correcto
El cliente escribe un correo que no sigue las pautas internacionales de correos.	El sistema avisa al cliente de que el formato del correo no es válido	Correcto
El cliente no escribe correctamente la repetición de la contraseña.	El sistema avisa al cliente de que las contraseñas introducidas no se corresponden.	Correcto
El cliente escribe una contraseña menor a 6 caracteres.	El sistema avisa al cliente de que la contraseña tiene que ser de entre 5 y 14 caracteres.	Correcto
El cliente escribe el correo de un usuario ya existente en el sistema.	El sistema avisa al cliente de que el usuario está en uso.	Correcto
El cliente presiona el icono de Google y selecciona una cuenta para el inicio de sesión.	El sistema registra la cuenta si es que no lo estaba anteriormente y lo redirige a la pantalla home correspondiente.	Correcto

Tabla 44: PCN-003

PCN – 004 Borrar cuenta		
Entrada	Resultado esperado	Resultado obtenido
El cliente previamente registrado pulsa el botón de borrar cuenta.	El sistema informa al usuario y borra los datos identificativos del usuario rediriéndole al inicio de sesión.	Correcto

Tabla 45: PCN-004

PCN – 005 Editar aplicación		
Entrada	Resultado esperado	Resultado obtenido
El administrador modifica el nombre/categoría/enlace/descripción de la aplicación de forma correcta y pulsa el botón de guardar.	El sistema guarda los cambios y manda al administrador a la pantalla home informándole de que los cambios han sido guardados de forma correcta.	Correcto
El administrador no completa algún campo y pulsa el botón de guardar.	El sistema avisa al administrador que los campos no están rellenos de forma adecuada.	Correcto
El administrador cambia el logo de la aplicación y pulsa el botón de guardado.	El sistema guarda el nuevo logo de la aplicación y redirige al administrador a la pantalla home informándolo de que el cambio ha sido exitoso.	Correcto
El administrador añade o borra las imágenes correspondientes a la aplicación.	El sistema guarda las nuevas imágenes asociadas a la aplicación e informa al usuario de que el cambio ha sido efectuado de manera exitosa.	Correcto
El administrador modifica la visibilidad de algunas de las imágenes de la aplicación.	La visibilidad de las imágenes cambia según lo deseado por el administrador. El administrador es redirigido a la pantalla home y es informado de que los cambios han sido un éxito.	Correcto

Tabla 46: PCN-005

PCN – 006 Añadir aplicación		
Entrada	Resultado esperado	Resultado obtenido
El administrador introduce el logo, nombre, categoría, enlace, descripción de la aplicación de forma correcta y pulsa el botón de guardar.	El sistema guarda la nueva aplicación añadiéndola a su catálogo de aplicaciones con modificaciones, redirige al administrador a la pantalla home y lo informa de que la aplicación se ha añadido con éxito.	Correcto
El administrador introduce imágenes adicionales.	El sistema guarda estas imágenes asociadas a la aplicación.	Correcto
El administrador deja en blanco alguno de los campos obligatorios.	El sistema informa al administrador que la operación no está disponible si no están rellenos todos los campos necesarios.	Correcto

Tabla 47: PCN-006

PCN – 007 Borrar aplicación		
Entrada	Resultado esperado	Resultado obtenido
El administrador presiona el botón de borrado de la aplicación y confirma su borrado.	El sistema borra la aplicación e informa al administrador que el borrado fue exitoso.	Correcto
El administrador presiona el botón de borrado de aplicación, pero no confirma su borrado	El sistema no borra la aplicación y cierra el pop up.	Correcto

Tabla 48: PCN-007

PCN – 008 Modificar visibilidad de aplicación		
Entrada	Resultado esperado	Resultado obtenido
El administrador pulsa el icono de visibilidad de una aplicación.	El sistema modifica la visibilidad de la aplicación de cara a los clientes en consecuencia al cambio e informa al administrador del éxito del cambio.	Correcto

Tabla 49: PCN-008

PCN – 009 Añadir administrador		
Entrada	Resultado esperado	Resultado obtenido
El súper administrador introduce los datos referentes al nuevo administrador y selecciona un grupo al que administrará las aplicaciones correspondientes. Cuando se han introducido los datos pulsa el botón de guardado.	El sistema guarda al nuevo administrador e informa al súper administrador de que la operación fue un éxito.	Correcto
El súper administrador deja en blanco algún campo del registro y pulsa el botón de guardado.	El sistema avisa al súper usuario de que no es posible añadir al administrador porque no están todos los datos completos.	Correcto
El súper administrador introduce un correo no válido y pulsa el botón de guardado.	El sistema informa al usuario de que el correo introducido no es válido.	Correcto

Tabla 50: PCN-009

PCN – 010 Editar grupos		
Entrada	Resultado esperado	Resultado obtenido
El súper administrador pulsa el botón de añadir, escribe el grupo o grupos que desee añadir y pulsa el botón de guardado.	El sistema guarda los grupos nuevos en el sistema y despliega la lista con ellos añadidos.	Correcto
El súper administrador pulsa el botón de borrado de algún grupo y guarda el resultado.	El sistema actualiza la lista sin los grupos borrados por el súper administrador.	Correcto
El súper administrador edita el nombre de un grupo y pulsa el botón de guardado.	El sistema actualiza la lista con el nombre del grupo actualizado.	Correcto
El súper administrador pulsa el botón de añadir grupo y deja el campo vacío.	El sistema avisa al súper administrador de que no pueden guardarse campos vacíos.	Correcto
El súper administrador edita, añade o borra cualquiera de los grupos y no selecciona el botón de guardado.	El sistema no efectúa ninguna operación sobre la base de datos.	Correcto

Tabla 51: PCN-010

PCN – 011 Editar categorías		
Entrada	Resultado esperado	Resultado obtenido
El súper administrador pulsa el botón de añadir, escribe el nombre de las nuevas categorías que desee añadir y pulsa el botón de guardado.	El sistema guarda las nuevas categorías en el sistema y despliega la lista con ellos añadidos.	Correcto
El súper administrador pulsa el botón de borrado de alguna categoría y guarda el resultado.	El sistema actualiza la lista sin las categorías borradas por el súper administrador.	Correcto
El súper administrador edita el nombre de una categoría y pulsa el botón de guardado.	El sistema actualiza la lista con el nombre de la categoría actualizada.	Correcto
El súper administrador pulsa el botón de añadir categoría y deja el campo vacío.	El sistema avisa al súper administrador de que no pueden guardarse campos vacíos.	Correcto
El súper administrador edita, añade o borra cualquiera de las categorías y no selecciona el botón de guardado.	El sistema no efectúa ninguna operación sobre la base de datos.	Correcto

Tabla 52: PCN-011

PCN – 012 Comentar aplicación		
Entrada	Resultado esperado	Resultado obtenido
El cliente pulsa el botón de calificar, da una puntuación y opinión sobre la aplicación y pulsa el botón de guardado.	El sistema guarda el nuevo comentario y lo hace visible para todo el mundo que visite la aplicación en detalle. También se actualiza la calificación media de la aplicación.	Correcto
El cliente no completa el campo de título del comentario.	El sistema no permite al cliente guardar el comentario.	Correcto
El cliente no completa el campo de descripción del comentario.	El sistema no permite al cliente guardar el comentario.	Correcto
El cliente completa los campos obligatorios del comentario y pulsa volver.	El sistema no guarda la información sobre el comentario.	Correcto
El cliente recalifica una aplicación dónde ya tenía un comentario y pulsa el botón de guardado.	El sistema actualiza el comentario del cliente y los valores medios de calificación acorde a los del nuevo comentario.	Correcto

Tabla 53: PCN-012

PCN – 013 Ver imágenes en detalle		
Entrada	Resultado esperado	Resultado obtenido
El cliente pulsa sobre cualquier imagen del carrusel.	El sistema ofrece una vista detallada de la imagen y permite al cliente navegar por las demás imágenes mediante las flechas laterales.	Correcto

Tabla 54: PCN-013

9 Despliegue

Una vez revisadas todas las funcionalidades pertinentes a la web a partir de las pruebas establecidas hay que desplegar la aplicación sobre el entorno web correspondiente y asegurarse de que todo sigue funcionando de la manera correcta.

El entorno web dónde se subirá esta web será en la página `onapp.infor.uva.es` que es la creada por Joaquín para la feria OnApp Experience dónde se expone dicha feria y las aplicaciones del alumnado. La página de este proyecto indexaría en la ruta `showroom` de esta página y se correspondería con el punto de inicio de sesión de esta.

Para que esto sea posible en primer lugar hay que construir la página. Para esto angular tiene un comando `ng build` el cuál ayuda mucho a realizar esta tarea. Con este comando los archivos resultantes de la construcción de la página irán a parar a un directorio `dist` en el cuál simplemente se tiene que copiar su contenido sobre el servidor correspondiente para su funcionamiento.

El servidor al que se va a subir la página es un servidor Apache con versión 2.4.41. El proceso se realizó tal y como se ha detallado. Teniendo los archivos en el directorio `dist` se copiaron a la ruta relativa al espacio `showroom` del servidor. Una vez subidos ya se podía navegar hasta la web `showroom` del proyecto con la url `onapp.infor.uva.es`. El único problema por resolver era el enrutamiento, debido a que angular tiene enrutamiento desde cliente, cuando el servidor intenta recargar una página que no es la principal y a la que apunta el `index.html` subido resuelve con un error 404.

Este problema tiene dos posibles soluciones. La primera era la inclusión de un fichero `.htaccess` que redirigiría el control del enrutamiento sobre el cliente a la hora de hacer una petición al servidor. Para esto se creó un fichero de configuración y se intentó añadir al servidor tras hacer unos cuantos ajustes. Como tras hacer estos cambios el enrutamiento seguía fallando se optó por la segunda opción. Un método muy utilizado a la hora de desplegar una aplicación web angular sobre un servidor es utilizar `Hash Location Strategy`. Con esta herramienta con soporte en el módulo de enrutamiento de angular y la ayuda de la almohadilla previa brinda al cliente la posibilidad de redirigir la página hasta su destino fijado sin que el servidor deba tener ningún tipo de información.

Una vez hecho esto la aplicación web era totalmente funcional y operativa en el servidor Apache y podía ser localizada mediante la URL asignada (`onapp.infor.uva.es/showroom`).

10 Conclusiones y posibles mejoras

En este apartado se echará un vistazo al resultado obtenido en el proyecto, se analizarán el producto obtenido y se expondrán algunas mejoras que podrían añadirse si no hubiera límites temporales.

10.1 Conclusiones

Para la finalización de este proyecto se ha obtenido una web funcional para la administración y visualización de aplicaciones. Si bien es cierto que conocía la herramienta de trabajo y que tuve un periodo inicial de adaptación la conexión con Firebase como base de datos y los diferentes tratamientos de información que esto supone me han supuesto una numerosa cantidad de problemas a lo largo de todo el desarrollo, aunque también es cierto que hay muchos apartados como el tema de la autenticación o las imágenes dónde es mucho más sencillo que habiéndolo hecho con alguna otra tecnología.

Dejando aspectos técnicos aparte, este proyecto me ha seguido por varios meses y, a pesar de que haya habido momentos negativos estoy bastante contento con el resultado. De cara a mi formación este proyecto me ha ayudado mucho a darme cuenta de que el desarrollo web es algo con lo que disfruto y espero estar relacionado en un futuro laboral.

Para completar la conclusión, este es el proyecto más grande que había afrontado en toda la carrera y ver como de algo que eran puras ideas y requisitos iniciales con el esfuerzo propio ha cobrado vida y se ha hecho funcional. Creo que el esfuerzo invertido y los conocimientos adquiridos han valido totalmente la pena.

10.2 Posibles mejoras

Si bien es cierto que este es un proyecto cerrado y específico siempre son posibles las mejoras que hagan que su funcionalidad e interacción con el usuario den un paso adelante. Aquí se listarán algunas de estas posibles mejoras que podrían darse:

- **Notificaciones:** Teniendo en cuenta que se conoce el correo de cada uno de los administradores sería posible activar una opción de notificar a cada uno de estos si sucede un evento en una de las aplicaciones que supervisan ya sea una descarga o comentario.
- **Acceso sin registro:** Se podría implementar un acceso sin registro para cualquier usuario. Este usuario no tendría el poder de comentar ya que no se le asociaría a ningún identificador o podría hacerlo de forma anónima.
- **Introducción de más aplicaciones:** De momento el número de aplicaciones es muy limitado y las herramientas como la de búsqueda por nombre quedan un poco inútiles. Cobrarían mucha más importancia si el número de aplicaciones fuera más elevado.
- **Manipulación de imágenes:** Sería más cómodo para el usuario si a la hora de ver las imágenes en profundidad pudiera hacer zoom en la imagen de alguna manera si

quisiera observar con más detalle algún punto. También estaría bien que a la hora de añadir una aplicación y su logo este pudiera ser recortado de forma que el administrador tuviera control absoluto del tamaño y ubicación del logo añadido.

- **Aumentar el nivel de detalle:** Se podría añadir más información sobre las aplicaciones como información sobre su autor, su fecha de realización... De esta forma se aportaría más información al usuario y eso siempre es bienvenido.

11 Bibliografía

- [1] What is Angular? [En línea] Último acceso: 12/03/2022
Disponible en: <https://angular.io/guide/what-is-angular>

- [2] Qué es Angular [En línea] Último acceso: 12/03/2022
Disponible en: <https://openwebinars.net/blog/que-es-angular-2021/>
- [3] What is Firebase? The complete story, abridged. [En línea] Último acceso: 14/03/2022
Disponible en: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- [4] Acerca de Astah UML [En línea] Último acceso: 14/03/2022
Disponible en: <https://www.componentsource.com/es/product/astah-uml/about>
- [5] Google Play Store [En línea] Último acceso: 16/03/2022
Disponible en: <https://neoattack.com/neowiki/google-play-store/>
- [6] Microsoft Store [En línea] Último acceso: 16/03/2022
Disponible en: https://es.wikipedia.org/wiki/Microsoft_Store
- [7] ¿Qué es el App Store? [En línea] Último acceso: 17/03/2022
Disponible en: <https://www.qode.pro/blog/que-es-el-app-store/>
- [8] Aptoide [En línea] Último acceso: 17/03/2022
Disponible en: <https://es.wikipedia.org/wiki/Aptoide>
- [9] Regex (direcciones de correo electrónico) [En línea] Último acceso: 04/04/2022
Disponible en: <http://w3.unpocodetodo.info/utiles/regex-ejemplos.php?type=email>
- [10] Angular 13 Email validation example [En línea] Último acceso: 04/04/2022
Disponible en: <https://www.cloudhadoop.com/angular-email-validation-example/>
- [11] How to “show more” rows using *ngFor [En línea] Último acceso: 04/04/2022
Disponible en: <https://stackoverflow.com/questions/59329628/how-to-show-more-rows-using-ngfor/59329756#59329756>
- [12] RegExp [En línea] Último acceso: 04/04/2022
Disponible en:
https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/RegExp
- [13] Gestión de riesgos en proyectos de software [En línea] Último acceso: 07/04/2022
Disponible en: <https://www.piranirisk.com/es/blog/gestion-de-riesgos-proyectos-de-software>
- [14] Diseño centrado en el Usuario [En línea] Último acceso: 10/04/2022
Disponible en: <https://neoattack.com/neowiki/diseño-centrado-en-el-usuario/#:~:text=El%20Dise%C3%B1o%20centrado%20en%20el%20usuario%20es%20una,grado%20de%20satisfacci%C3%B3n%20por%20parte%20de%20los%20consumidores.>

- [15] Router Link [En línea] Último acceso: 11/04/2022
Disponible en: <https://angular.io/api/router/RouterLink>
- [16] Angular User Rating Component Using 'NgBootstrap' Library [En línea] Último acceso: 12/04/2022
Disponible en: <https://www.learnmoreseekmore.com/2021/01/angular-user-rating-component-using-NgBootstrap.html>
- [17] Proceso Racional Unificado [En línea] Último acceso: 11/04/2022
Disponible en: https://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
- [18] Arquitectura en Angular [En línea] Último acceso: 12/04/2022
Disponible en: <https://dev.to/vanessamarely/arquitectura-en-angular-5c23>
- [19] Angular: Arquitectura del Framework [En línea] Último acceso: 12/04/2022
Disponible en: <https://medium.com/angular-chile/angular-arquitectura-del-framework-e46204f38fef>
- [20] Miguel Ángel Laguna, Félix Prieto. Apuntes de la asignatura "Fundamentos de Ingeniería Software" [En línea] Último acceso: 14/04/2022
- [21] Full Angular 13 Firebase Authentication Tutorial Example [En línea] Último acceso: 30/04/2022
Disponible en: <https://www.positronx.io/full-angular-7-firebase-authentication-system/>
- [22] Autentica con Firebase mediante cuentas basadas en contraseñas con JavaScript [En línea] Último acceso: 03/05/2022
Disponible en: <https://firebase.google.com/docs/auth/web/password-auth>
- [23] Autentica mediante Google con JavaScript [En línea] Último acceso: 03/05/2022
Disponible en: <https://firebase.google.com/docs/auth/web/google-signin>
- [24] Administra usuarios en Firebase [En línea] Último acceso: 05/05/2022
Disponible en: <https://firebase.google.com/docs/auth/web/manage-users>
- [25] Agrega datos a Cloud Firestore [En línea] Último acceso: 10/05/2022
Disponible en: <https://firebase.google.com/docs/firestore/manage-data/add-data>
- [26] Borra datos de Cloud Firestore [En línea] Último acceso: 12/05/2022
Disponible en: <https://firebase.google.com/docs/firestore/manage-data/delete-data>
- [27] Realiza consultas simples y compuestas en Cloud Firestore [En línea] Último acceso: 15/05/2022
Disponible en: <https://firebase.google.com/docs/firestore/query-data/queries>
- [28] Sube archivos con Cloud Storage en la Web [En línea] Último acceso: 15/05/2022
Disponible en: <https://firebase.google.com/docs/storage/web/upload-files?hl=es-419>

- [29] Font Awesome [En línea] Último acceso: 17/05/2022
Disponible en: <https://fontawesome.com/>
- [30] Ordena y limita datos con Cloud Firestore [En línea] Último acceso: 20/05/2022
Disponible en: <https://firebase.google.com/docs/firestore/query-data/order-limit-data>
- [31] Learn Angular: Modal Pop Up | Angular Material | With Source Code [En línea] Último acceso: 26/05/2022
Disponible en: <https://www.youtube.com/watch?v=FThtv9iorao>
- [32] Material Dialog Overview [En línea] Último acceso: 28/05/2022
Disponible en: <https://material.angular.io/components/dialog/overview>
- [33] Angular Deployment [En línea] Último acceso: 30/05/2022
Disponible en: <https://angular.io/guide/deployment>

12 ANEXOS

12.1 Manual de usuario

En esta sección se analizará con detalle la web exponiendo las acciones que pueden ejercer cualquier tipo de usuario.

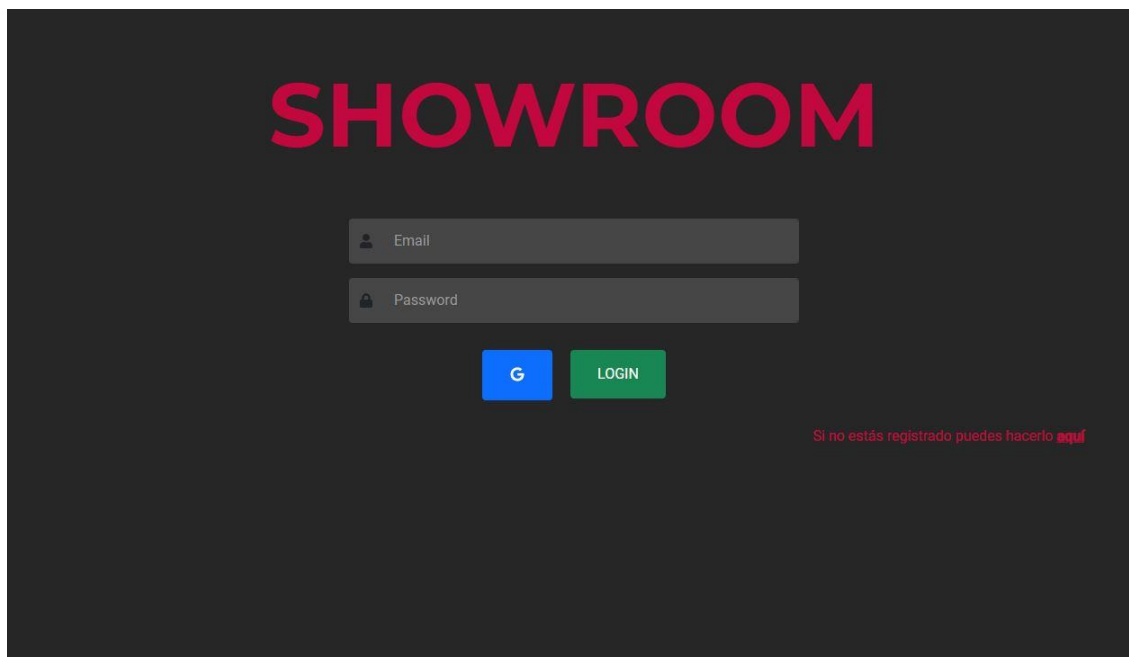


Figura 54: MU – Pantalla de inicio de sesión

Una vez estemos en el Inicio de sesión puede haber varias posibilidades. La primera es que nos hayamos registrado previamente la cual nos permitirá poner nuestras credenciales y nos llevará a la pantalla principal que tenga asignada el rol al que pertenezcamos. Otra opción es iniciar sesión con Google. En este caso será indiferente que sea la primera vez o no puesto que si un usuario no está registrado lo hará automáticamente la primera vez que ingrese en la página y se le asignará el rol de usuario lo que hace que sea la opción más cómoda para su uso.

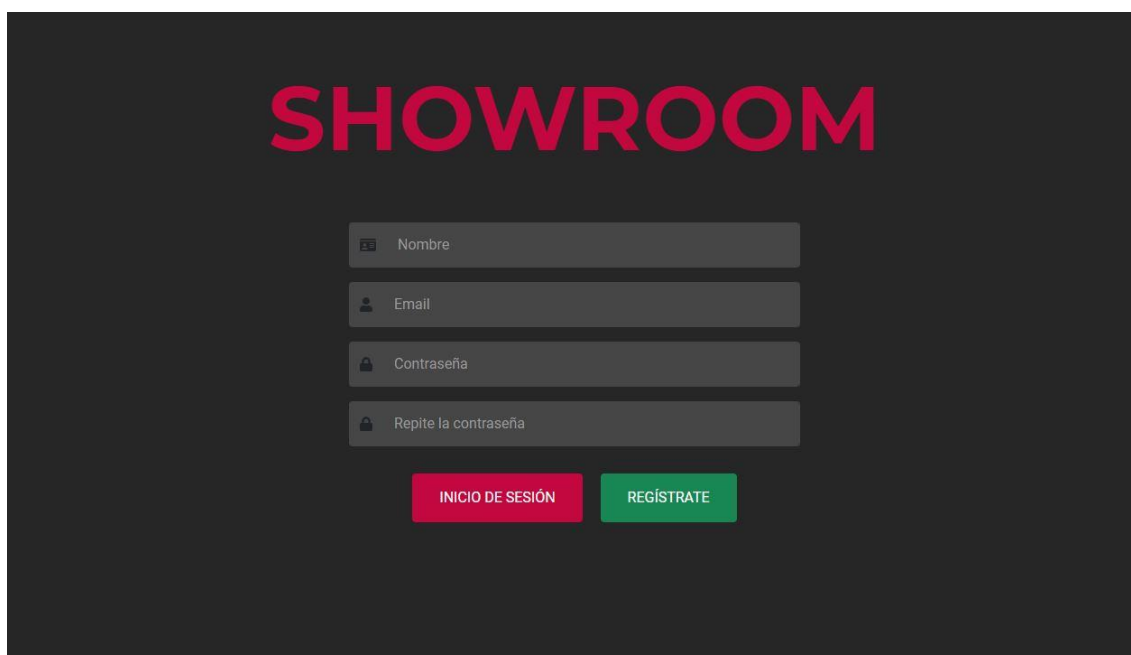


Figura 55: MU – Pantalla de registro de usuario

En caso de que el usuario se quiera registrar habrá pulsado en el botón de registro en la primera pantalla y será redirigido a esta en la que podrá registrarse añadiendo nombre, correo y contraseña. Si estos apartados son escritos de manera correcta el usuario será registrado y accederá directamente a la pantalla de Home de usuario en la cuál son expuestas todas las aplicaciones visibles.

Aquí es dónde bifurcaremos los tres tipos de uso de la aplicación en tres puntos de manera que sea más sencillo de observar.

12.2 Manual de usuario

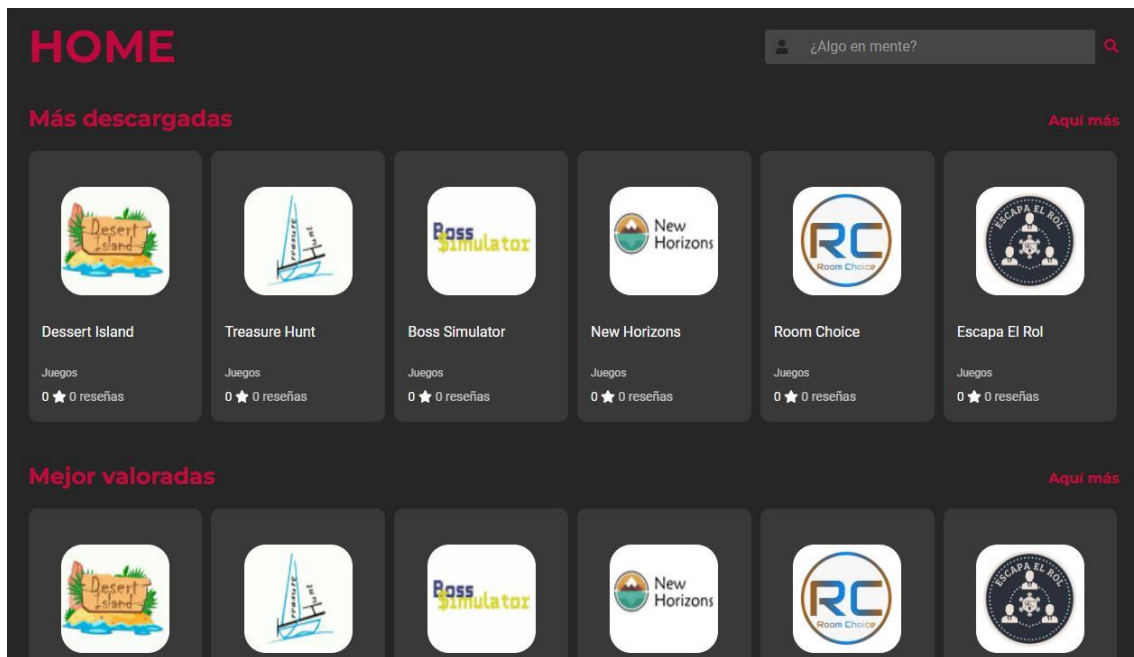


Figura 56: MU - Pantalla home de usuario

Al iniciar sesión como usuario nos aparecerá el letrero de HOME lo que indicará que estamos en la pantalla principal. Aquí podremos navegar de forma vertical por las diferentes categorías. Si se quieren ver todas las disponibles en cierta categoría solo habría que dar al botón alineado con su nombre en la parte derecha en el que pone “Aquí más”. También son posibles búsquedas por nombre que serían efectuadas o bien con el botón intro o con la lupa anexa al campo de búsqueda. Si queremos ver en detalle una aplicación tendríamos que pulsar en el recuadro de esta.

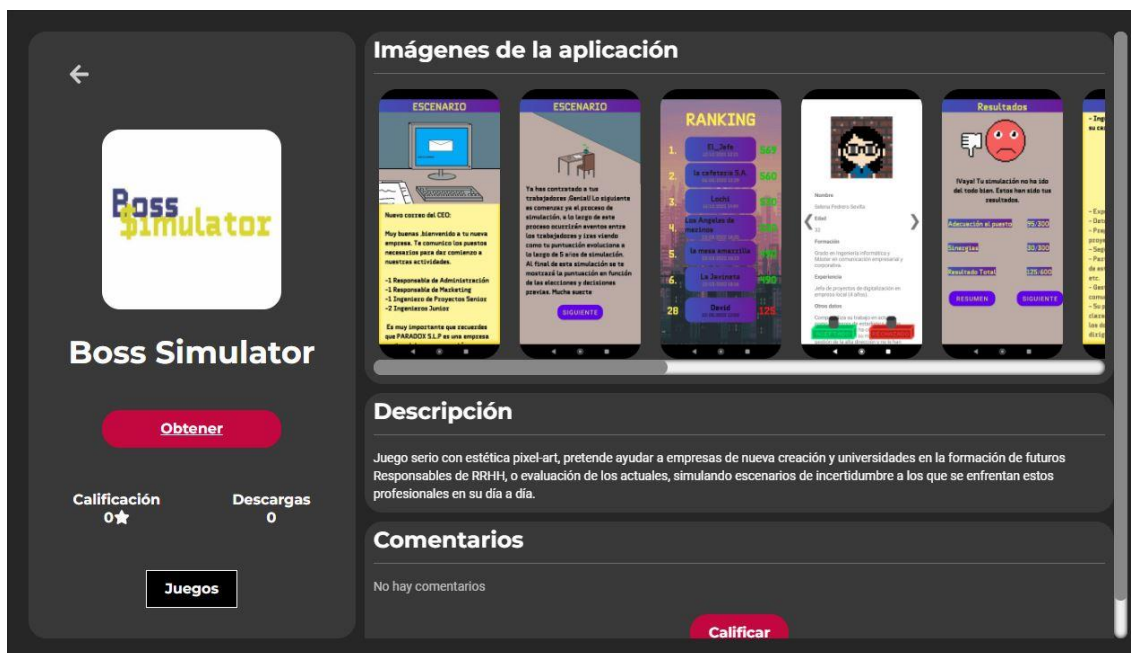


Figura 57: MU - Pantalla de detalla de aplicación de usuario

Al ver los detalles de la aplicación observamos que en la columna izquierda se nos brinda la información principal de la aplicación como logo y nombre y sus calificaciones y descargas. En la parte superior izquierda se mostrarán las imágenes de la aplicación. Inferiormente aparecerá la descripción breve de la aplicación y por último la sección de comentarios ordenados por fecha.

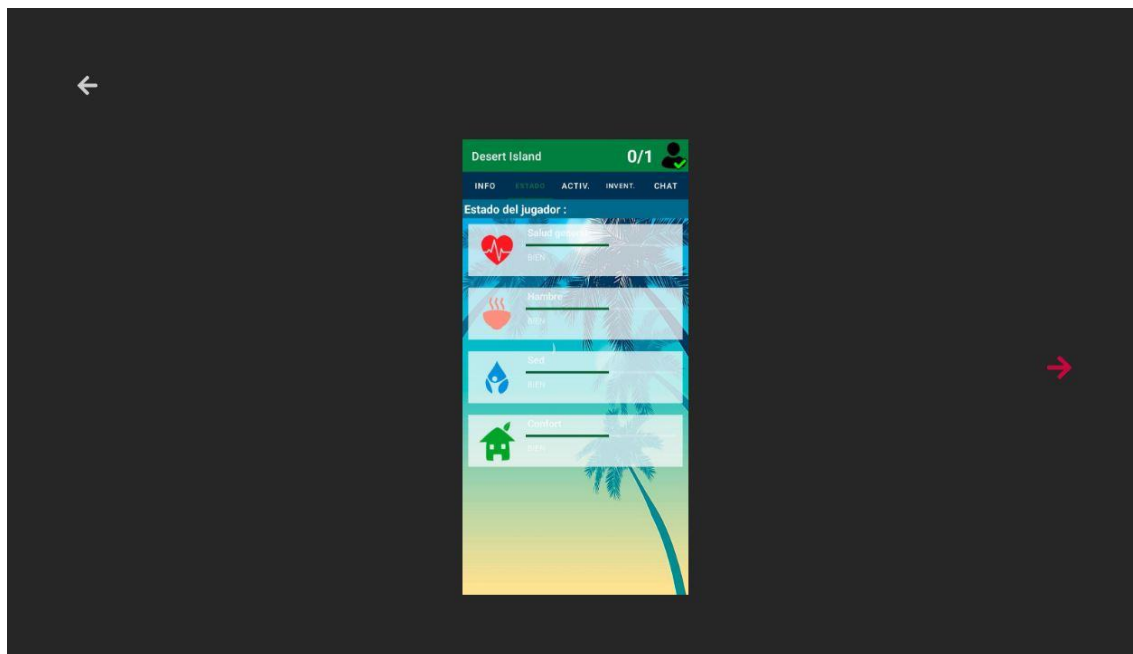


Figura 58: MU - Pantalla de detalle de imagen

Esta pantalla es el resultado de pulsar una imagen del carrusel de imágenes del detalle de aplicación. Se mostrarán las imágenes a un tamaño ampliado y habrá dos flechas a los lados que nos permitirán navegar entre la primera y la última.

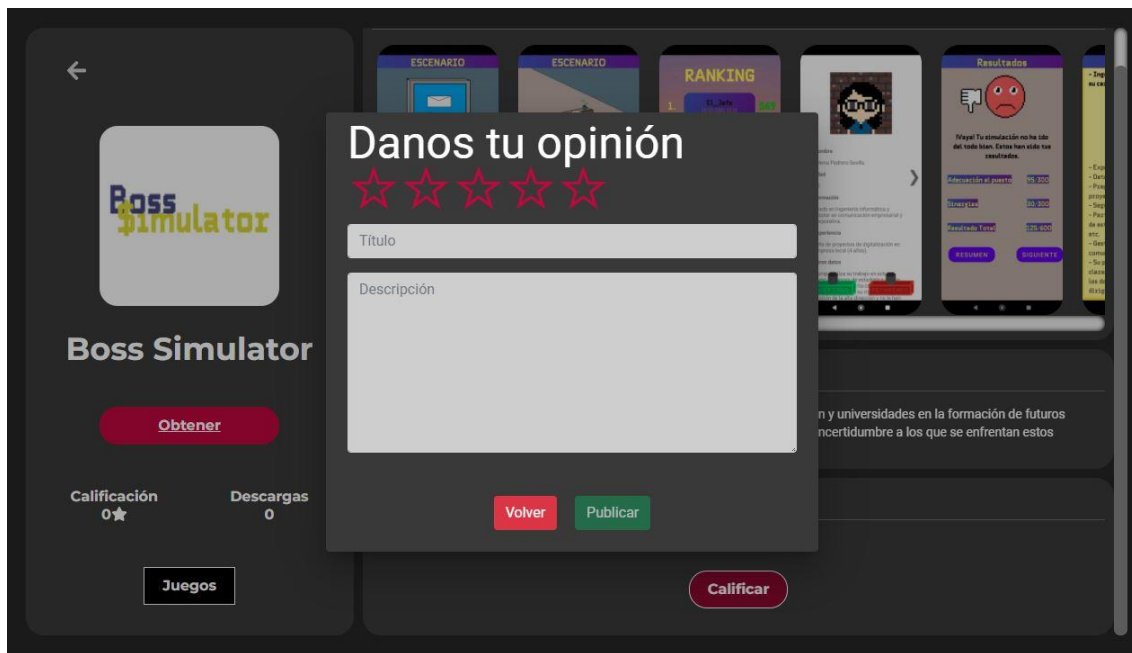


Figura 59: MU - Pantalla de comentario

Si queremos calificar una aplicación tendremos que pulsar el botón inferior del detalle de aplicación “Calificar”. Una vez aquí podremos tanto poner una nueva opinión como renovar una previa. Las estrellas son sinónimos de la puntuación dada y se rellenarían con un click en la estrella equivalente a nuestra puntuación entre 0 y 5. Cuando los campos de título y descripción están rellenos podríamos publicar el comentario lo que actualizaría de forma automática la aplicación.

12.3 Manual de administrador

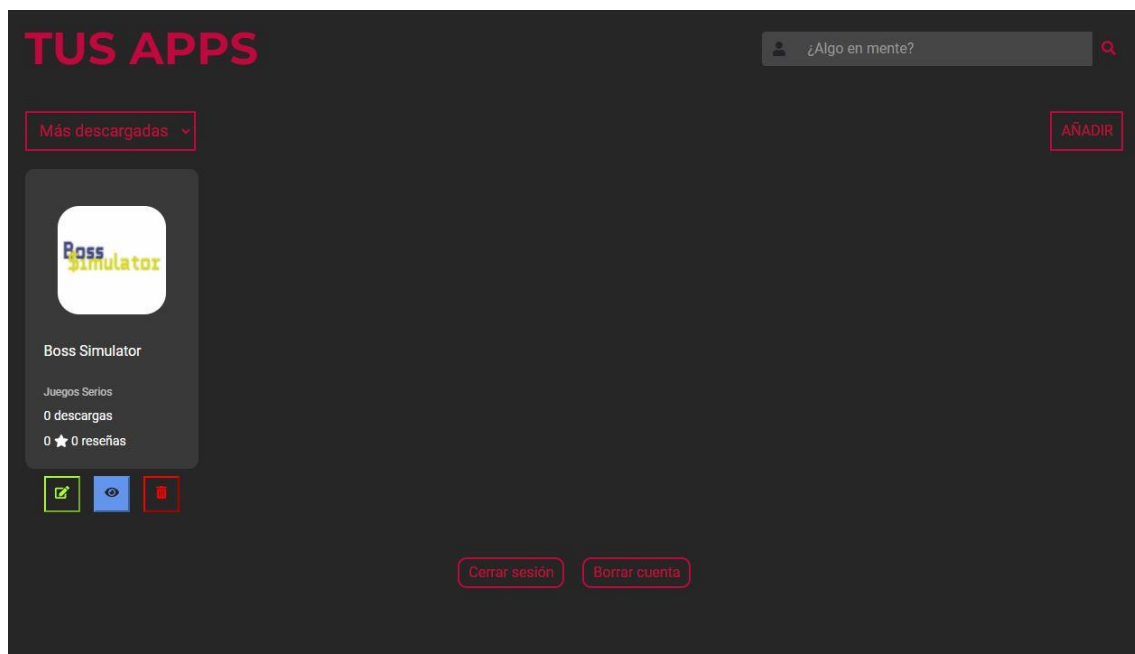


Figura 60: MU - Pantalla home de administrador

En la pantalla principal del administrador tendrá las aplicaciones listadas a las que tiene acceso dependiendo del grupo. Al igual que en la pantalla del usuario tiene las búsqueda por nombre. También puede listar las aplicaciones disponibles con varios filtros situados en el menú desplegable en la parte superior izquierda. Todas las aplicaciones constan de tres botones en su parte inferior que citados de izquierda a derecha son: edición, modificación de visibilidad y borrado. En la parte inferior aparecerán dos botones para cerrar sesión y otro para borrar la cuenta del administrador.



Figura 61: MU - Pantalla de detalle de aplicación de administrador

Al pulsar el botón de edición de la aplicación aparecerán los datos de la aplicación con campos editables. El logo y las imágenes son también editables. La categoría de la aplicación se mostrará en el desplegable dónde estarán todas aquellas que haya añadido el súper administrador. Cada imagen contará con dos botones en su parte inferior para alternar la visibilidad o borrar. Cuando el administrador de por buenos los cambios efectuados guardará y el sistema renovará los datos de la aplicación.

12.4 Manual de súper administrador



Figura 62: MU - Pantalla de selección de súper administrador

Una vez registrado el súper administrador tendrá a su disposición un menú que separará las distintas funciones que puede desempeñar en la aplicación. El botón de Cerrar Sesión estará en la parte inferior.

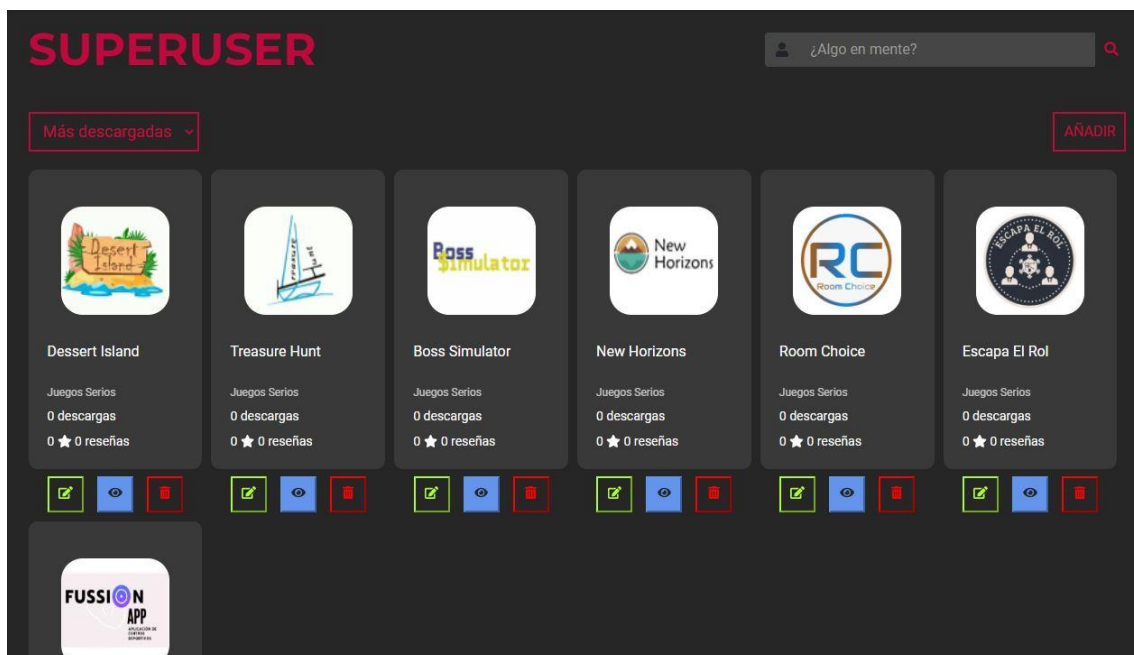



Figura 63: MU - Pantalla home de súper administrador

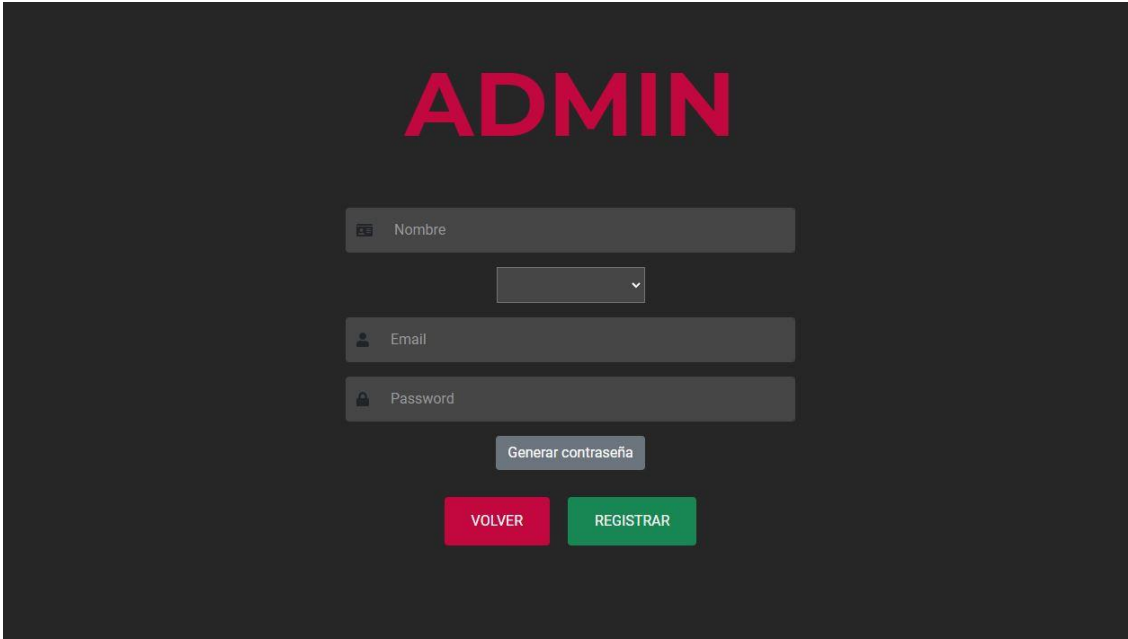
Al pulsar el botón de Home se mostrará una pantalla en la que la funcionalidad es igual a la pantalla del administrador. La única diferencia es que el súper administrador tendrá a su disponibilidad todas las aplicaciones del sistema.



The screenshot shows a dark-themed mobile application interface. On the left, there is a square button with a red plus sign. To the right, there is a form with the following fields: 'Nombre de aplicación' (Application Name) with a text input field containing the placeholder 'Nombre'; 'Categoría' (Category) with a dropdown menu; 'Grupo' (Group) with a dropdown menu; 'Enlace' (Link) with a text input field; and 'Descripción' (Description) with a larger text input field. A green 'GUARDAR' (Save) button is located to the right of the 'Grupo' field. A back arrow icon is in the top left corner.

Figura 64: MU - Pantalla de detalle de aplicación de súper administrador

Cuando añade o edita una aplicación la única diferencia con la pantalla del administrador es que tiene una referencia al grupo al que pertenece esta aplicación. De esta manera cuando añade una aplicación en el sistema siempre tiene que ir adherida a un grupo que pueda moderarla.



The screenshot shows a dark-themed mobile application interface for adding administrators. At the top, the word 'ADMIN' is displayed in large, bold, red letters. Below it, there is a form with the following fields: 'Nombre' (Name) with a text input field; a dropdown menu; 'Email' with a text input field; 'Password' with a text input field; and a 'Generar contraseña' (Generate password) button. At the bottom, there are two buttons: 'VOLVER' (Back) in red and 'REGISTRAR' (Register) in green.

Figura 65: MU - Pantalla de añadir administradores

Al pulsar el botón de gestor se entra en la pantalla de adición de administradores. El súper administrador deberá de introducir el nombre, grupo al que estará asociado y, por tanto, las aplicaciones que moderará serán las pertenecientes a dicho grupo, su correo y contraseña. Para facilitar la tarea de atribuir contraseñas se ha puesto un generador de contraseñas válidas en el sistema. Una vez estén los datos de forma correcta solo tendrá que pulsar el botón de registrar para que el administrador sea añadido de forma instantánea al sistema.

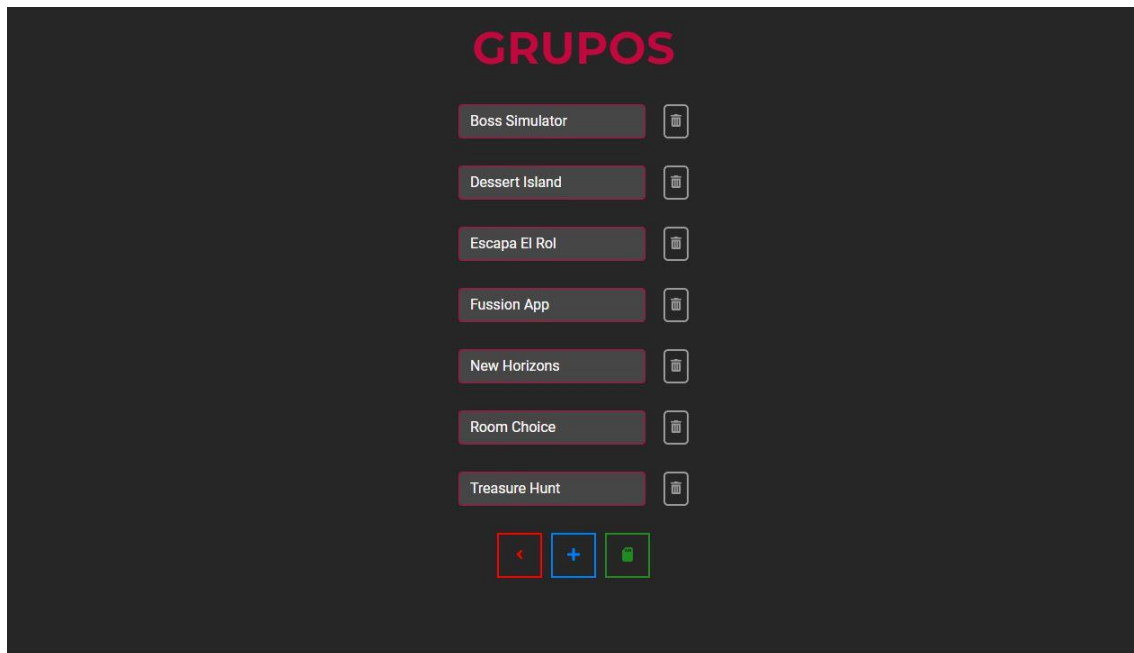


Figura 66: MU - Pantalla de edición de grupos

Al pulsar el botón de grupos el súper administrador entra en esta pantalla dónde puede borrar, añadir y editar los grupos disponibles. Los cambios solo serán guardados si se pulsa el botón con el símbolo de guardado. Los cambios estarán disponibles automáticamente a la hora de añadir administradores.

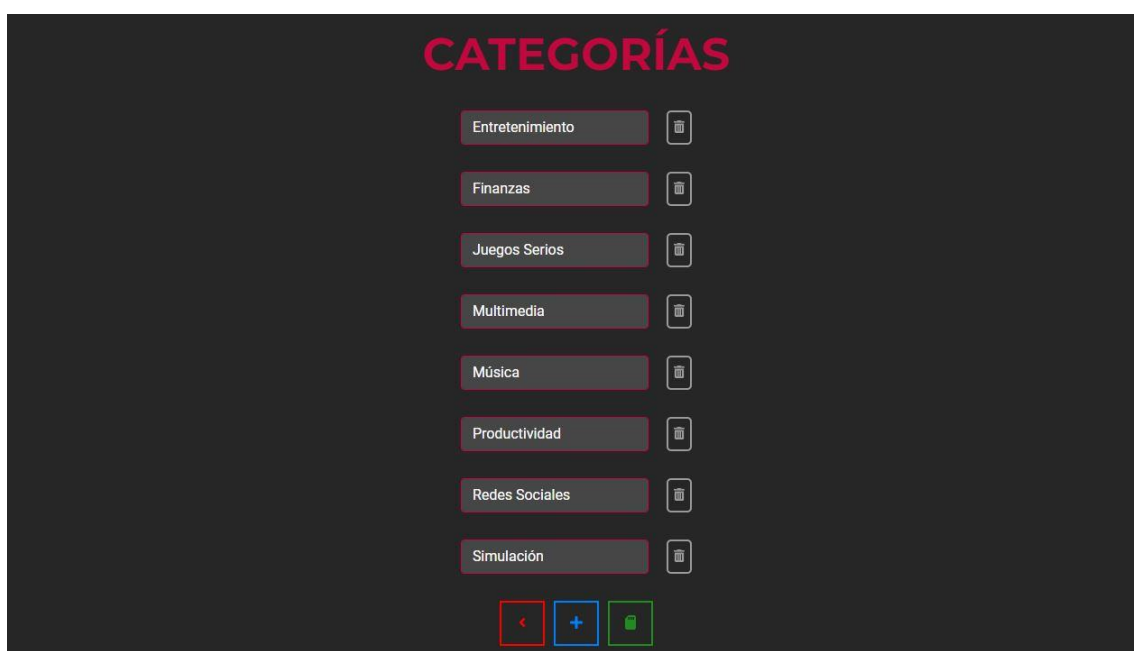


Figura 67: MU - Pantalla de edición de categorías

Al pulsar el botón de categorías el súper administrador puede editar de la misma manera que los grupos disponibles. Estarán disponibles las nuevas categorías a la hora de añadir una nueva aplicación en el sistema.