



Proyecto Final De Grado

SalonBook



Realizado por: José Mateo Romero
Tutor del proyecto: Juan Manuel Hernández Pérez

Índice

1. Definición del proyecto	1
A) Ámbito	1
B) Alcance del proyecto	1
2. Análisis y requisitos del sistema	3
A) Mockup de pantallas	3
B) Modelado de datos	9
3. Diagramas de casos de uso	14
4. Diagramas de clase	17
5. Especificación funcional del sistema	21
A) Manual de usuario	21
B) Manual técnico	45
C) Licencia de uso	52
6. Colorario	54
A) Posibles mejoras	54
B) Viabilidad	55
7. Bibliografía	56
Enlace proyecto Github	56

1. Definición del proyecto

A) Ámbito

Este proyecto tiene como objetivo suplir la necesidad de implementacion de un sistema de gestión de citas para una peluquería situada en Algeciras, esta aplicación mejorará la eficiencia de la peluquería ya que los trabajadores no tendrán que estar atendiendo vía telefónica y con un simple vistazo a su panel de administrador podrán ver cuantas reservas tienen y poder cancelar las reservas, confirmarlas y marcarlas como completadas una vez que el servicio ya ha sido completado, también podrán visualizar los pedidos que han realizado los clientes y de nuevo podrán cancelar el pedido o bien tramitarlo, estas opciones solo las tendrán los administradores de la aplicación. A nivel de consumidor también será retribuido ya que pueden asegurarse de coger la cita en cualquier momento del día con un simple clic y además podrán acceder al contenido extra de la aplicación, como el apartado de compras.

Todo esto ayudará a mejorar la posición de la mencionada peluquería ya que muchos establecimientos de la competencia ya cuentan con este tipo de aplicaciones y han conseguido aumentar la clientela.

B) Alcance del proyecto

El objetivo del proyecto es desarrollar una aplicación para dispositivos móviles, usando las tecnologías más punteras y actuales como lo son actualmente Flutter que utiliza un motor propio de renderización de ventanas a diferencia de Jetpack Compose que usa las nativas de Android, se ha elegido Flutter por el auge de uso en el mercado actual y por la facilidad de adaptación a IOS y por la versatilidad del mismo, para la base de datos se usa Firestore. En el proyecto se ha priorizado la implementación y funcionamiento de:

A) Registro e Inicio de sesión

- Los usuarios tendrán una pantalla donde se les dará la bienvenida a la aplicación y donde podrán iniciar sesión o registrarse en el caso de que sea la primera vez que usa la aplicación. También aparecerá la opción de restablecer la contraseña, con esta opción al usuario le llegará un correo electrónico para establecer una nueva contraseña
- Para los usuarios administradores el sistema gestionará internamente para cuando uno de estos usuarios inicie sesión le aparezca un panel para gestionar las reservas y los pedidos.

B) Calendario

- Los clientes al pulsar reservar una cita se le abrirá un calendario donde podrán seleccionar el día en el que quieran recibir el servicio, una vez seleccionado el día les aparecerán las horas disponibles. Una vez hecha la reserva los propios usuarios podrán cancelarlas sin tener que contactar con la peluquería.
- Los usuarios administradores pueden ver que citas han seleccionado los clientes y podrán confirmarlas o bien cancelarlas

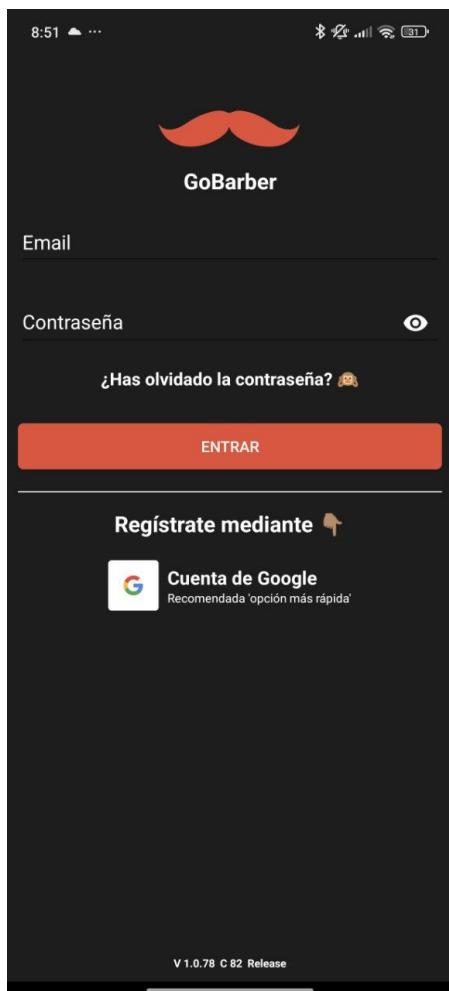
C) Catálogo de Productos

- Se incluirá un apartado informativo donde los clientes podrán visualizar los productos disponibles en la peluquería.
- Cada producto tendrá una breve descripción, una serie de imágenes y su precio.
- La aplicación permitirá la compra de productos, siendo esta sección muy útil para informar a los clientes de los distintos productos y para generar nuevos ingresos

2. Análisis y requisitos del sistema

A) Mockup de pantallas

Como referencia de lo que pretendo que sea mi aplicación he usado la aplicación de gobarber como inspiración para mi proyecto, esta aplicación permite realizar reservas desde el teléfono y tiene más de 100.000 descargas en el playstore.



Cuando entramos en la aplicación lo primero que podemos hacer es registrarnos con nuestros datos



En esta aplicación hay muchos establecimientos registrados, en mi proyecto la app va a ser destinada a una única peluquería, así que centrarme en los datos, pretendo que mi aplicación tenga un apartado donde aparezca la información de la peluquería en cuestión



Se pueden ver todos los servicios que ofrece el negocio, junto al precio y a la duración estimada del servicio.

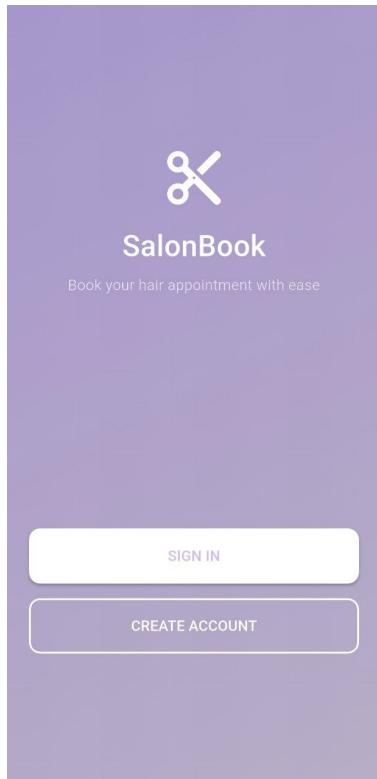
Esto tengo planteado introducirlo si o si en mi proyecto ya que es fundamental para poder solicitar las citas



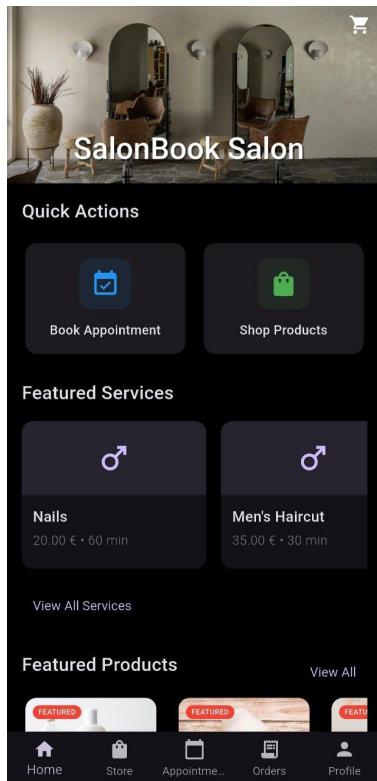
Otra parte en la cual quiero implementar en mi app es la facilidad de hacer una reserva, como se ve en pantalla solo con una vista rápida podemos ver los días que podemos pedir cita.



Otro apartado muy útil es la edición de perfil, ya que con ello podemos editar cualquier dato erróneo



A partir de esta imagen sería mi MockUp, esta primera imagen sería el inicio de sesión

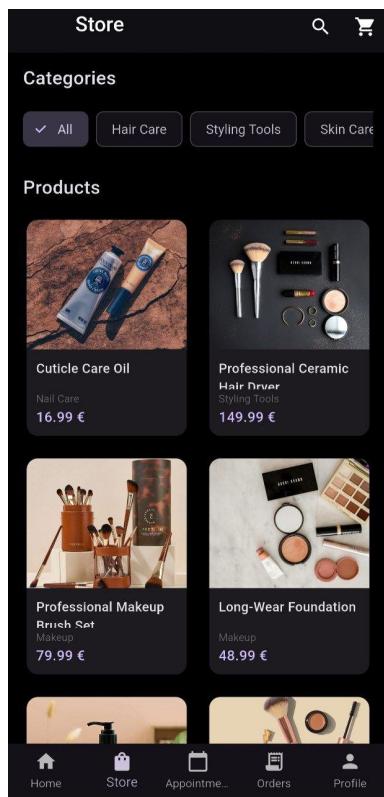


Esta sería la página principal,
En la parte superior pondría una imagen del establecimiento y su nombre,

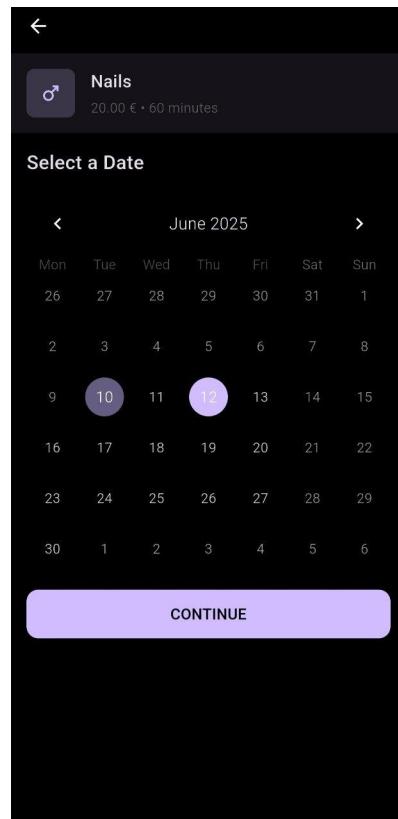
En la parte central accesos directos al apartado de realizar reservas y de compras.

Justo debajo de los accesos directos un cuadrante con información de la peluquería.

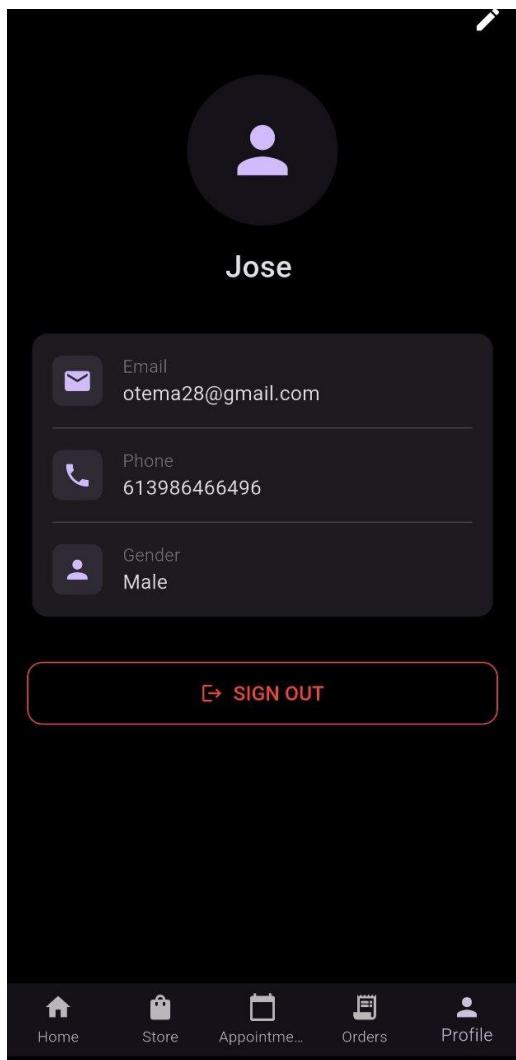
Por último en la parte inferior unos accesos directos de las distintas páginas de la que dispone



En el apartado de la tienda podremos ver distintos productos y dentro de cada producto una descripción y la opción de añadirlo a la cesta para pagar



En el apartado de las reservas aparecería un calendario intercambiable no como el que aparece en la imagen, donde el usuario puede elegir el día y la fecha en la que quiere reservar



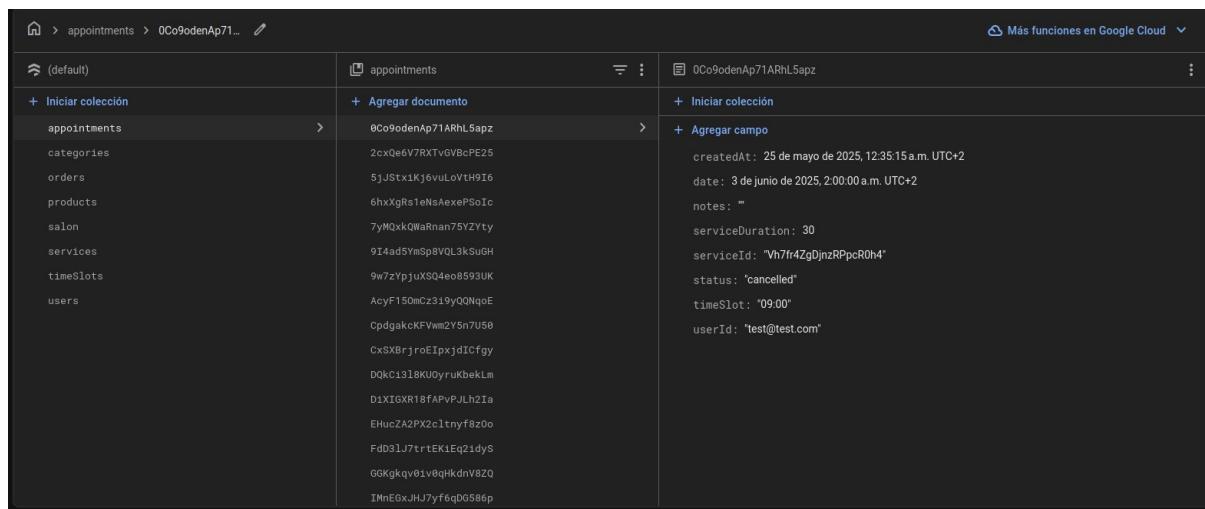
Y este sería el apartado de perfil de usuario, se muestran todos los datos del usuario y les dejaría editarlos si pulsan sobre el ícono del lápiz

B) Modelado de datos

Mi proyecto no tiene una base de datos relacional ya que la única base de datos de la que dispone es firestore la cual es una base de datos no relacional por las siguientes ventajas:

- Mejor rendimiento: las bases de datos no relacionales ofrecen mejor rendimiento a la hora de introducir nuevos datos ya que son más flexibles.
- Flexible en cuanto almacenamiento ya que podemos almacenar información desestructurada, semi estructurada o estructurada ya depende de la necesidad de nuestro caso
- No es necesario configurar un servidor como en el caso de en una base de datos relacional solo con vincularlo estaría realizado el proceso

Por lo tanto la base de datos firestore maneja todo lo que se va a ir explicando a continuación:



The screenshot shows the Google Cloud Firestore interface. The left sidebar lists collections: 'appointments' (selected), 'categories', 'orders', 'products', 'salon', 'services', 'timeSlots', and 'users'. The main area shows a document in the 'appointments' collection with the ID '0Co9odenAp71ARhL5apz'. The document contains the following fields and their values:

campo	valor
createdAt	25 de mayo de 2025, 12:35:15 a.m. UTC+2
date	3 de junio de 2025, 2:00:00 a.m. UTC+2
notes	"
serviceDuration	30
serviceId	"Vh7f4ZgDjnRPpcR0h4"
status	"cancelled"
timeSlot	"09:00"
userId	"test@test.com"

Como podemos ver en la imagen, firestore se encarga de todo el apartado de las reservas, almacenando la información de cada una de las reservas y su estado

The screenshot shows the Google Cloud Firestore interface. On the left, the sidebar lists collections: appointments, categories (selected), orders, products, salon, services, timeSlots, and users. The main area shows the 'categories' collection with a list of documents. One document is selected for detailed view:

```

+ Iniciar colección
+ Agregar documento
3Xk3zHJunEEw9ehqUsld
BmB2EdbQFfUyD4NRgIa
PdQCE8n1wG77aGuRQk1
eXQLd55wfcu6FabPaZz
peAOX8LF0yuvXp0XkCCC

+ Iniciar colección
+ Agregar campo
createdAt: 24 de mayo de 2025, 6:30:59 p.m. UTC+2
description: "Complete nail care collection including polishes and treatments"
imageURL: "https://images.unsplash.com/photo-1604654894610-df63bc536371?w=400"
isActive: true
name: "NailCare"
sortItemsOrder: 3
sortOrder: 5

```

En esta parte se guardan todos los datos sobre las distintas categorías en las que pueden ser ordenadas los productos

The screenshot shows the Google Cloud Firestore interface. On the left, the sidebar lists collections: appointments, categories, orders (selected), products, salon, services, timeSlots, and users. The main area shows the 'orders' collection with a list of documents. One document is selected for detailed view:

```

+ Iniciar colección
+ Agregar documento
0DDRvheM4CQu8ASU2U1D
9RKY8KdhtG86GSPRM76b
AvGEW54Gv1JDCJdaKuFh
B2y5NFFMEijAItvN1J08
DJon18DR4X5lWHWXq181
G9NhwsnP7Vz6EGr21zT
RGUcu6fdzjOrgovJ0H9
VEzIpqU7kRC0DnlRPK5e
abdLsFeAB029Z50gCrj5
c6A5x3H8tih2GrnA9720
n3IBXPtdh2P5bSlCgrA
yGn1zaxOnNJIXzoID4BG

+ Iniciar colección
+ Agregar campo
billigAddress
address: "rrjjndnsn"
city: "ndkkndn"
name: "mandolo@mandolo"
state: "kdnndn"
zip: "ndkdknd"
createdAt: 5 de junio de 2025, 2:08:32 p.m. UTC+2
items
  0
    imageUrl: "https://es.loccitane.com/dw/image/v2/BCDO_PRD/on/demandware.static/-/Sites-occ_master/es_ES/dweft1e8e7b/large/01HN0075K23_lifestyle.png?sw=500&sh=500"
    price: 16.99
    productId: "988Y11FvotJr4IJsGJ0"
    productName: "Cuticle Care Oil"
    ...

```

A continuación se maneja los datos relacionados con los pedidos, dirección, ciudad, nombre, provincia, código postal, el artículo, las unidades...

products > 988Y11FvotJr4IJsGJ0	products >	988Y11FvotJr4IJsGJ0
+ Iniciar colección appointments categories orders products salon services timeSlots users	+ Agregar documento 988Y11FvotJr4IJsGJ0 9fhtRXyGUOiykU5wv5B BHAq3UVeNmQjR5c3XRKf Nj8sx1t9wI713XRQLorM es4hnaGUkH82pMENYH2 gyHqK8n9gEMnhGhMSx6 l2AbQR5Rs1cvEVVAnp0 qdexsSyT6Ttho392UPzV s7WwYfY04qBuXX317Yw v1Oy0ctux6W5xaJWytUG y4hXb2e4NSXMmzkH2bBU	+ Iniciar colección + Agregar campo categoryId: "3Xk3HZJunEEw9ehqUsld" categoryName: "Nail Care" createdAt: 24 de mayo de 2025, 6:31:00 p.m. UTC+2 description: "Nourishing cuticle oil with vitamin E and jojoba oil for healthy, hydrated cuticles." + images 0 "https://es.loccitane.com/dw/image/v2/BCDQ_PRD/on/demandware.static/-Sites-occ_master/-ES/dwef1e8eb7/large/01HN0075K23_lifestyle.png?sw=500&sh=500" isActive: true isFeatured: false name: "Cuticle Care Oil" price: 16.99 + specifications Application: "Brush-on" Key Ingredients: "Vitamin E, Jojoba Oil" Size: "15ml"

En la parte “products” se manejan los datos referentes a productos como el id, el nombre, la categoría, el precio, sus especificaciones...

salon > info	salon >	info
+ Iniciar colección appointments categories orders products salon services timeSlots users	+ Agregar documento info	+ Iniciar colección + Agregar campo address: "Avda. C. Duque de Rivas, s/n, 11207 Algeciras, Cádiz" createdAt: 24 de mayo de 2025, 10:54:07 p.m. UTC+2 description: "Your premier destination for hair styling and beauty treatments." email: "otema28@gmail.com" + images 0 "https://img.peerspace.com/image/upload/f_auto,q_auto,dpr_auto,w_3840/q5nxzhwrf4yaewo5mr" 1 "https://good2b.es/wp-content/uploads/2021/02/25-1024x682.jpg" name: "SalonBook Salon" + openHours + friday close: "18:00" open: "09:00" + monday close: "18:00"

A continuación en el apartado “salon” está toda la información respecto a la peluquería, nombre, teléfono de contacto, dirección...

The screenshot shows the Google Cloud Firestore interface. On the left, the navigation bar includes 'services' and a document ID '1uVkiGo8i5UA79FnkCVf'. The main area shows the 'services' collection with documents for 'appointments', 'categories', 'orders', 'products', 'salon', and 'services'. A specific document under 'services' is selected, showing its details:

- description:** "Professional hair coloring service."
- duration:** 90
- gender:** "All"
- isActive:** false
- name:** "Hair Coloring"
- price:** 75

En el apartado “services” está almacenados todos los servicios que ofrece el establecimiento, con su respectiva información y estado

The screenshot shows the Google Cloud Firestore interface. On the left, the navigation bar includes 'timeSlots' and the date '2025-05-26'. The main area shows the 'timeSlots' collection with documents for each day from May 26 to June 16. A specific slot for May 26 is selected, showing its available time slots:

Available Slot
09:00
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
14:30
15:00
15:30

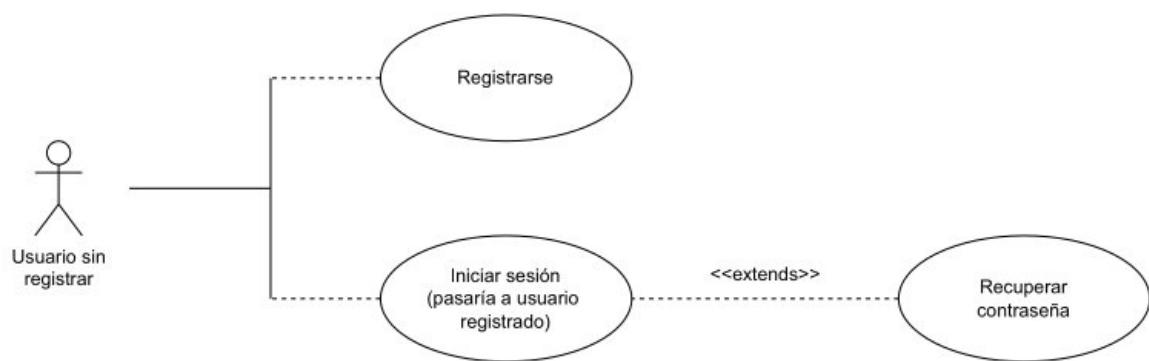
Este apartado que maneja la base de datos es muy importante ya que son las zonas horarias en las que están disponibles las citas, cada documento representa un día en específico, lo que permite una gestión eficiente de horarios. AvailableSlots es un array que contiene strings con los horarios disponibles, estos están comprendidos entre intervalos de 30 minutos, el horario está en formato de 24 horas ya que en formato de 12 puede dar problemas si no se gestiona correctamente.

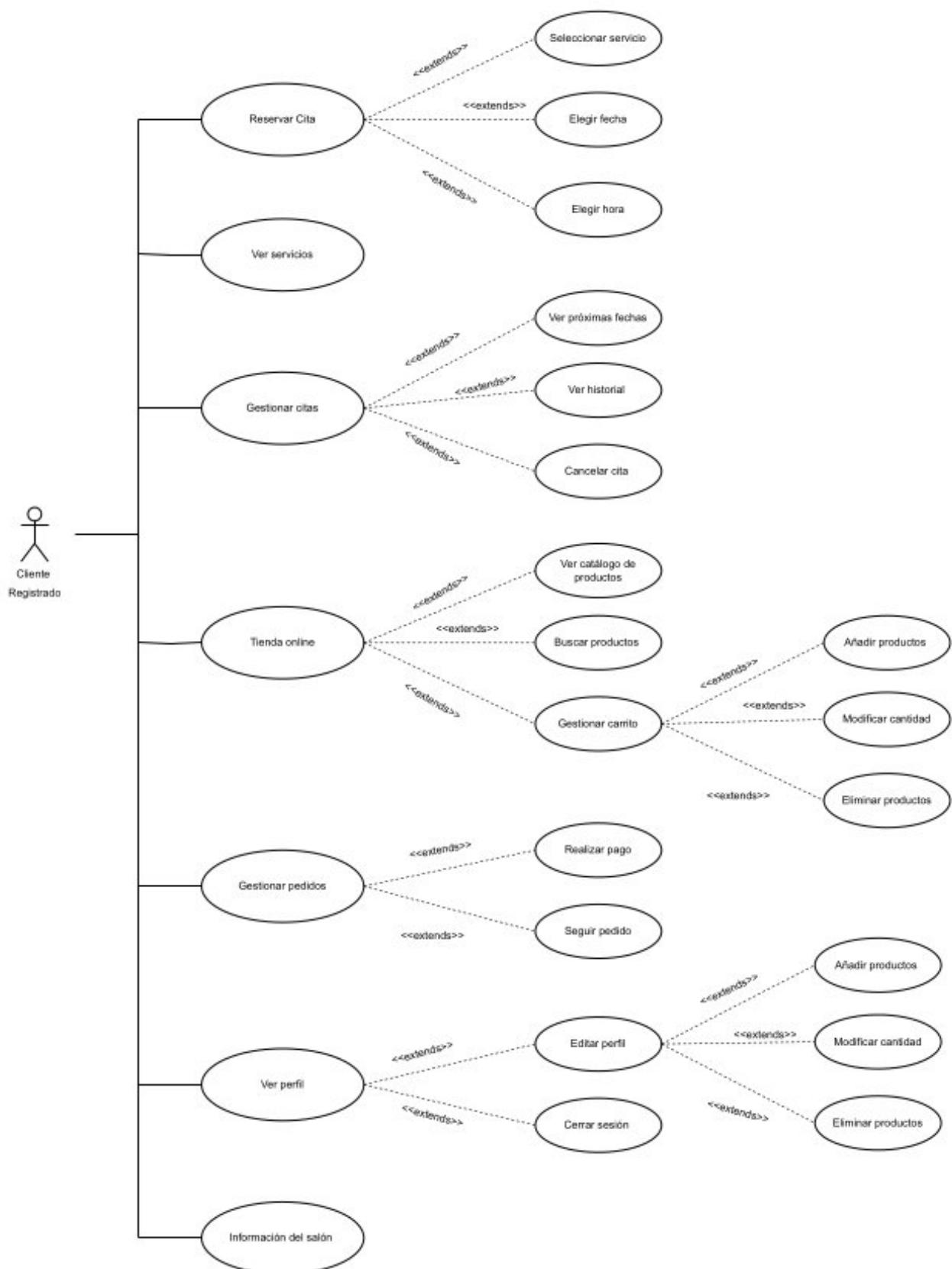
The screenshot shows the Google Cloud Firestore interface. On the left, there's a sidebar with a tree view of collections: (default), appointments, categories, orders, products, salon, services, and timeSlots. Below these is a section for the 'users' collection, which is currently selected, indicated by a blue border. The main content area shows a list of user documents, each represented by a small icon and an email address. One document, 'cuenta1@gmail.com', is expanded, showing its detailed fields: createdAt (20 de mayo de 2025, 11:21:52 p.m. UTC+2), email ('cuenta1@gmail.com'), gender ('Male'), name ('cuenta1'), phone ('655666555'), and role ('client').

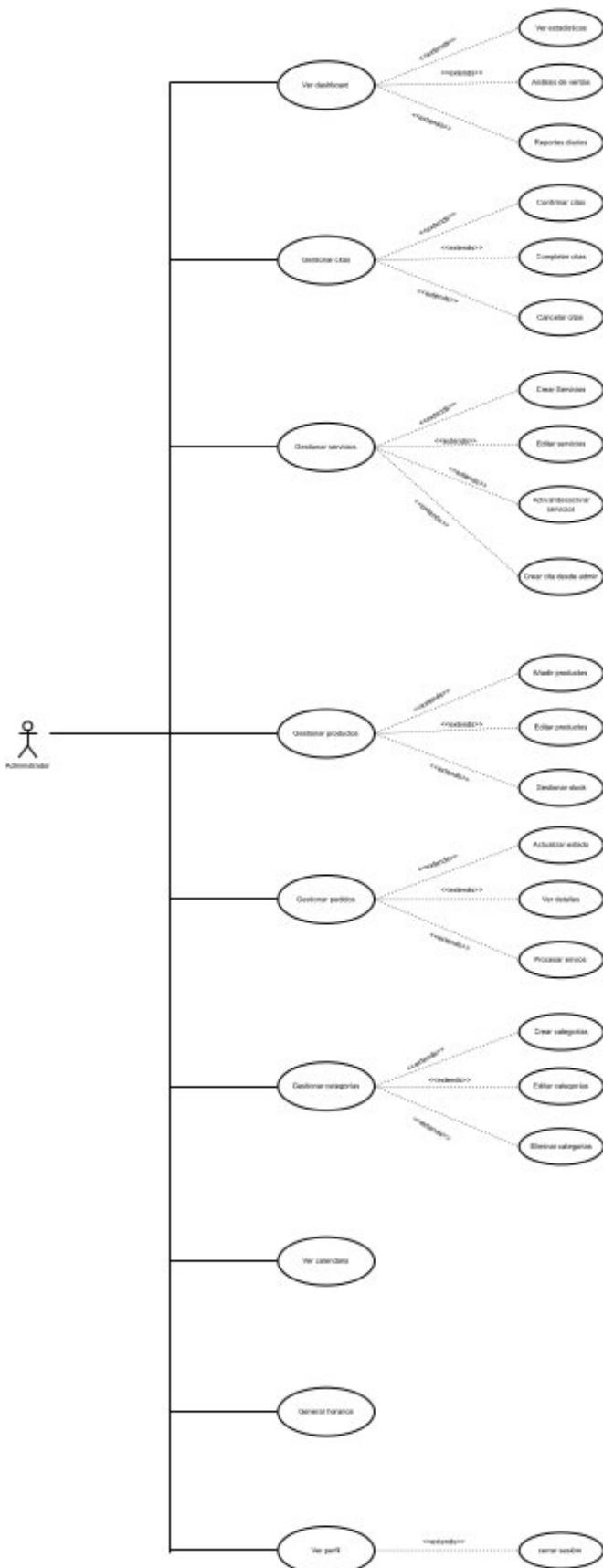
En el apartado de users, se clasifica toda la información respecto a los usuarios, esta parte es muy importante ya que aparte de toda la información de los usuarios aparece el rol, por lo cual hay que tener mucho cuidado con los usuarios que pueden tener acceso a la base de datos ya que si a un usuario se le da acceso de administrador puede acceder a datos confidenciales.

3. Diagramas de casos de uso

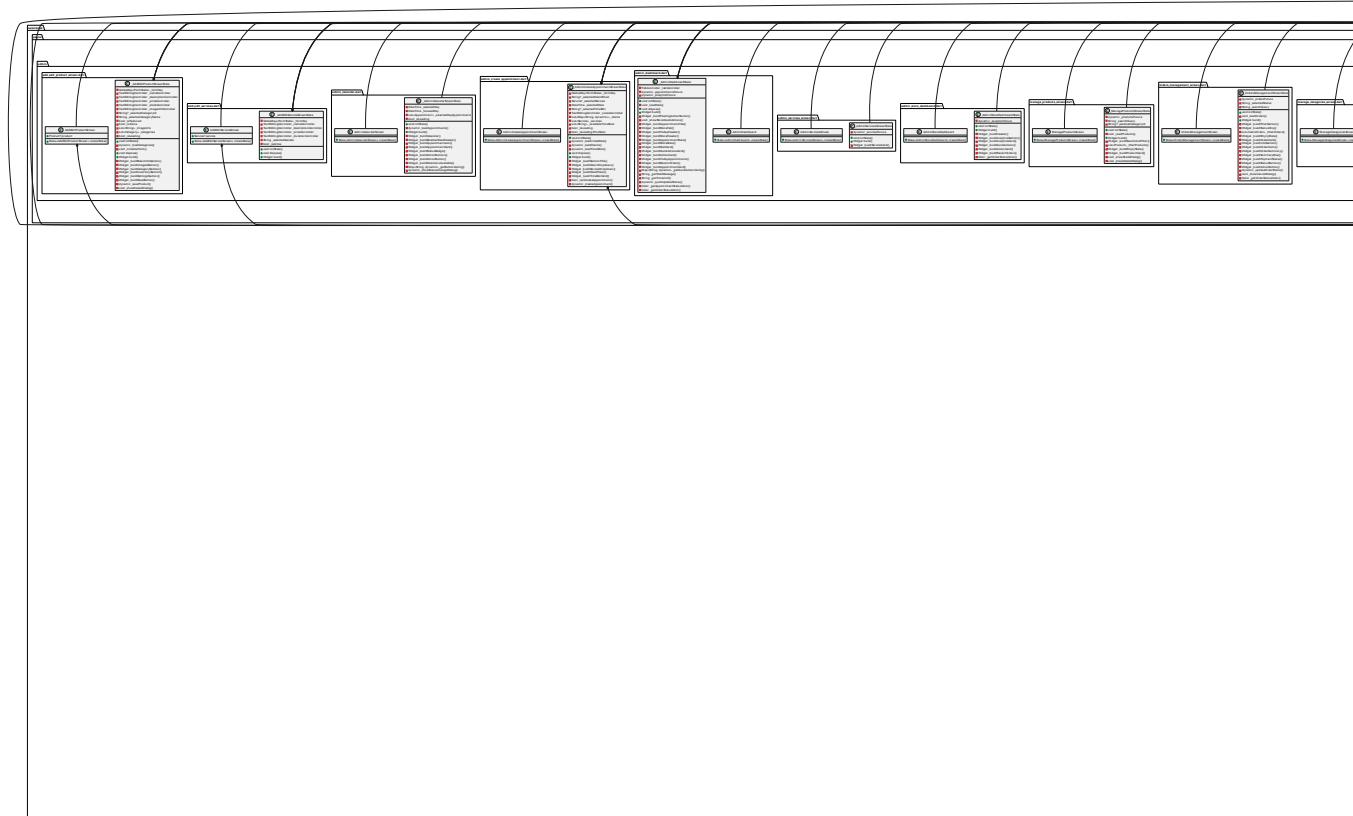
Los diagramas de caso de usos son extremadamente importante ya que nos facilita la identificación y el aprendizaje del proyecto, con ello podemos visualizar las funcionalidades de manera sencilla y rápida

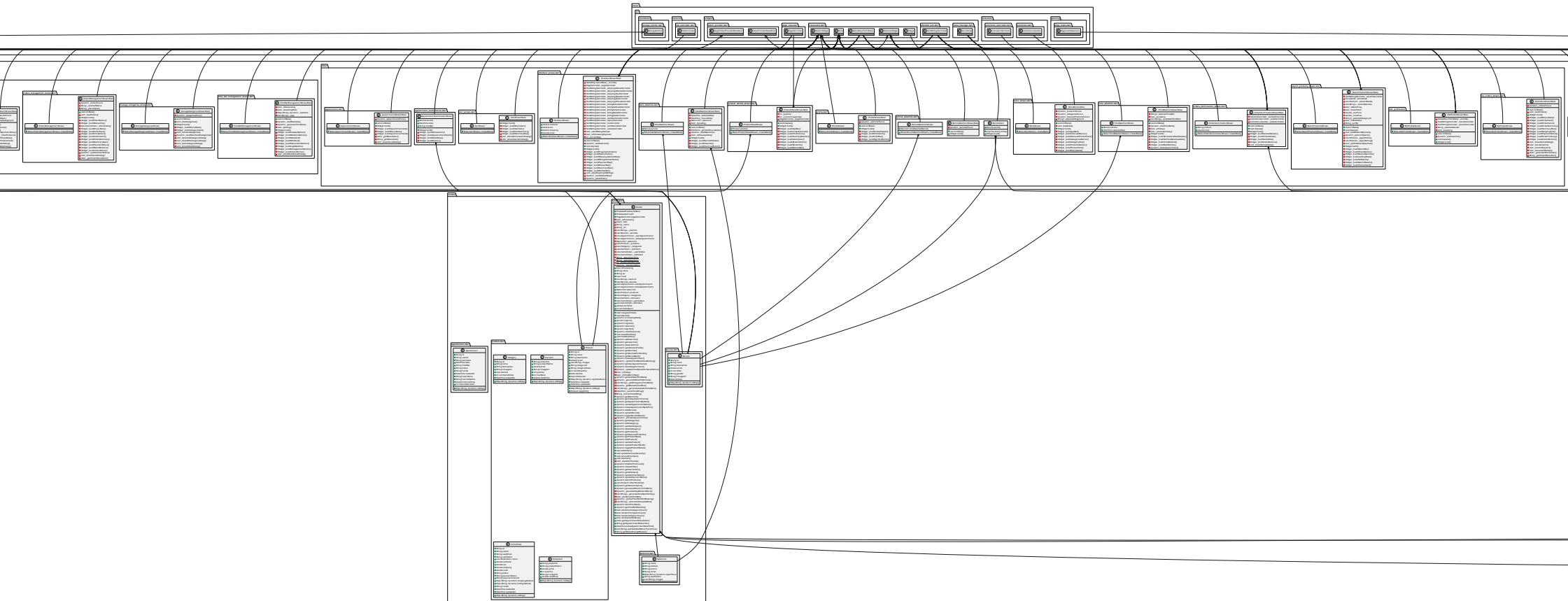


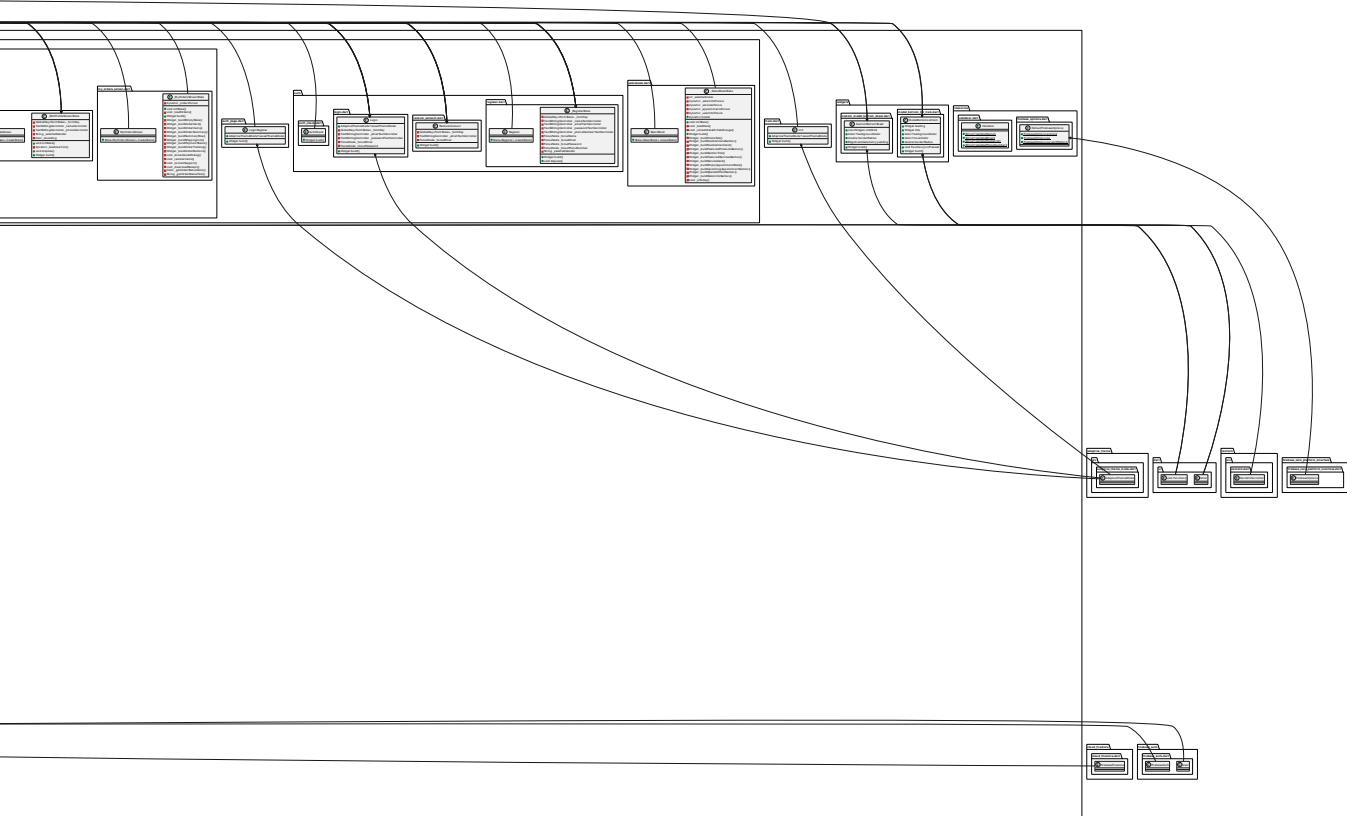




4. Diagramas de clase







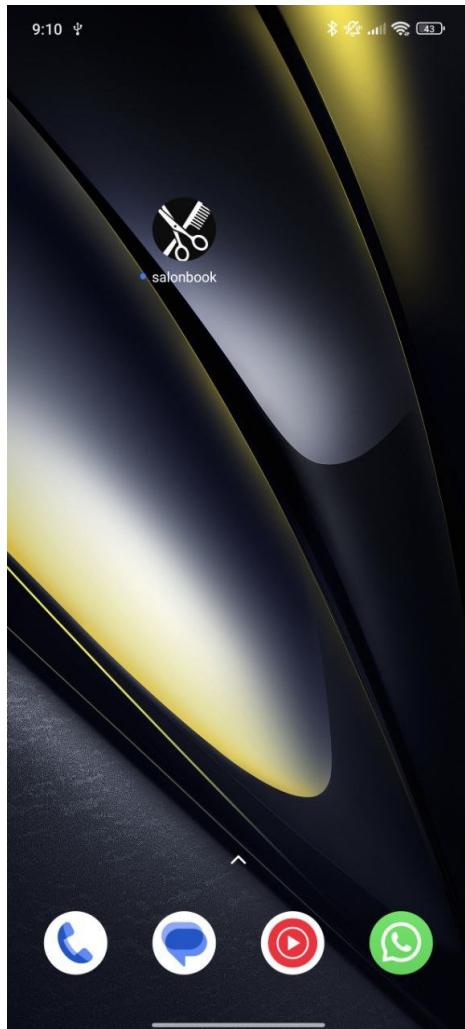
5. Especificación funcional del sistema

A) Manual de usuario

Para la maquetación de las distintas pantallas vamos a dividir en dos partes, vista usuario y vista administrador.

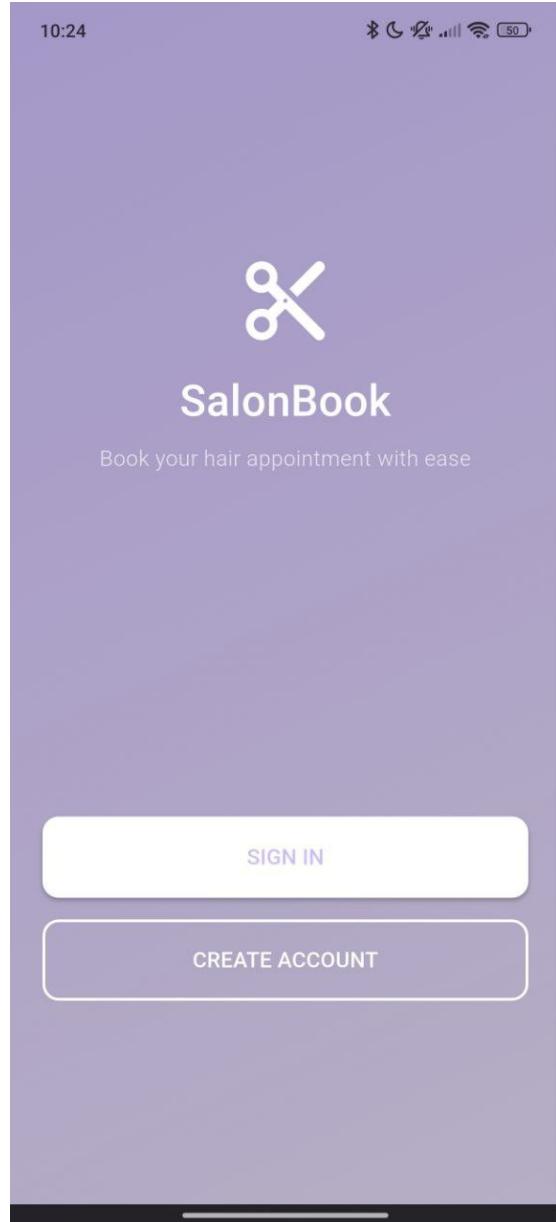
Vista usuario

Esta aplicación llevada al uso real se descargaría e instalaría mediante play store de momento se instala de forma manual con una apk



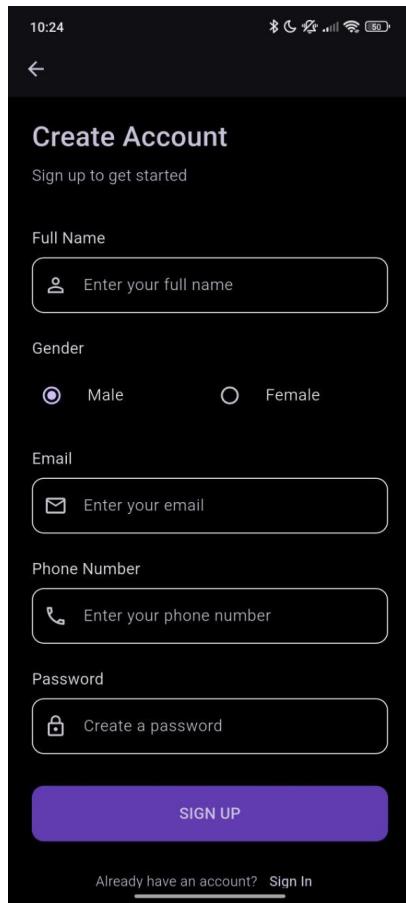
El usuario encontraría este ícono en su pantalla, tocaría sobre el y se le abriría en primer plano la aplicación.

Apertura de la aplicación



Al abrir la aplicación aparecerá la primera pantalla donde tendremos un ícono de la aplicación y un texto que nos da una pequeña idea de cual es la funcionalidad de la aplicación “Book your hair appointment with ease” que sería reservar tu cita de peluquería fácil.

A continuación nos aparecen dos opciones crear una cuenta o bien iniciar sesión en ella.



Al pulsar crear cuenta nos redirige a la siguiente pantalla, aquí debemos de introducir algunos datos cruciales para proceder con la creación de la cuenta como son el nombre, el género, el correo electronico, el número de telefono y la contraseña que debe de tener una longitud minima 8, esta debe estar compuesta por al menos una letra minuscula, una mayúscula, un carácter especial como el @ y números.

Una vez introducido estos datos le damos a Sing Up para finalizar con el registro.

En la parte inferior tenemos un enlace que nos pregunta si tenemos cuenta para llevarnos al apartado de iniciar sesión

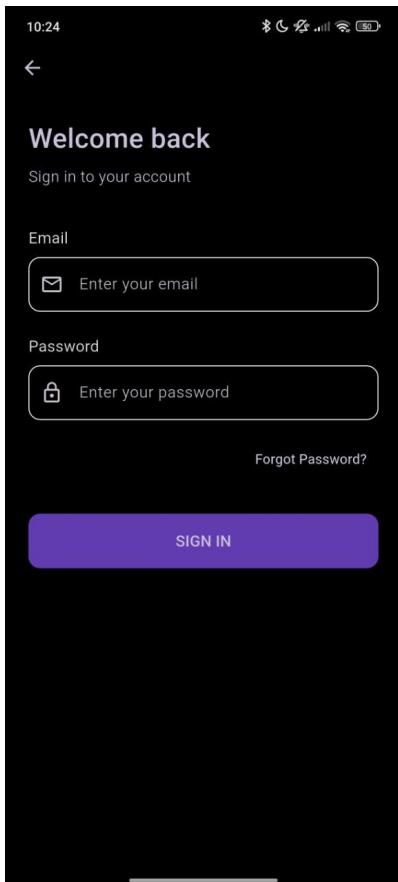
Para crear un usuario administrador el proceso sería el mismo pero una vez finalizada la creación del usuario nos dirigimos a firebase y cambiamos el rol de usuario a administrador

The screenshot shows the Google Cloud Firestore interface. On the left, the sidebar lists collections: (default), appointments, categories, orders, products, salon, services, timeSlots, userinfo, and users. The 'users' collection is expanded, showing documents for 'cuenta1@gmail.com', 'fabiola.cr.2705@gmail.com', 'h@gm.com', 'juan@juan.com' (which is selected and expanded), 'manolo@manolo.com', 'mateo@mateo.com', 'otema28@gmail.com', 'rafamartinez99@gmail.com', 's@s.com', and 'test@test.com'. On the right, the details for the selected document 'juan@juan.com' are displayed:

```

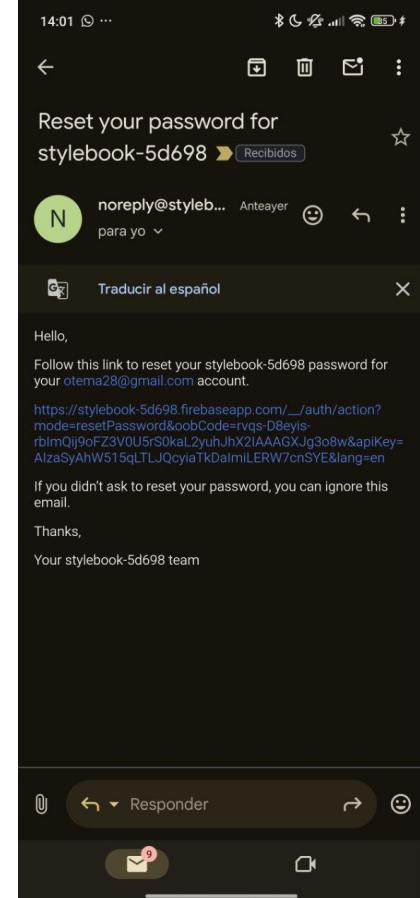
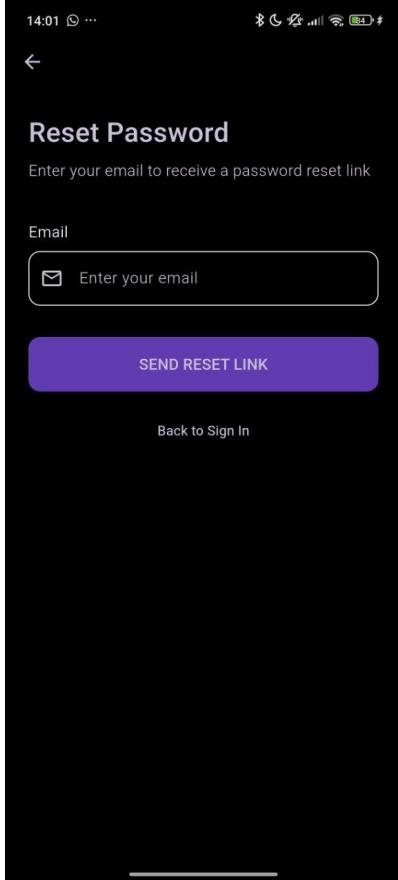
+ Iniciar colección
+ Agregar documento
cuenta1@gmail.com
fabiola.cr.2705@gmail.com
h@gm.com
juan@juan.com >
manolo@manolo.com
mateo@mateo.com
otema28@gmail.com
rafamartinez99@gmail.com
s@s.com
test@test.com

+ Iniciar colección
+ Agregar campo
createdAt: 27 de mayo de 2025, 8:23:14 p.m. UTC+2
email: "juan@juan.com"
gender: "Male"
name: "juan"
phone: "123456789"
role: "admin"
  
```



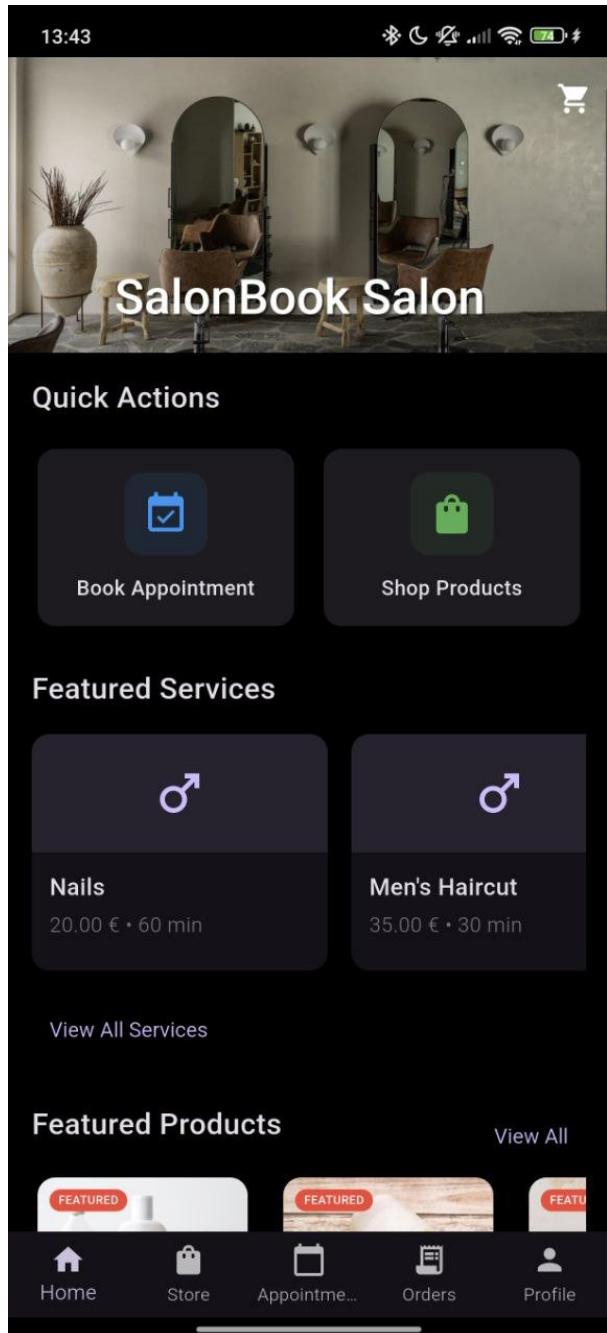
Si en la pantalla principal pulsamos iniciar sesión o bien en la pantalla de creación de cuenta pulsamos sobre el enlace inferior nos dirigirá a esta nueva pantalla. Aquí deberemos de introducir los datos del usuario tanto el correo como la contraseña y clicaremos sobre iniciar sesión para acceder a la pantalla principal.

Si por otro lado el usuario no recuerda cual es su contraseña dispone de un apartado para recibir un correo con un enlace para su debido restablecimiento



Vista principal

La vista aplicación es demasiado grande para solo incluirla en una imagen así que se dividirá en varias



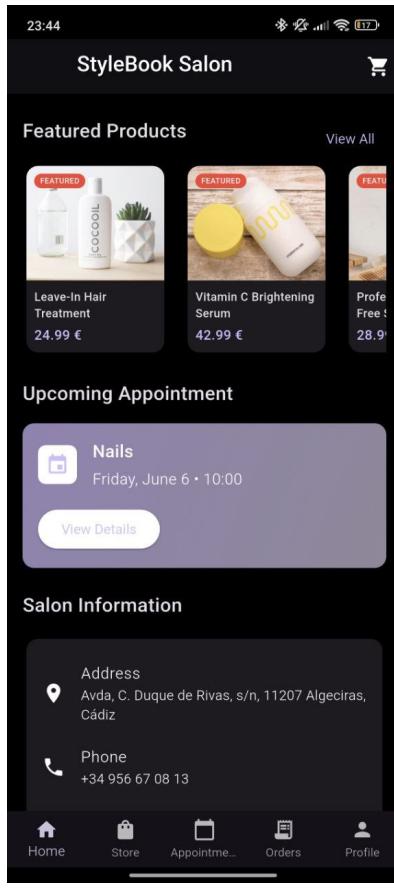
En la página principal encontramos una vista muy cargada de objetos interactivos pero a la vez muy intuitivo.

En primer lugar tenemos una imagen en la parte superior, esta imagen es de una peluquería de sacada de internet pero se podría personalizar en cualquier momento, a continuación en la parte superior derecha vemos un ícono de un carrito si hacemos clic sobre él nos llevará a la tienda.

Dentro de la parte superior también aparece el nombre de la peluquería que en este caso es SalonBook Salon, depende del establecimiento con el que se trabaja se cambiaría el nombre.

Ya en el siguiente apartado “Quick Actions” (acciones rápidas) son atajos para que el usuario tenga a mano el apartado de realizar reservas en la peluquería o compras.

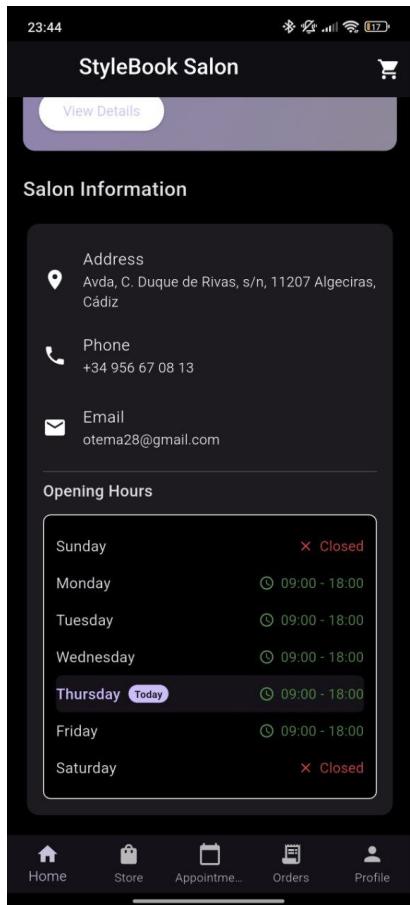
Justo en el siguiente apartado nos aparecen servicios destacados de la peluquería, esto desde la vista admin se puede elegir que servicios se quieren que aparezcan en la página principal,



Siguiendo en la página principal pero desplazada, podemos ver que la imagen de la peluquería ya no está en la parte superior pero sigue quedando el nombre de la peluqueria y en la esquina el icono del carro.

Se puede ver un apartado de Featured Products que serían los productos destacados y vemos una selección similar a la de los servicios, si hacemos clic sobre alguno de los productos nos lleva a su apartado de información y para añadir a la cesta.

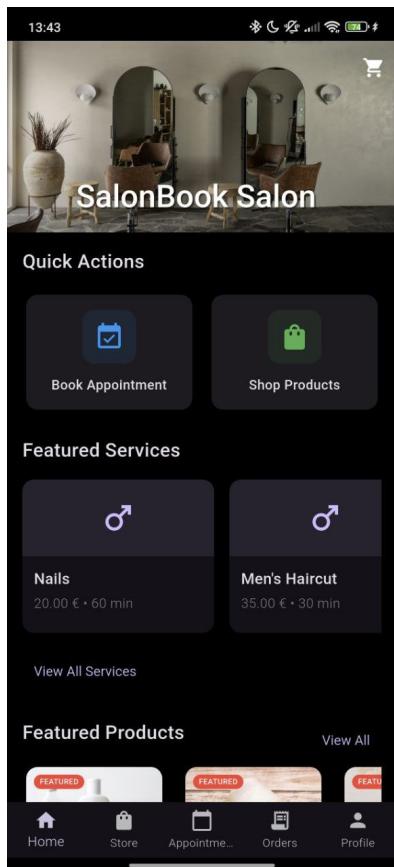
El siguiente apartado le muestra al usuario si tiene una cita concertada, de esta forma es más fácil la visualización y no tiene que irse necesariamente al apartado de citas



Esta sería la parte final de la vista principal. Aquí podemos ver la información de la tienda, ubicación teléfono y número.

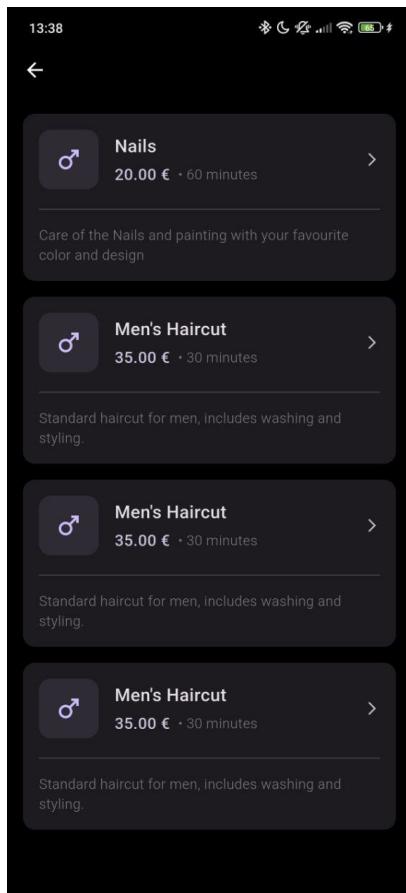
Y por último podemos visualizar el horario de esta peluquería. Además tiene una función que te marca el día en el que estás actualmente

Vista de citas



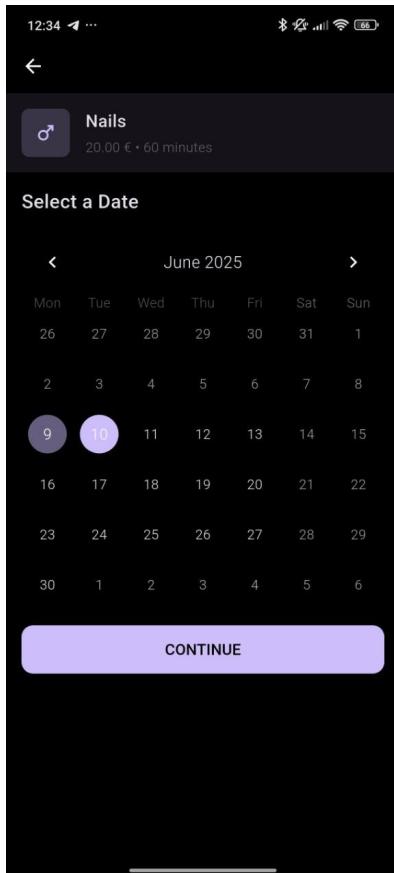
La gestión de citas sería el principal objetivo de la aplicación, todas las demás funciones son extras que le aportan mayor calidad a la aplicación.

Aclarado la prioridad de las citas en el funcionamiento de la aplicación para realizar una debemos de pulsar “Book Appointment”.



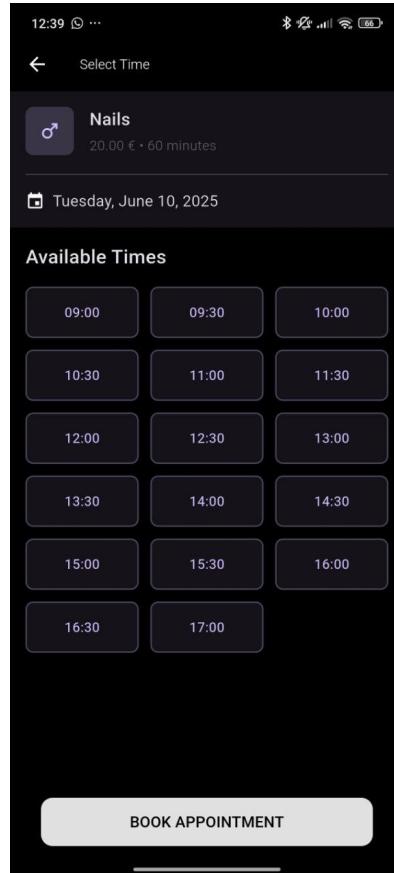
En esta ventana aparecerían todos los servicios disponibles, en cada uno de ellos aparece el nombre, una breve descripción, el precio y el tiempo que se tarda en realizar el servicio.

El usuario seleccionará el servicio que quiere obtener y pasará a la siguiente ventana

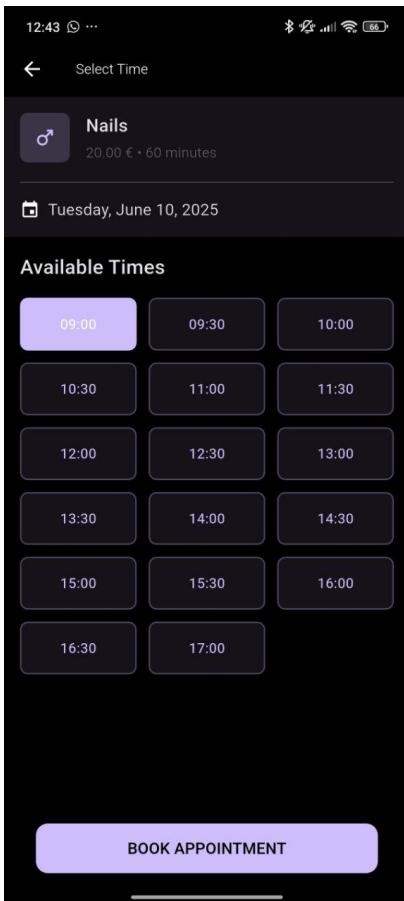


En la siguiente ventana el usuario podra seleccionar el día en el que quiere recibir el servicio el morado mas oscuro es el día en el que estamos y el morado claro es el que el usuario selecciona.

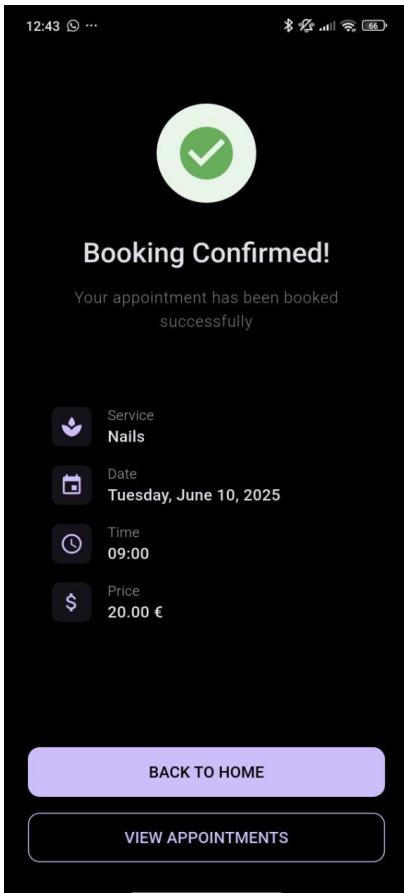
A continuación le damos a continuar para seguir con la reserva



En esta ventana el usuario eligirá entre las horas disponibles la que más le convenga

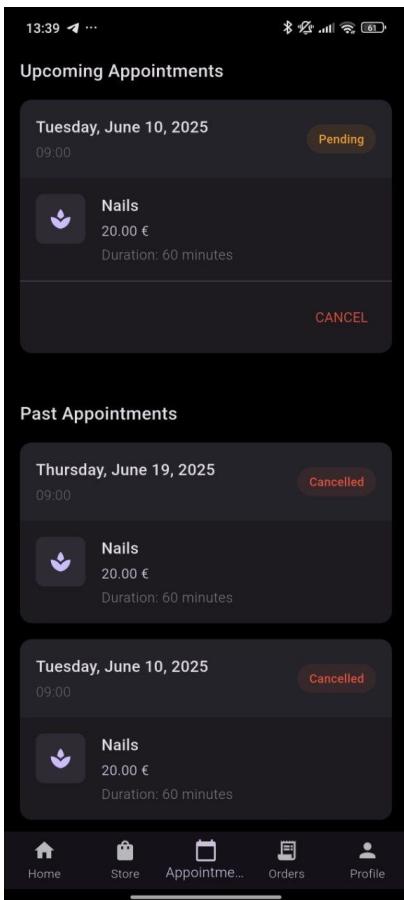


Aquí ya estaría la hora seleccionada y el siguiente paso sería pulsar sobre “Book Appointment” para realizar la reserva

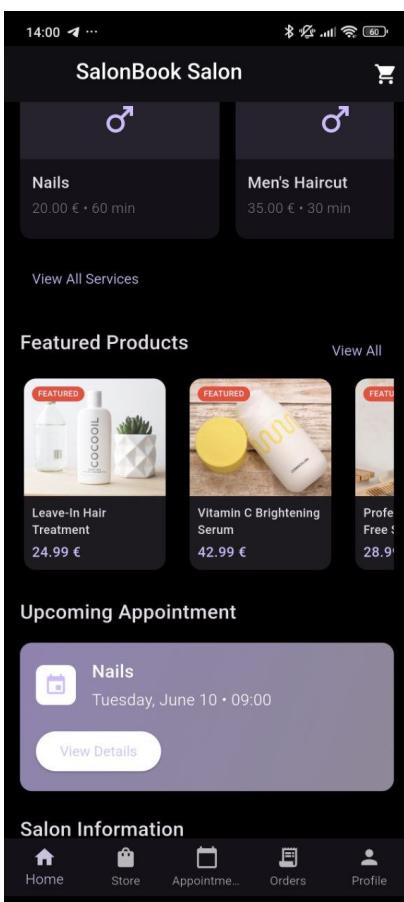


Justo después de hacer clic sobre realizar reserva, aparecerá la confirmación con un breve resumen de la fecha y hora elegida, del servicio y del precio.

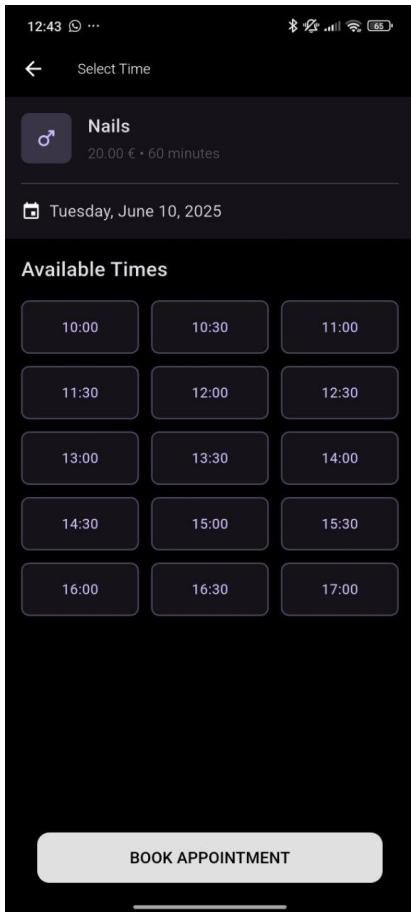
También en la parte inferior nos aparecerán dos botones cuales uno de ellos es para dirigirnos a la página principal de la aplicación y bien la otra es para ir al apartado de las reservas donde nos aparecerá el histórico de solicitudes.



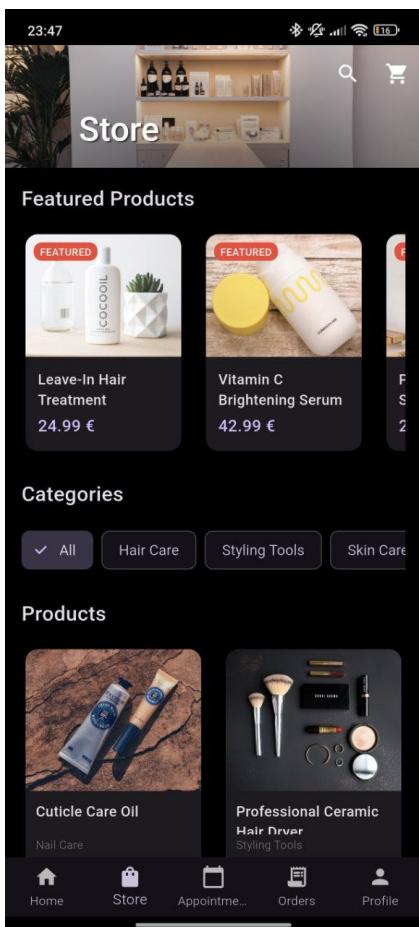
En este apartado podemos ver todas las citas que ha tenido el cliente y el estado de las mismas, el estado puede ser cancelado, ya sea por que el propio cliente lo ha cancelado o por que algun administrador lo ha hecho, también puede ser confirmado, que esto significa que desde la peluquería le han dado luz verde a su cita y esta confirmada o bien el estado de completada que es cuando el cliente ya ha recibido el servicio, para esto el administrador debe marcarlo.



En la vista principal también nos aparecerá en el apartado de próximas citas, la cita que hemos reservado anteriormente, y desaparecerá cuando se cancele o se complete



Si volvemos para hacer una reserva podemos ver que han desaparecido dos tramos horarios que son los de 9 y 9:30 ya que el servicio elegido dura 1 hora y la siguiente hora disponible serían las 10.

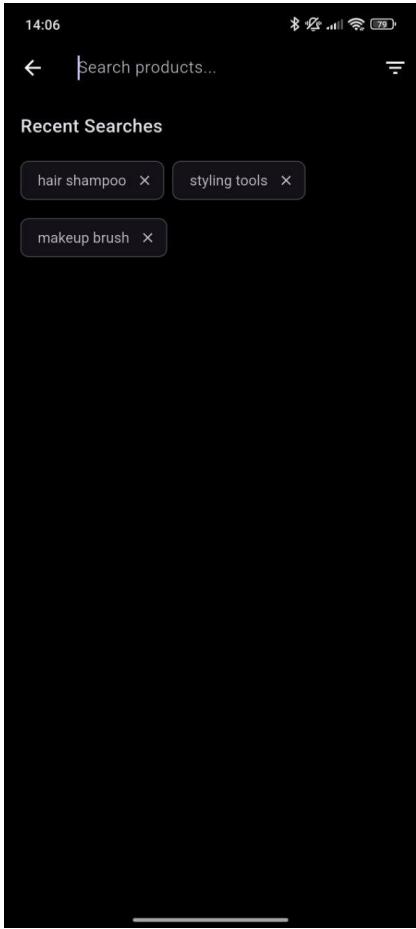


Vista tienda

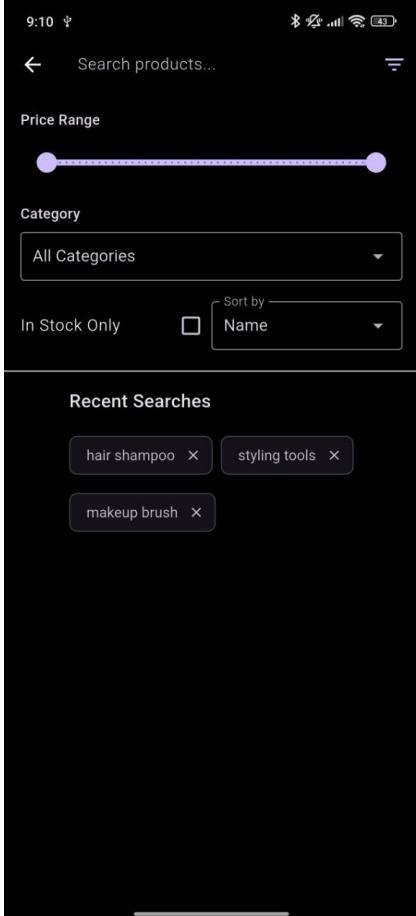
Esta sería la vista de la store, en la parte superior tenemos una imagen de una estantería de productos que se puede modificar en cualquier momento. En la parte superior derecha tenemos un ícono de búsqueda para poder encontrar el producto que deseemos más facilmente. Junto al ícono de búsqueda nos aparece el carrito de la cesta ahí podremos tramitar el pedido.

Como en la pantalla principal en la store nos encontramos los productos destacados para potenciar sus ventas

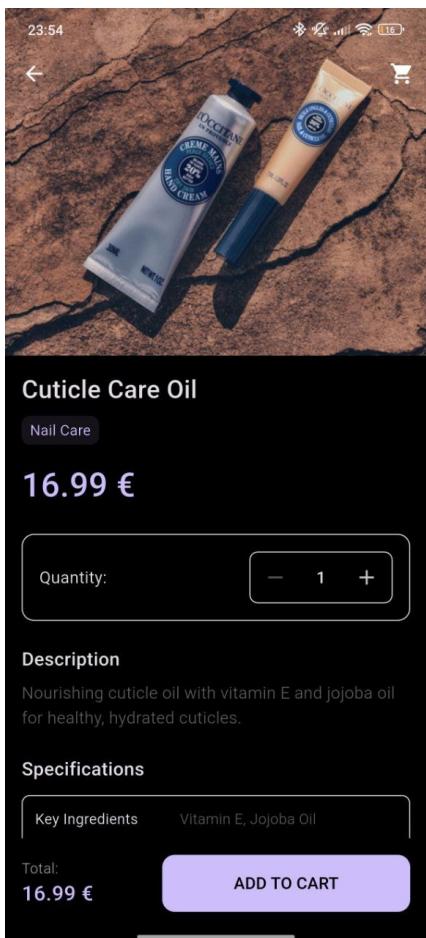
Justo debajo de los productos destacados ya podemos encontrar todos los artículos que ofrece la peluquería con la opción de filtrarlos por clases



Aquí podemos buscar el artículo en el buscador y si pulsamos en las tres barras superiores se nos abrirá un filtrado avanzado



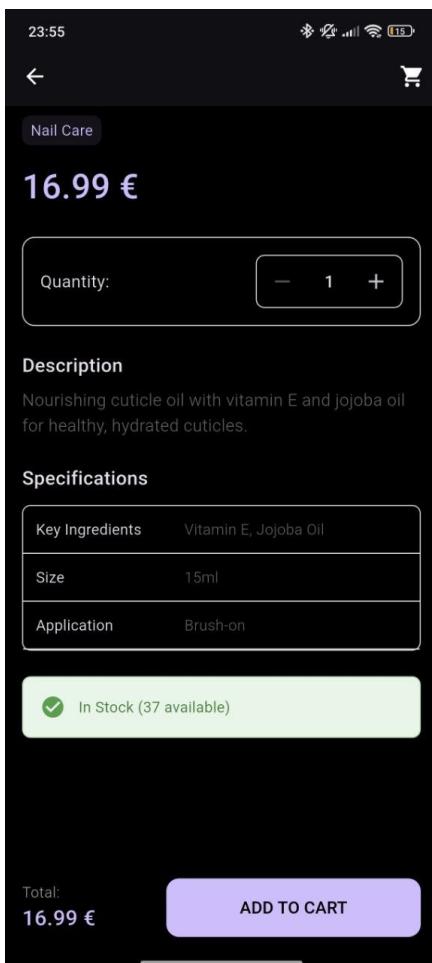
En el filtrado avanzado, como podemos ver nos deja filtrar por precios, por categoría, si hay stock, si queremos filtrar por nombre o por precio.



Cuando accedemos al producto que queremos ya sea desde la página store, desde el apartado de productos destacados o bien desde el buscador nos aparecerá esta ventana,

Aquí podremos ver una imagen del producto con su respectivo nombre, las unidades que deseamos de comprar y una breve descripción.

En la siguiente imagen será una continuación a esta pero más desplazada



Continuando con la vista de un producto podemos ver las especificaciones, tamaño del articulo como aplicarlo y los ingredientes que tiene.

Además que aparece el número de stock disponible, este stock decremente automaticamente cuando un usuario realiza un pedido.

En la parte inferior tenemos el total que se calcula con las unidades que tengamos seleccionadas y el botón de añadir a la cesta

Cuando pulsamos añadir a la cesta aparece en el ícono del carrito un número que concuerda con el número de productos que hemos añadido.

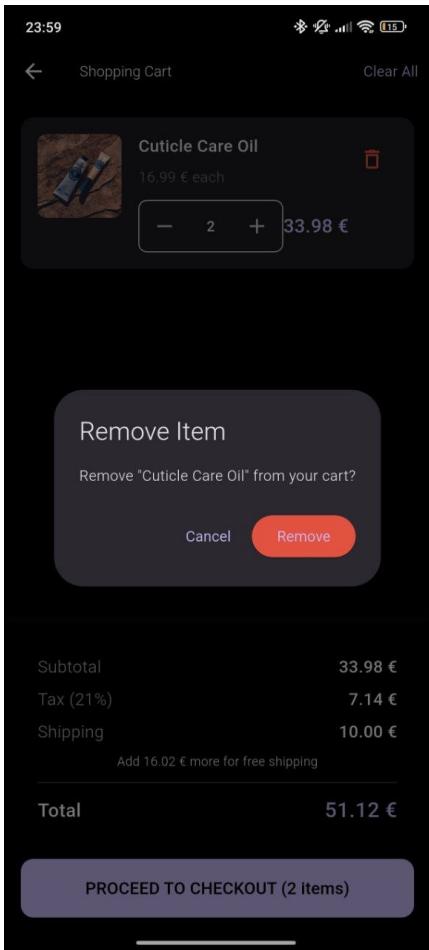
El siguiente paso es pulsar el ícono para poder continuar con el proceso de compra

Subtotal	33.98 €
Tax (21%)	7.14 €
Shipping	10.00 €
Add 16.02 € more for free shipping	
Total	51.12 €

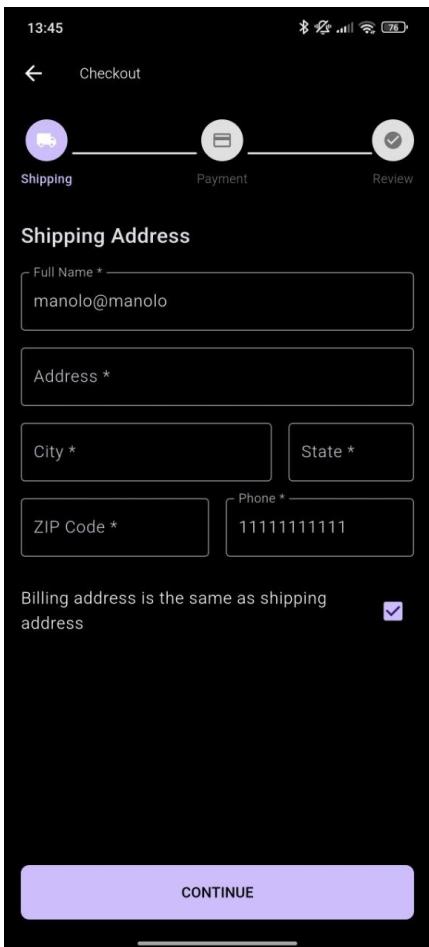
Esta sería la vista del carrito, de nuevo podemos modificar la cantidad y eliminar el artículo.

En la parte inferior podemos ver el desglose del pedido un 21% de IVA y 10€ de envío

SI pulsamos sobre continuar nos llevará a la siguiente vista para continuar con la compra

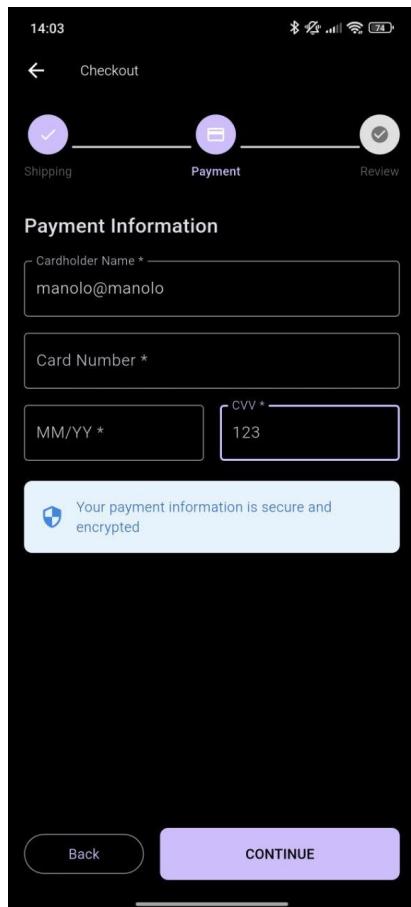


Este sería el cuadrante de confirmación para eliminar el artículo



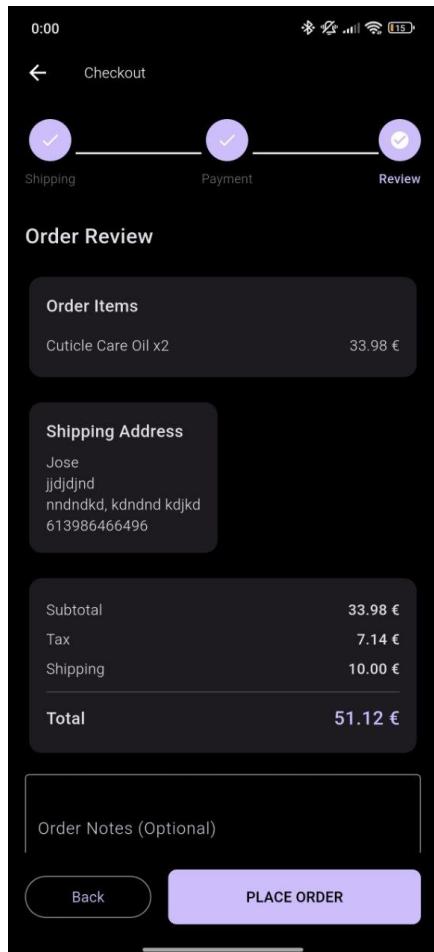
Continuando con el proceso de compra de productos, la siguiente vista, será un formulario para llenar con los datos de dirección del cliente para recibir su pedido y para facturarlo.

Una vez más una vez llenado los datos pulsamos sobre continuar para el siguiente paso



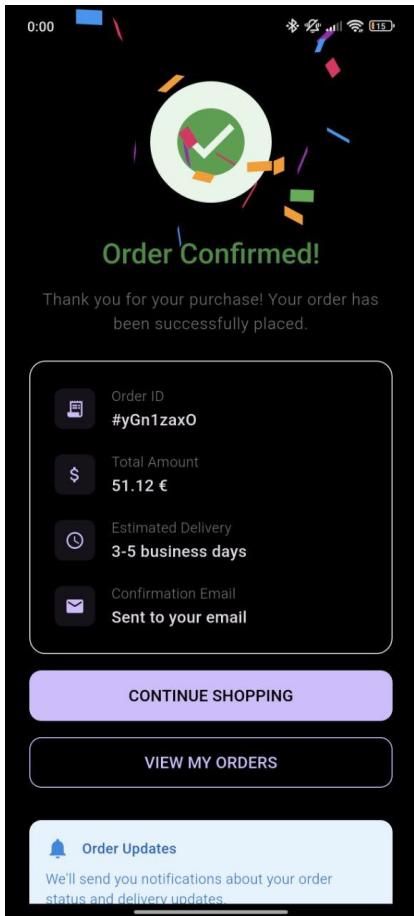
Continuando con el proceso de compra esta tal vez es la parte más polemica de toda la aplicación ya que simplemente es una pasarela a la siguiente vista. Con ello quiero decir a que no se va a realizar ningún cargo a la tarjeta ya que no está indexada a nada. Si la aplicación acaba implantándose cambiaría esta vista por la pasarela de pago de PayPal que es gratuita o bien algún plugin que son de pago.

Bueno siguiendo con el proceso deberíamos de introducir los datos y pulsar continuar



En esta vista el cliente tendría un resumen de su pedido y la opción de dejar una nota para que la tenga en cuenta a la hora de tramitar el pedido el establecimiento. Esta pantalla es muy útil debido a que el cliente puede confirmar que todo está bien.

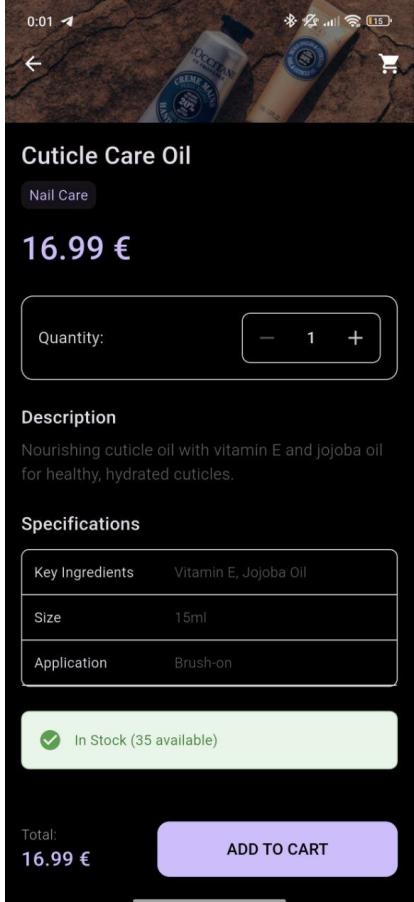
Una vez comprobado todo el cliente debería de pulsar en place order para efectuar la compra.



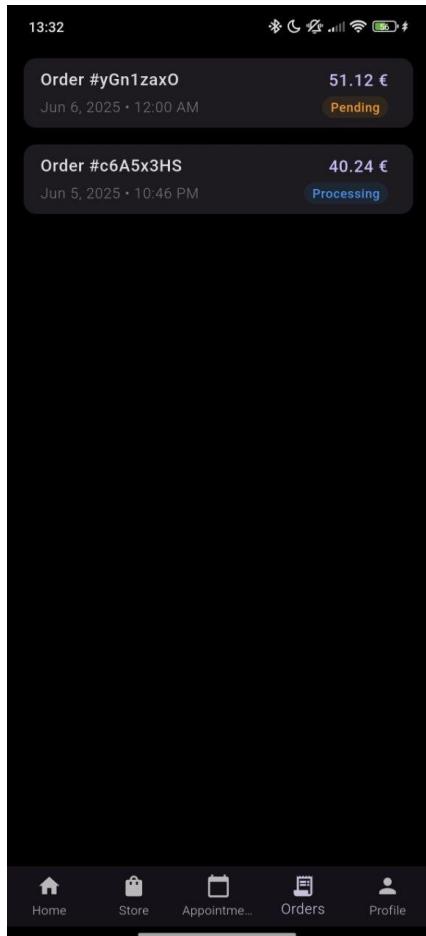
Esta pantalla es la confirmación de que la compra se ha realizado correctamente y aparece con una animación, le da la información del id que se ha generado para su pedido y una estimación de entrega.

Además al usuario le llega una confirmación del pedido a su correo electrónico

En la parte inferior hay dos botones uno que llevaría de nuevo a la store “continue shopping” o bien el otro que le lleva a la ventana donde aparecen todos los pedidos “view my orders”

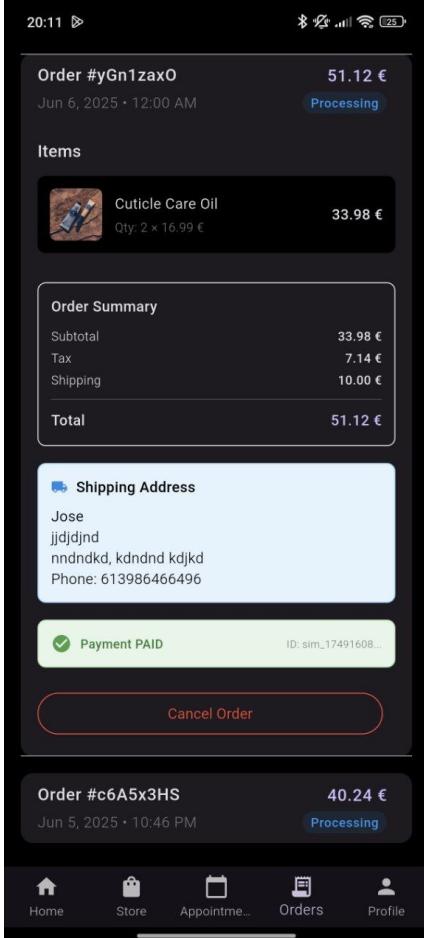


Si volvemos al artículo podemos ver que el stock ha bajado en dos unidades



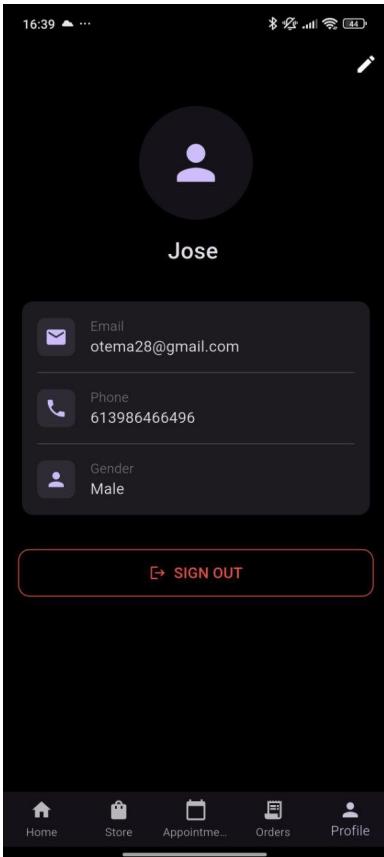
En esta ventana podemos ver que en el apartado de pedidos vemos un historico de los pedidos que hemos realizado

Además podemos ver dos de los tres posibles estados de un pedido, pendiente de confirmación por la tienda y procesando que sería que la peluquería esta preparando el pedido



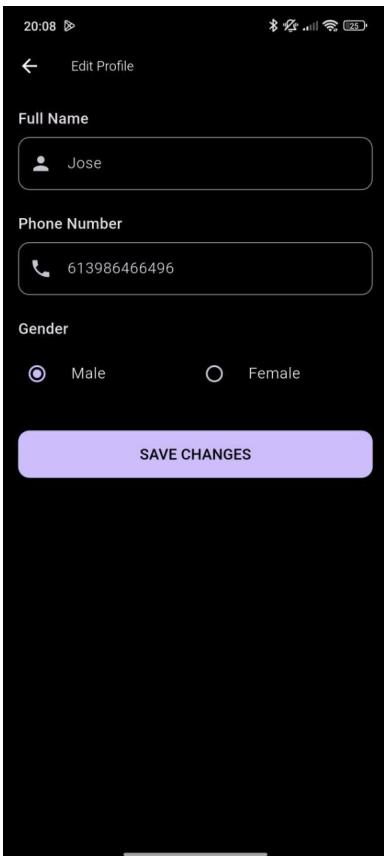
Si pulsamos sobre el pedido en la vista de orders se nos desplegará esta venta que nos informa del desglose del pedido, la dirección del pedido, si se ha pagado correctamente y en la parte inferior un botón de cancelar pedido

Vista perfil



En esta vista aparecen los datos del usuario, su nombre, su correo electrónico, el número de teléfono y su género.

Aquí podemos o bien cerrar sesión pulsando "Sign Out" o bien editar esta información pulsando en el ícono del lápiz en la parte superior

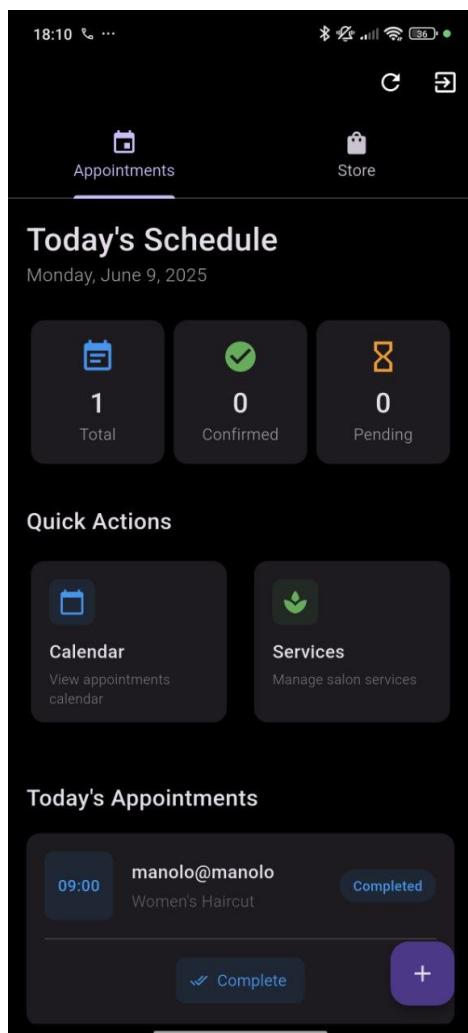


Desde la página de edición de perfil podemos cambiar el nombre, el número de teléfono y el género.

Pulsando el botón efectuaremos los cambios que hayamos realizados

Vista Administrador

La principal función de la vista administrador es poder administrar correctamente todo lo relacionado con la peluquería por lo cual el administrador será un trabajador de la propia peluquería y tendrá la opción de poder añadir las citas que no han sido concertadas para llevar un historial de todos los servicios que han servido y así gestionar la facturación eficientemente. En todas las vistas de administrador tiene un ícono para refrescar la página y así poder ver las nuevas reservas o pedidos.



Esta sería la vista principal del administrador, se divide en dos secciones la que se muestra en la imagen que sería la de las citas y la de la tienda que se selecciona en la parte superior.

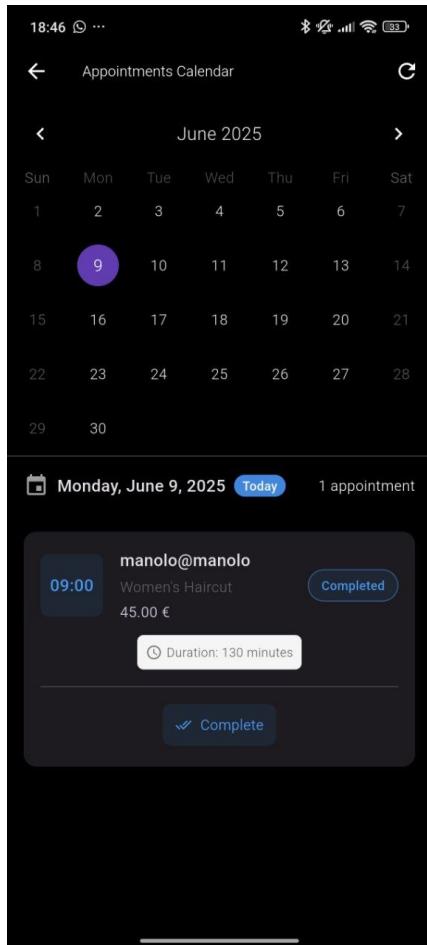
Centrándonos en la vista de las citas, podemos ver tres accesos directos que muestran de manera sencilla y útil ya que hace un pequeño resumen del total de reservas que hay en el día de hoy, las reservas que ya están confirmados y las que están pendientes de confirmación.

En el ícono de calendario nos aparecerá el propio calendario y según el día que seleccionemos nos aparecerán las reservas

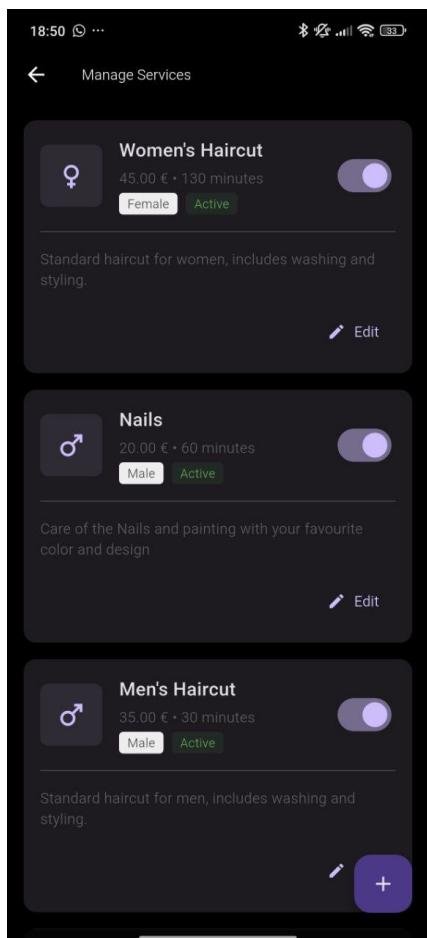
En el ícono de servicios podremos añadir o quitar servicios y demás

Y luego en la parte inferior tenemos todas las citas del día

El ícono de + es para añadir citas manualmente, estas citas pueden ser las que han sido seleccionadas por teléfono o presencial, ayuda a que haya una buena organización



Así se vería una cita que ya está completa



Si pulsamos en el ícono de servicios accederemos a la vista de gestión de servicios donde pulsando el botón podremos activar o desactivar algún servicio rápidamente, además si pulsamos en editar podemos editar el servicio.

Si pulsamos sobre el ícono + podremos añadir nuevos servicios

18:50 ⓘ ...

Edit Service

Service Name

Women's Haircut

Description

Standard haircut for women, includes washing and styling.

Price (€) Duration (minutes)

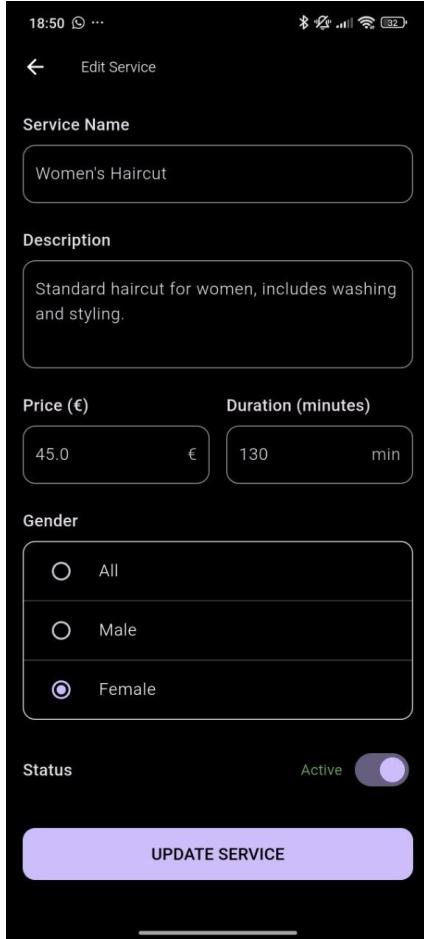
45.0 € 130 min

Gender

All
 Male
 Female

Status Active

UPDATE SERVICE



En la vista de edición podremos editar absolutamente todo lo que aparece en pantalla, el nombre del servicio, descripción, precio, la duración, en que perfiles debe aparecer si solo masculino, solo femenino o ambos, también el estado que puede ser activo o desactivo y por último en la parte inferior encontramos un botón para poder actualizar todos los servicios

19:09 ⓘ ...

Add Service

Service Name

Enter service name

Description

Enter service description

Price (€) Duration (minutes)

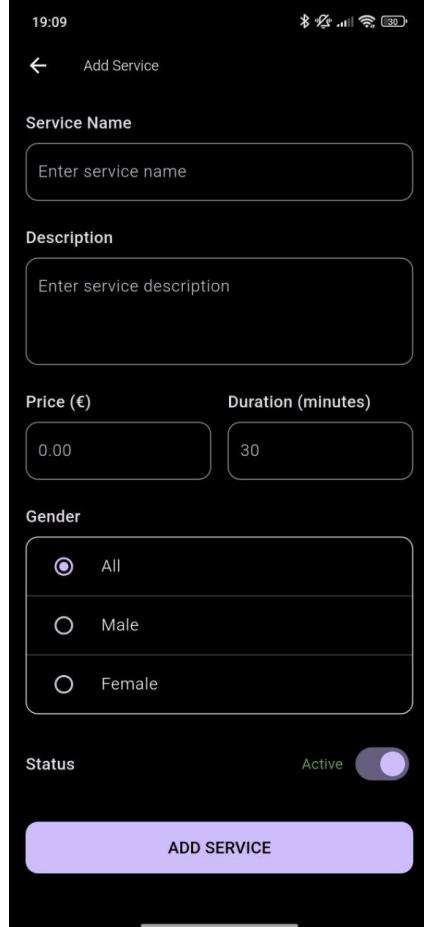
0.00 30

Gender

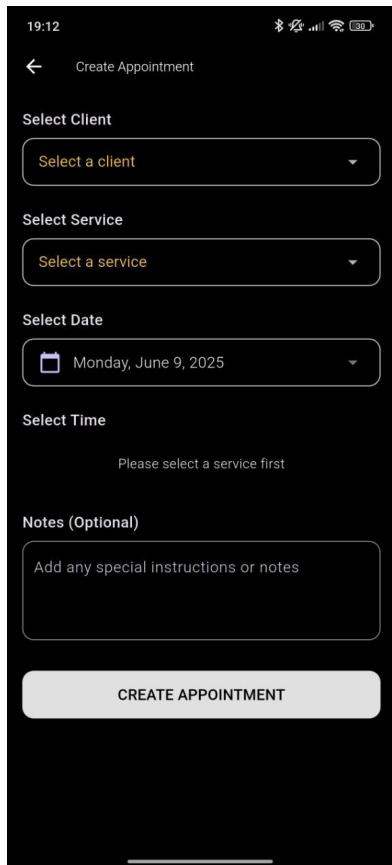
All
 Male
 Female

Status Active

ADD SERVICE

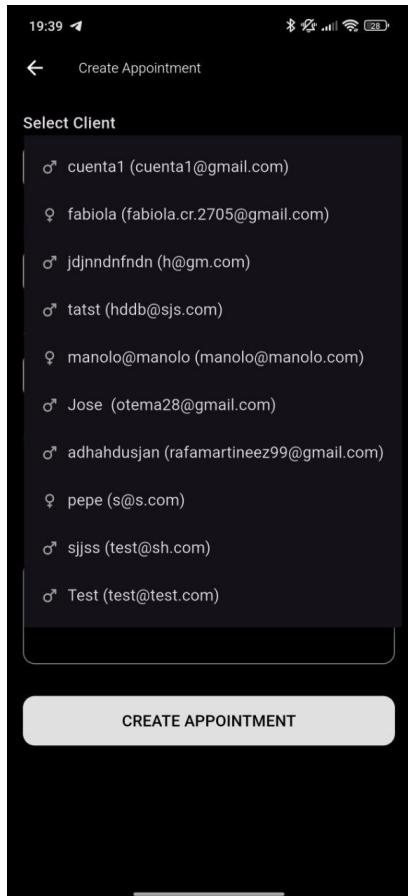


Y esta sería la vista para crear un nuevo servicio, parecida a la de actualizar pero con todos los campos en blanco y con el botón de añadir servicio

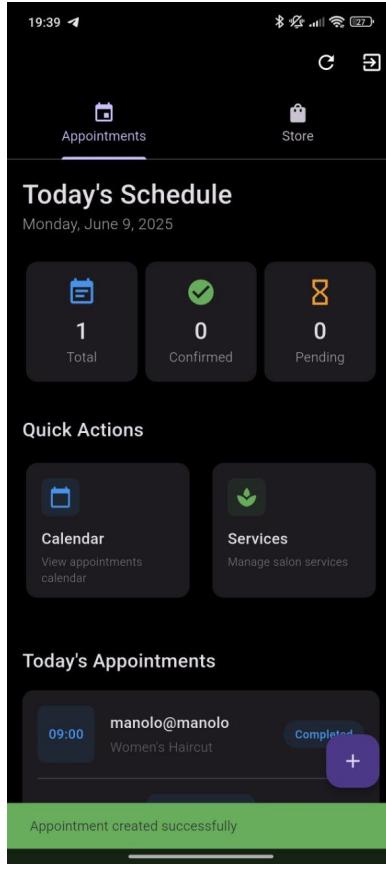


Volviendo a la vista principal de de reserva de citas, si pulsamos en el icono + se abrirá esta nueva ventana donde podremos introducir reservas manualmente.

Este apartado es muy útil ya que si se introducen las reservas realizadas vía telefónica o bien las presenciales, con ello llevaremos un mejor control de las horas disponibles y tambien pueden usar los datos para una posterior facturación.

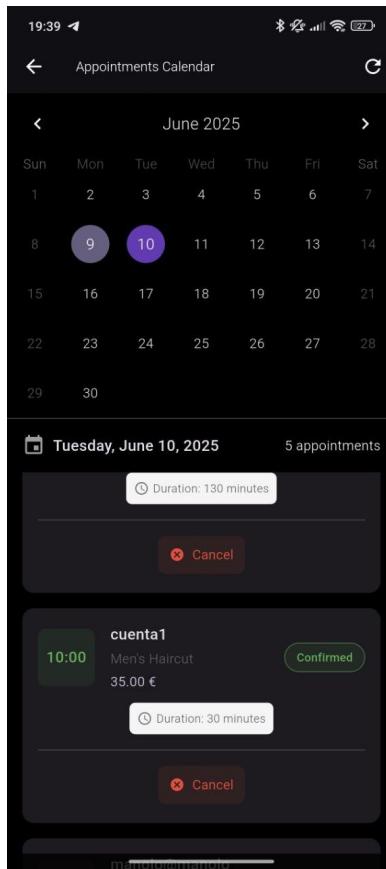


Para dar de alta una cita introducida manualmente, debemos de elegir una cuenta, si en un caso hipotético de que un cliente acuda presencialmente a la peluquería y no tenga un perfil creado puede usar una cuenta que tenga la peluquería para hacer pruebas como la cuenta1



Una vez introducido todos los datos, pulsamos en el botón inferior para crear la cita.

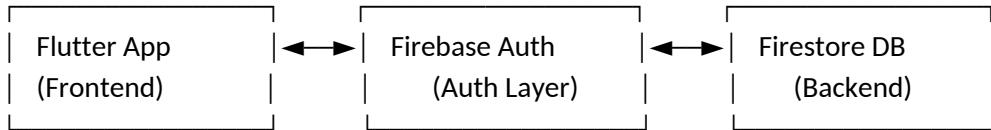
Una vez hecho nos aparecerá un mensaje en el inferior de la pantalla que nos indicará cita reserva correctamente



En el calendario podemos ver que la reserva se ha registrado correctamente

B) Manual técnico

Arquitectura del sistema:



Patrón: MVVM con Provider para gestión de estado

Backend: Serverless con Firebase

Base de Datos: NoSQL (Firestore)

Tecnologías Usadas:

Frontend

- **Flutter**: 3.32.x - Framework multiplataforma
- **Dart**: 3.8.0+ - Lenguaje de programación
- **Provider**: ^6.1.5 - Gestión de estado reactivo
- **Material Design 3**: Sistema de diseño UI

Backend

- **Firebase Firestore**: Base de datos NoSQL en tiempo real
- **Firebase Authentication**: Sistema de autenticación
- **Firebase Security Rules**: Control de acceso

Librerías Adicionales

dependencies:

```
table_calendar: ^3.2.0 # Componente calendario  
confetti: ^0.8.0      # Animaciones  
intl: ^0.20.2        # Internacionalización
```

Esquema de clases:

```
lib/
  main.dart      # Punto de entrada
  └── Init        # Clase de inicialización
  models/
    model.dart    # Model (Provider principal)
    appointment.dart # Appointment
    service.dart   # Service
    product.dart   # Product, Category, CartItem, ItemsOrder, OrderItem
    saloninfo.dart # SalonInfo
  pages/          # Pantallas UI
    auth/          # Autenticación
      auth_check.dart # AuthCheck
      login.dart     # Login
      register.dart  # Register
      recover_account.dart # RecoverAccount
      auth_page.dart # LoginRegister
    client/         # Interfaz cliente (16 clases)
      appointment.dart # AppointmentsScreen
      appointment_confirmation.dart # AppointmentConfirmationScreen
      cart_screen.dart # CartScreen
      checkout_screen.dart # CheckoutScreen
      date_selection.dart # DateSelectionScreen
      edit_profile.dart # EditProfileScreen
      my_orders_screen.dart # MyOrdersScreen
      orders_confirmation_screen.dart # OrderConfirmationScreen
      product_details_screen.dart # ProductDetailScreen
      profile.dart   # ProfileScreen
      search_products_screen.dart # SearchProductsScreen
      service_selection.dart # ServiceSelectionScreen, ServiceCard
      store_screen.dart # StoreScreen
      time_selection.dart # TimeSelectionScreen
    admin/          # Panel administración (13 clases)
      add_edit_product_screen.dart # AddEditProductScreen
      add_edit_services.dart # AddEditServiceScreen
      admin_calendar.dart # AdminCalendarScreen
      admin_create_appointment.dart # AdminCreateAppointmentScreen
      admin_dashboard.dart # AdminDashboard
      admin_services_screen.dart # AdminServicesScreen
      admin_store_dashboard.dart # AdminStoreDashboard
      manage_categories_screen.dart # ManageCategoriesScreen
      manage_products_screen.dart # ManageProductsScreen
      orders_management_screen.dart # OrdersManagementScreen
      time_slot_management_screen.dart # TimeSlotManagementScreen
      salonbook.dart     # SalonBook (pantalla principal)
  resources/       # Utilidades
    validator.dart  # Validator
  widgets/         # Componentes reutilizables
    custom_modal_bottom_sheet.dart # CustomBottomSheet
    modal_bottom_list_item.dart # ModalBottomListItem
  libraries.dart    # Exports de librerías adaptativas
```

Resumen de Clases por Categoría

- **Core:** 1 clase (Init)
- **Modelos:** 8 clases (Model, Appointment, Service, Product, Category, CartItem, ItemsOrder, OrderItem, SalonInfo)
- **Autenticación:** 4 clases (AuthCheck, Login, Register, RecoverAccount, LoginRegister)
- **Cliente:** 16 clases (interfaces y funcionalidades para usuarios)
- **Administración:** 11 clases (panel de control y gestión)
- **Utilidades:** 1 clase (Validator)
- **Widgets:** 2 clases (componentes reutilizables)

Total: 43 clases principales

Base de datos firestore:

Collections Principales

Users

```
users/{userEmail}
{
  "name": "string",
  "email": "string",
  "phone": "string",
  "gender": "Male|Female",
  "role": "client|admin",
  "createdAt": timestamp
}
```

Services

```
services/{serviceId}
{
  "name": "string",
  "description": "string",
  "price": number,
  "duration": number,    // minutos
  "gender": "Male|Female|All",
  "isActive": boolean
}
```

Appointments

```
appointments/{appointmentId}  
{  
  "userId": "string",    // email usuario  
  "serviceId": "string",  
  "date": timestamp,  
  "timeSlot": "string",  // "HH:mm"  
  "status": "pending|confirmed|completed|cancelled",  
  "notes": "string?",  
  "createdAt": timestamp  
}
```

Products

```
products/{productId}  
{  
  "name": "string",  
  "description": "string",  
  "price": number,  
  "images": ["string"],  
  "categoryId": "string",  
  "stockQuantity": number,  
  "isActive": boolean,  
  "isFeatured": boolean  
}
```

Orders

```
orders/{orderId}  
{  
  "userId": "string",  
  "items": [{"productId": "string", "quantity": number, "price": number}],  
  "total": number,  
  "status": "pending|processing|shipped|delivered",  
  "shippingAddress": {},  
  "createdAt": timestamp  
}
```

Índices Requeridos

```
appointments: ["userId", "date desc"]  
products: ["categoryId", "createdAt desc"]  
orders: ["userId", "createdAt desc"]
```

Funcionalidades principales:

Sistema de Autenticación

```
Future<User?> signIn({required String email, required String password}) async {
  try {
    UserCredential userCredential = await auth.signInWithEmailAndPassword(
      email: email, password: password
    );

    bool isAdmin = await isUserAdmin();
    return userCredential.user;
  } on FirebaseAuthException catch (e) {
    handleAuthError(e, context);
    return null;
  }
}

Future<bool> isUserAdmin() async {
  var doc = await fbStore.collection("users").doc(auth.currentUser?.email).get();
  return doc.data()?["role"] == "admin";
}
```

Sistema de Reservas

```
Future<String> createAppointment({
  required String serviceId,
  required DateTime date,
  required String timeSlot,
}) async {
  final availableSlots = await getAvailableTimeSlots(date, service.duration);
  if (!availableSlots.contains(timeSlot)) {
    throw Exception("Time slot not available");
  }

  final docRef = await fbStore.collection("appointments").add({
    "userId": auth.currentUser!.email,
    "serviceId": serviceId,
    "date": Timestamp.fromDate(date),
    "timeSlot": timeSlot,
    "status": "pending",
    "createdAt": FieldValue.serverTimestamp(),
  });

  await _updateTimeSlotsAfterBooking(date, timeSlot, service.duration);

  return docRef.id;
}
```

E-commerce y Carrito

```
void addToCart(Product product, int quantity) {
    if (quantity <= product.stockQuantity) {
        _cartItems.add(CartItem(
            productId: product.id,
            productName: product.name,
            price: product.price,
            quantity: quantity,
        ));
        notifyListeners();
    }
}

Future<String> createOrder({
    required Map<String, dynamic> shippingAddress,
}) async {
    final subtotal = cartTotal;
    final tax = subtotal * 0.21;
    final total = subtotal + tax;

    final docRef = await fbStore.collection("orders").add({
        "userId": auth.currentUser!.email,
        "items": _cartItems.map((item) => item.toMap()).toList(),
        "total": total,
        "status": "pending",
        "shippingAddress": shippingAddress,
        "createdAt": FieldValue.serverTimestamp(),
    });

    clearCart();
    return docRef.id;
}
```

Validaciones

```
class Validator {  
    static String? validateEmail({required String? email}) {  
        if (email?.isEmpty ?? true) return "Email is required";  
        RegExp emailRegExp = RegExp(r"^\S+@\S+\.\S+$");  
        if (!emailRegExp.hasMatch(email!)) return "Invalid email format";  
        return null;  
    }  
  
    static String? validatePassword({required String? password, required bool register}) {  
        if (password?.isEmpty ?? true) return "Password is required";  
  
        if (register) {  
            RegExp strongPassword = RegExp(r"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@#\$&*~]).{8,}$");  
            if (!strongPassword.hasMatch(password!)) {  
                return "Password must be 8+ chars with uppercase, lowercase, number, and symbol";  
            }  
        }  
        return null;  
    }  
}
```

Información del Proyecto

Stack Tecnológico Completo

- **Frontend:** Flutter 3.32.x + Dart 3.8.0+
- **State Management:** Provider 6.1.5
- **Backend:** Firebase (Firestore + Auth + Functions)
- **UI Framework:** Material Design 3
- **Build Tools:** Android Studio, Xcode
- **Deployment:** Google Play Store, Apple App Store

Estructura de Datos Resumida

- **7 Collections principales:** users, services, appointments, products, categories, orders, timeSlots
- **2 Roles de usuario:** client, admin
- **5 Estados de citas:** pending, confirmed, completed, cancelled, no-show
- **5 Estados de pedidos:** pending, processing, shipped, delivered, cancelled

Métricas de la Aplicación

- **Código:** ~15,000 líneas Dart
- **Pantallas:** 25+ screens
- **Funcionalidades:** Autenticación, Reservas, E-commerce, Administración
- **Compatibilidad:** Android 7.0+, iOS 12.0+

Contacto y Soporte

- **Desarrollador:** Jose Mateo Romero
- **Email:** otema28@gmail.com
- **Licencia:** GNU General Public License v3.0
- **Repositorio:** GitHub - SalonBook

C) Licencia de uso

La licencia elegida para la aplicación móvil SalonBook es la GPL v3.0, esta aplica a todas las versiones ya creadas o futuras.

Términos de la licencia mencionada:

- Derechos de uso: el uso, modificación y distribución de software queda sesgada a los términos de la licencia y el incumplimiento del mismo puede derivar en penalizaciones.
- Distribución del Código Fuente: En caso de distribuir el software SalonBook o cualquier componente del mismo, tanto en su versión original como modificada, será obligatorio facilitar el código fuente correspondiente de forma accesible y en formato apropiado. Es imprescindible incluir una copia de esta licencia acompañando al software distribuido.
- Condiciones para Modificaciones: Cualquier alteración efectuada al software SalonBook debe estar debidamente documentada, especificando de manera clara que el software ha sido objeto de modificaciones. Del mismo modo, toda distribución de las versiones modificadas quedará sujeta a las disposiciones establecidas en esta licencia.
- Compatibilidad de Licencias: Cuando se combine o distribuya el software SalonBook junto con otros programas que posean licencias incompatibles, será necesario adoptar las medidas pertinentes para asegurar la compatibilidad entre dichas licencias.
- Exención de Garantías: El software SalonBook se suministra "en su estado actual", sin ningún tipo de garantía, ya sea explícita o implícita, incluyendo pero no limitándose a las garantías de comercialización, aptitud para un fin específico y ausencia de violación de derechos. El usuario acepta todos los riesgos asociados con la utilización del software.

- Limitación de Responsabilidad: Bajo ninguna circunstancia los autores o propietarios de los derechos de autor serán responsables de daños directos, indirectos, fortuitos, especiales, ejemplares o derivados (incluyendo, sin limitación, pérdida de información, interrupción de actividades comerciales, o pérdida de ganancias) que resulten del uso o la incapacidad de utilizar el software SalonBook, aun cuando se haya notificado sobre la posibilidad de dichos daños.
- Finalización: Esta Licencia permanecerá vigente hasta que se produzca el incumplimiento de cualquiera de los términos y condiciones aquí establecidos. En caso de violación, los derechos concedidos por esta Licencia cesarán de forma automática y se deberá suspender inmediatamente todo uso y distribución del software SalonBook.

6. Colorario

A) Posibles mejoras

Actualmente el proyecto está muy depurado y tiene muchas funcionalidades complejas que han requerido tiempo para el aprendizaje de las mismas y la implementación correcta, aún así si el proyecto se llevara a un uso real me gustaría implementar algunos apartados más y ciertas mejoras:

1. Inicio de sesión con cuenta de google: no es difícil implementarlo y agiliza el proceso para los usuarios siendo más dinámico el inicio de sesión y el registro, también disminuimos el posible número de usuarios que necesitan recuperar la contraseña.
2. Añadir varios idiomas: actualmente todas las pantallas están únicamente en inglés ya que es el idioma más usado pero si la aplicación llegara a usarse se debería de añadir como mínimo el español, esto no significa reemplazar el inglés por el español si no de tener la opción de elegir el idioma, o incluso que depende del idioma que use el dispositivo se elija el idioma automáticamente. Este cambio permitiría que un gran número de usuarios se sientan cómodos usando la aplicación, este cambio se puede realizar usando la librería propia de flutter: [flutter_localizations](#).
3. Pagos: como se comentó anteriormente, actualmente no se pueden hacer pagos en la aplicación, tenemos una ventana donde se puede introducir los datos de la tarjeta y tal pero es una pasarela para poder ir al siguiente paso no causa ningún efecto, por lo cual, en el caso de llevar la aplicación al uso habría dos opciones reemplazar la pasarela de pago mencionada por un único medio de pago que sería el pago en efectivo o bien implementar la API de PayPal, es sencillo de vincular, también hay otros plugins que se pueden añadir para realizar pagos.
4. Doble factor de autenticación: Es útil ya que le complica el inicio de sesión a una persona no autorizada a la cuenta de un cliente, para ello los usuarios deberían además de introducir correo y contraseña un segundo factor que puede ser recibir un código por mensaje o por una aplicación de autenticación como lo es Google authenticator, de esta forma se protegen los datos de los clientes.
5. Vincular reseñas reales a la aplicación: Una funcionalidad que puede darle cierta calidad extra a este proyecto sería crear un nuevo apartado de valoraciones donde el cliente pudiera ver las opiniones que los clientes han dejado sobre el establecimiento en Google.
6. Nuevo apartado de cuidados: Este nuevo apartado sería enfocado al correcto uso de los productos, a consejos para el cuidado del pelo, uñas etc. Podrían incluirse videos para informativos.

7. Manejo de notificaciones: Una mejora que sería realmente útil sería el manejo de las notificaciones tanto en la propia aplicación como vía correo electrónico, esto sería útil para que el usuario no tenga que entrar en la propia aplicación para ver si le han confirmado la cita, le llegaría una notificación al correo y a la aplicación.
8. Sistema de referidos: Una funcionalidad que podría añadir mayor calidad a la aplicación es un sistema de bonificación donde los clientes cada x números de servicios obtendrían uno gratuito o bien que al invitar a algún conocido obtenga algún descuento, este tipo de promociones está demostrada que aumentan el número de clientes y la regularidad en la que acuden

B) Viabilidad

SalonBook tiene todo para ser un proyecto exitoso: es sólido a nivel técnico, viable económicamente y está bien alineado con las necesidades del mercado.

Desde el lado técnico, la app está construida con Flutter y Firebase, lo que garantiza buen rendimiento y escalabilidad. Su arquitectura modular, un sistema de autenticación seguro, una gestión eficiente de estados y una interfaz moderna basada en Material Design reflejan un desarrollo bien pensado y profesional.

En lo económico, SalonBook abre varias posibilidades para generar ingresos. Por ejemplo, se pueden ofrecer planes de suscripción mensual adaptados al tamaño y necesidades de cada salón. También existe la opción de ganar mediante comisiones por ventas dentro del e-commerce integrado y establecer alianzas publicitarias con marcas de productos de belleza. A eso se suma el potencial de crear colaboraciones estratégicas con proveedores del rubro y academias de estética, formando así un ecosistema completo alrededor del sector.

En cuanto al mercado, las apps para la gestión de salones están en auge, sobre todo después del impulso digital que trajo la pandemia. Los negocios de belleza están buscando herramientas que les ayuden a organizar citas, evitar cancelaciones y aumentar ingresos adicionales. Ahí es donde SalonBook se destaca, al combinar la gestión de reservas con un módulo de comercio electrónico integrado—una propuesta poco común entre sus competidores. Esto le da una ventaja clara para cubrir tanto las necesidades operativas como las oportunidades comerciales de los salones.

7. Bibliografía

En este apartado se recogen todos los artículos/cursos/foros que han sido de utilidad para el desarrollo del proyecto

[01]: <https://docs.flutter.dev>

[02]: <https://docs.flutter.dev/data-and-backend/firebase>

[03]: <https://blog.logrocket.com/tag/flutter/>

[04]: https://pub.dev/packages/firebase_core

[05]: https://pub.dev/packages/firebase_auth

[06]: https://pub.dev/packages/cloud_firestore

[07]: <https://pub.dev/packages/provider>

[08]: <https://pub.dev/packages/intl>

[09]: https://pub.dev/packages/table_calendar

[10]: <https://pub.dev/packages/confetti>

[11]: https://pub.dev/packages/flutter_launcher_icons

[12]: <https://plantuml.com/es/guide>

[13]: <https://plantuml.com/>

[14]: www.udemy.com/course/flutter-the-guide-to-build-android-ios-and-web-apps/

Enlace proyecto Github

<https://github.com/BuhoObservador/salonbook/>

