

**I.E.S LAS SALINAS**



**PROYECTO FINAL FIN DE GRADO**

**CURSO 22-23**

**StockMaster**

**CICLO FORMATIVO GRADO SUPERIOR**

**DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**Autor: José Ángel Sanz Sánchez**

# ÍNDICE

1. Introducción.....	4
1.1. Motivación.....	4
1.2. Abstract.....	4
1.3. Justificación.....	5
1.4. Objetivos propuestos. ....	6
2. Técnicas y herramientas empleadas. ....	6
2.1. Lenguajes de programación. ....	6
2.2. Herramientas utilizadas.....	6
2.3. Sistema operativo utilizado.....	7
3.Estimación de recursos y planificación.....	7
3.1. Requisitos funcionales.....	8
3.2. Objetivo principal. ....	9
3.3. Objetivo secundario.....	9
3.3.1. Estructura de carpetas.....	9
3.3.2. Diseño de vistas. ....	10
3.3.3 Empleo y conexión de las distintas librerías de Firebase y la propia base de datos . .....	10
3.3.4. Desarrollo de métodos.....	10
3.3.4.1. Registrar usuario.....	10
3.3.4.2. Iniciar sesión.....	11
3.3.4.3 Insertar producto/ vender producto.....	11
3.3.4.4. Modificar. ....	11
3.3.4.5. Eliminar.....	12
3.3.4.6. Buscar. ....	12
3.3.5. Validaciones de campos. ....	12
4.Desarrollo. ....	13

4.1.Tablas en base de datos .....	13
4.2.Diagramas de caso de usos. ....	14
4.3.Vistas.....	14
4.3.1 MainActivity.....	15
4.3.2. HomeActivity. ....	16
4.3.3. Nuevo_producto_Activity. ....	16
4.3.4 MostrarProductoActivity. ....	16
4.3.5. DetallesProductoActivity. ....	17
4.3.6. VentasActivity.....	17
4.3.7. DetalleVentaActivity. ....	17
4.4. Estructura de carpetas. ....	18
4.5 Ejecutable. ....	18
4.6. Despliegue y pruebas. ....	19
5. Conclusiones.....	21
6. Bibliografía.....	22
7. Vías futuras.....	22
8. Glosario. ....	23
9. Anexos.....	24

# 1. Introducción

## 1.1. Motivación

Para la realización del siguiente proyecto se tuvo la idea de realizar una aplicación que facilitara a las pequeñas empresas a gestionar sus productos y ventas mediante el uso de los conocimientos impartidos a lo largo del curso en el entorno que más favorable me resultaba como es Android Studio.

Esta idea de proyecto abarcaba todos los contenidos impartidos en la asignatura de programación y desarrollo multimedia tales como la creación de objetos en una base de datos, que en este caso se trabaja sobre el entorno de Google Firebase; modificación y eliminado de los objetos creados previamente, así como trabajar con imágenes asignadas a un objeto dentro de la BBDD.

Por tanto, el proyecto consistirá en una aplicación que, haciendo uso de una base de datos, un usuario registrado pueda crear, modificar y borrar un producto, así como asignarle una imagen, venderlo y ver todas las ventas realizadas.

## 1.2. Abstract

To carry out the following project, the idea of making an application that would make it easier for small companies to manage their products and sales by using the knowledge imparted throughout the course in the environment that was most favorable to me was Android Studio.

This project idea covered all the content taught in the programming and multimedia development subject, such as the creation of objects in a database, which in this case works on the Google Firebase environment; modification and deletion of previously created objects, as well as working with images assigned to an object within the database.

Therefore, the project will consist of an application that, using a database, a registered user can create, modify and delete a product, as well as assign an image to it, sell it and see all the sales made.

### 1.3. Justificación

Propongo el desarrollo de una aplicación de gestión de inventarios diseñada específicamente para pequeños negocios, con el objetivo de proporcionarles una herramienta intuitiva y eficaz para controlar y optimizar sus inventarios.

Justificación:

Optimización de recursos:

Los pequeños negocios a menudo operan con recursos limitados, y la falta de una gestión de inventarios eficiente puede llevar a problemas como la sobrecompra de existencias, la falta de productos en stock o el almacenamiento inadecuado de los mismos. Una aplicación de gestión de inventarios les permitiría optimizar sus recursos al proporcionar información en tiempo real sobre los niveles de stock, la demanda de productos y la necesidad de reabastecimiento.

Reducción de costos:

Un control preciso de los inventarios puede ayudar a reducir los costos operativos y mejorar la rentabilidad de los pequeños negocios. Con una aplicación de gestión de inventarios, los empresarios podrían identificar productos obsoletos o de baja rotación, lo que les permitiría tomar decisiones informadas sobre descuentos, promociones o incluso la eliminación de dichos productos del inventario.

Mejora del servicio al cliente:

Un factor clave para el éxito de cualquier negocio es la capacidad de proporcionar un excelente servicio al cliente. La falta de un inventario adecuado puede resultar en la incapacidad de satisfacer las demandas de los clientes de manera oportuna, lo que podría afectar negativamente la satisfacción del cliente. Una aplicación de gestión de inventarios permitiría a

los pequeños negocios mantener niveles óptimos de stock, prever la demanda y cumplir con los pedidos de manera eficiente.

Adaptabilidad a las necesidades del negocio:

Cada pequeño negocio tiene sus propias particularidades y requisitos específicos. Una aplicación de gestión de inventarios puede ser diseñada de manera flexible y personalizable para adaptarse a las necesidades particulares de cada negocio.

## 1.4. Objetivos propuestos

Como se ha mencionado anteriormente, el objetivo de la aplicación será que, a través de una base de datos, un usuario dado de alta (el personal de la tienda en este caso) sea capaz de poder registrar un producto, el podrá adjuntarle una imagen, modificarlo, borrarlo o venderlo. Dichas ventas también podrán ser visualizadas por el usuario y todos los cambios anteriores aparecerán en la base de datos.

## 2. Técnicas y herramientas empleadas.

### 2.1. Lenguajes de programación

Java ha sido el lenguaje de programación por excelencia empleado para hacer el proyecto, tanto para la parte lógica de la aplicación como la conexión a la base de datos.

### 2.2. Herramientas utilizadas.

- **IDE Android Studio API 32 (Android 5.0 Lollipop):** elegí este IDE porque el que más hemos utilizado en estos años en la asignatura de Programación Multimedia y Dispositivos Móviles (PMDM) y en el que me siento más cómodo a la hora de realizar proyectos de esta envergadura.
- **Firebase Authentication:** Con este módulo de Firebase he realizado el registro y la autenticación de usuarios en la aplicación. Lo he implementado debido a su fácil uso e implementación dentro de la aplicación.

- **Firebase Realtime Database:** he empleado este módulo de Firebase a modo de base de datos en la nube. Elegí esta y no su alternativa Firestore debido a que su implementación me parecía más sencilla y además se sincroniza con el cliente conectado.
- **Firebase Storage:** Implementando este módulo, he podido almacenar en la nube todas las imágenes empleadas en la aplicación al registrar o modificar los productos.
- **Github:** utilicé esta herramienta para tener un repositorio en la nube que me permita tener una dinámica de trabajo más eficiente en la que pueda realizar cambios en el código de la aplicación, hacer ramas, pruebas de implementaciones y retroceder a un punto de guardado anterior con mayor facilidad. Dejo el enlace del proyecto: [https://github.com/JoseAngelSS/TFG\\_JoseAngel\\_FINAL.git](https://github.com/JoseAngelSS/TFG_JoseAngel_FINAL.git)
- **Venngage:** Para la elaboración de los diagramas de Gantt.
- **Microsoft Word:** Para la elaboración de esta memoria.
- **Creately:** Para hacer los diagramas de caso de uso.

### 2.3. Sistema operativo utilizado

Decidí utilizar el sistema operativo Windows en su versión 10 debido a que es el sistema operativo con el que más familiarizado estoy a la hora de trabajar.

## 3. Estimación de recursos y planificación

El tiempo que se estimó necesitar para realizar la aplicación es de alrededor de 50 horas, este tiempo lo dividiré en las diferentes fases de desarrollo que he propuesto para llevar a cabo el proyecto (análisis y planificación, diseño de la aplicación, codificación y pruebas)

De manera inicial estimamos que necesitaremos:

- **Análisis y planificación:** Se dedicará alrededor de unas 5 horas para el análisis y planificación iniciales.
- **Diseño de la aplicación:** Se dedicará alrededor de unas 15 horas para elaborar el diseño de la aplicación.
- **Codificación y pruebas:** Se dedicará alrededor de unas 30 horas en elaborar toda la programación de la aplicación, así como de la realización de las pruebas unitarias.

# Stock Master

## Inventory manager service app

2023-2024

### TIEMPO EMPLEADO TOTAL

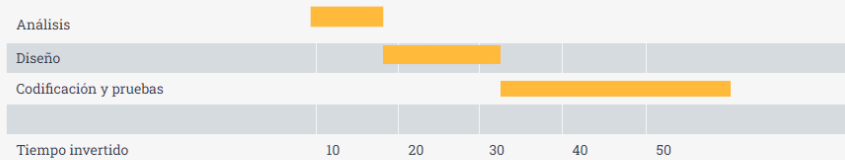


Diagrama de Gantt tiempo empleado total

# Stock Master

## Inventory manager service app

2023-2024

### ESTIMACIÓN TIEMPO INICIAL

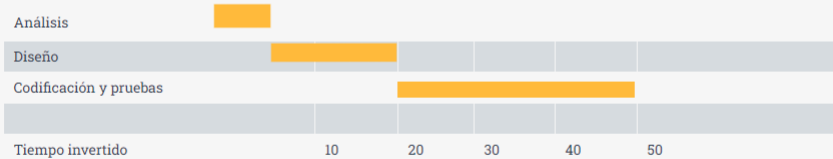


Diagrama de Gantt tiempo empleado inicial

## 3.1 Requisitos funcionales

El usuario debe ser capaz de:

- Usar las opciones de inicio de sesión o registro de usuario para poder acceder al menú principal de la aplicación.
- Utilizar los botones de la ventana "Home" para moverse entre las diferentes pantallas de la aplicación o cerrar sesión.
- En la ventana "Nuevo producto" se debe poder interactuar y rellenar los campos del componente, así como poder introducir una imagen existente en la base de datos.



- En la ventana “Mostrar producto” se debe poder ver los productos que han sido creados, deben poderse seleccionar y la aplicación debe redirigir al usuario a la ventana “detalles componente”.
- En la ventana “detalles componente”, el usuario debe poder ver los datos del producto y tras haber rellenado los campos opcionales, debe ser capaz de poder modificarlo, borrarlo o venderlo. En caso de este último, el usuario debe ser redirigido a la ventana “ventas”.
- En la ventana “ventas” el usuario debe poder visualizar los componentes vendidos y al igual que se ha hecho anteriormente, al seleccionar una venta, el usuario será redirigido a la ventana “detalles ventas”.
- En la ventana “detalles venta” el usuario debe poder ver los detalles de la venta, volver a la pantalla anterior o devolver el producto (borrar la venta de la base de datos).

### 3.2 Objetivo principal.

El objetivo principal es poder desarrollar una aplicación que sea capaz de ayudar al personal de una tienda a gestionar el inventario de la misma con mayor facilidad

- Facilitar el manejo del programa.
- Agilizar la gestión de los distintos componentes.
- Facilitar la visión del empleado sobre las ventas realizadas.
- Mejorar la eficiencia de la empresa cliente.

### 3.3 Objetivos secundarios

Estos son los objetivos que han ido apareciendo en el desarrollo del proyecto:

#### 3.3.1 Estructura de carpetas.

Lo primero que se necesita es estructurar el proyecto con carpetas, para localizar los archivos más fácilmente y que le sea más sencillo al cliente buscar un archivo específico dentro de la aplicación.

### 3.3.2 Diseño de las vistas.

El siguiente paso es diseñar las vistas para poder ver como se introducen, modifican y borran datos e imágenes.

### 3.3.3 Empleo y conexión de las distintas librerías de Firebase y la propia base de datos.

A continuación, se realiza la conexión con la base de datos se importan las librerías y se crean los métodos de Firebase, tablas y clases (Componente, Venta y Usuario).

### 3.3.4 Desarrollo de métodos.

Una vez conectada la base de datos y creadas las tablas se pueden desarrollar métodos para la inserción de datos en la propia base de datos.

Se generan los siguientes métodos:

#### 3.3.4.1 Registrar usuario

En el método “registrar()” tenemos:

- Declaración de variables.
- Condiciones de validación de campos vacíos.
- Condiciones de validación de la longitud de la clave.
- Consulta a la BBDD para validar la existencia del correo:
  - Si el correo existe, notifica al usuario de que ya existe ese una cuenta con ese email.
  - Si no existe, procede a crearlo dentro de una instancia “Usuarios” en la base de datos (Si dicha instancia no existe, la crea) no sin antes pasar las condiciones de validación previamente declaradas.

#### 3.3.4.2 Iniciar sesión.

En el método “login()” tenemos:

- Declaración de variables.

- Condiciones de validación de campos vacíos.
- Autenticación de usuario, si existe, da paso a la siguiente pantalla; si no existe o hay algún problema, devuelve error.

### 3.3.4.3 Insertar producto/ vender producto.

En el método “new\_product()” tenemos:

- dataSnapshot() verifica si en la base de datos existe el id del componente creado, si existe aloja error; si no existe prosigue con la creación del objeto.
- Creación de instancia donde alojará los datos en Firebase (compshasmap o ventashasmap).
- Obtención de los datos de los “EditText” y creación de un objeto con dichos datos.
  - Guarda el objeto en la BBDD usando el “idComp” a modo de clave para identificarlo (“idVenta” en caso del objeto venta).

### 3.3.4.4 Modificar.

En el método “detalles\_editar\_componente” tenemos:

- Declaración de variables y enlace con los elementos de la parte gráfica.
- Creación de una lista con los elementos del componente.
- Obtención de la instancia de Firebase Realtime Database y de la referencia apodada “myRef”.
- Si el producto es modificado, busca el “id\_previo” del componente y lo reemplaza por el actual (reescribe el componente con los nuevos campos). Realiza el mismo proceso con la imagen.

#### 3.3.4.5 Eliminar.

En los métodos “detalles\_borrar\_componente()” y “devolver()” tenemos:

- Declaración de variables y enlace con los elementos de la parte gráfica.
- Se crea una lista con el objeto que reúne las variables declaradas anteriormente.
- Se establece condición, si el nombre previo es igual al actual, hace referencia a la instancia de la base de datos y elimina el producto. En caso contrario, devuelve un error.
- Si elimina el producto, busca en la carpeta la imagen correspondiente y la elimina

#### 3.3.4.6 Buscar producto/ venta.

En el método “buscarProducto()” o “buscarventa()” tenemos:

- Instancia el objeto de la interfaz, obtiene el texto introducido en la variable y lo guarda en la variable “texto”.
- Hace referencia a la instancia de la BBDD(myRefproductos1).
- Se crea un “ArrayList” del objeto (componente o venta) para almacenar las claves de los nodos.
- Busca en la instancia un objeto con el id introducido

#### 3.3.5 Validaciones de los campos.

En las vistas de registro, iniciar sesión, insertar producto y modificación se validarán los campos insertados, si alguna validación no se cumple, no se podrá realizar el registro.

## 4 Desarrollo.

### 4.1 Tablas en base de datos.

Lo primero que se necesita para que la aplicación pueda llevar el registro de componentes es una base de datos con tablas para: Venta y componentes.

Sus atributos son:

-Componentes (insertar, modificar, borrar):

- IdComp (Clave principal)
- idMarca (nombre de la marca)
- Cantidad (stock)
- Nombre
- Precio

-Ventas (Insertar, modificar borrar):

- idCompvend (Fk)
- Referencia (Ref, clave principal).
- Unidad

-Usuario:

- Nombre.
- Correo.
- Clave.

### 4.2 Diagramas de caso de usos

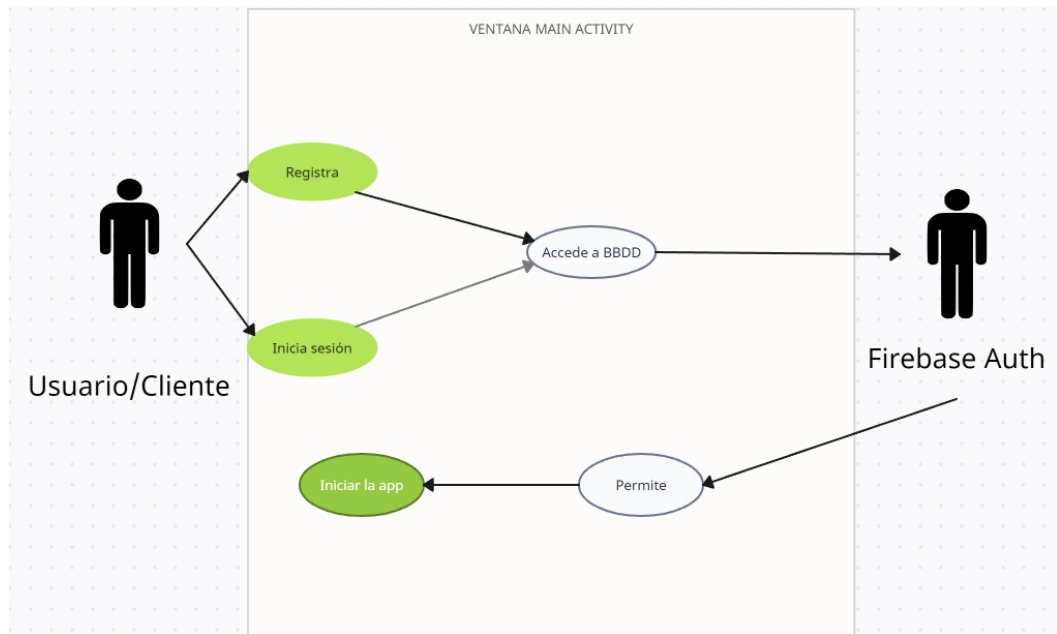


Diagrama de caso de uso ventana principal

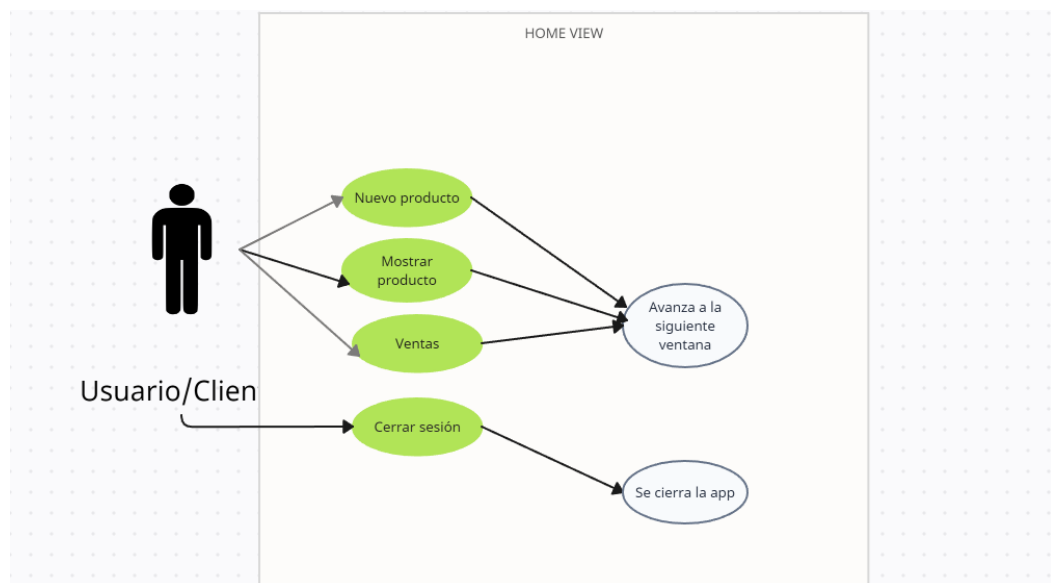


Diagrama de caso de uso ventana Home

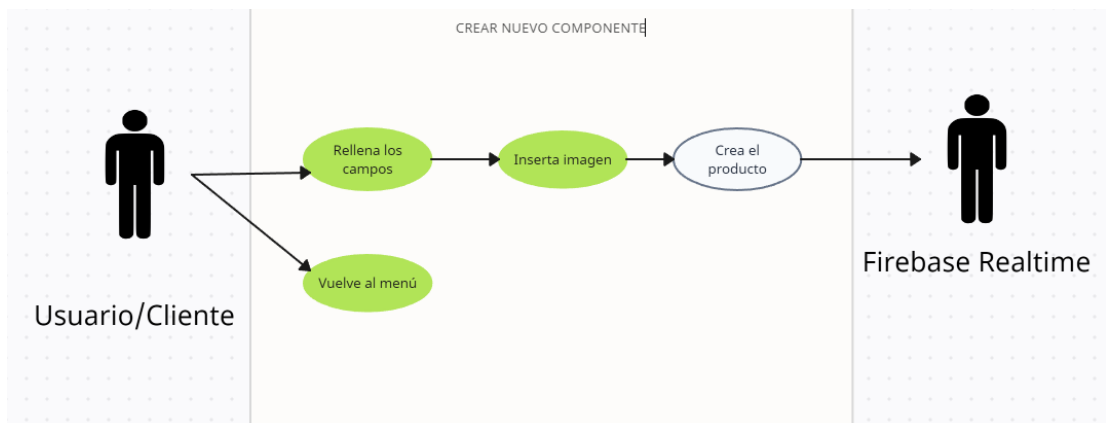


Diagrama de caso de uso ventana "Crear componente"

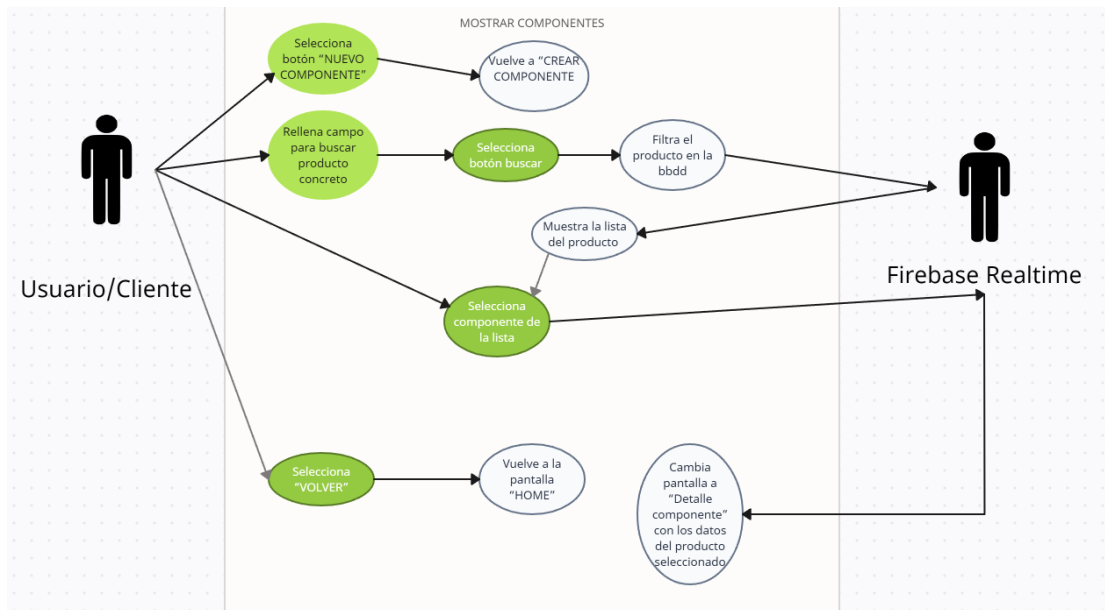


Diagrama de caso de uso ventana “Mostrar componentes”

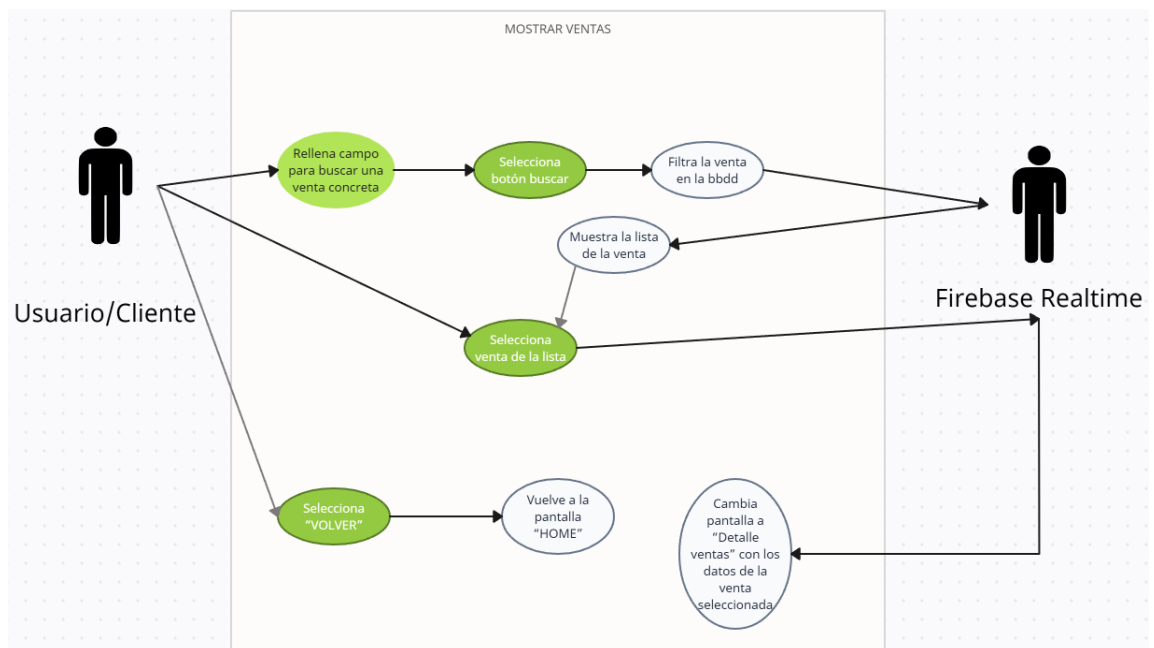


Diagrama de caso de uso ventana “Mostrar ventas”

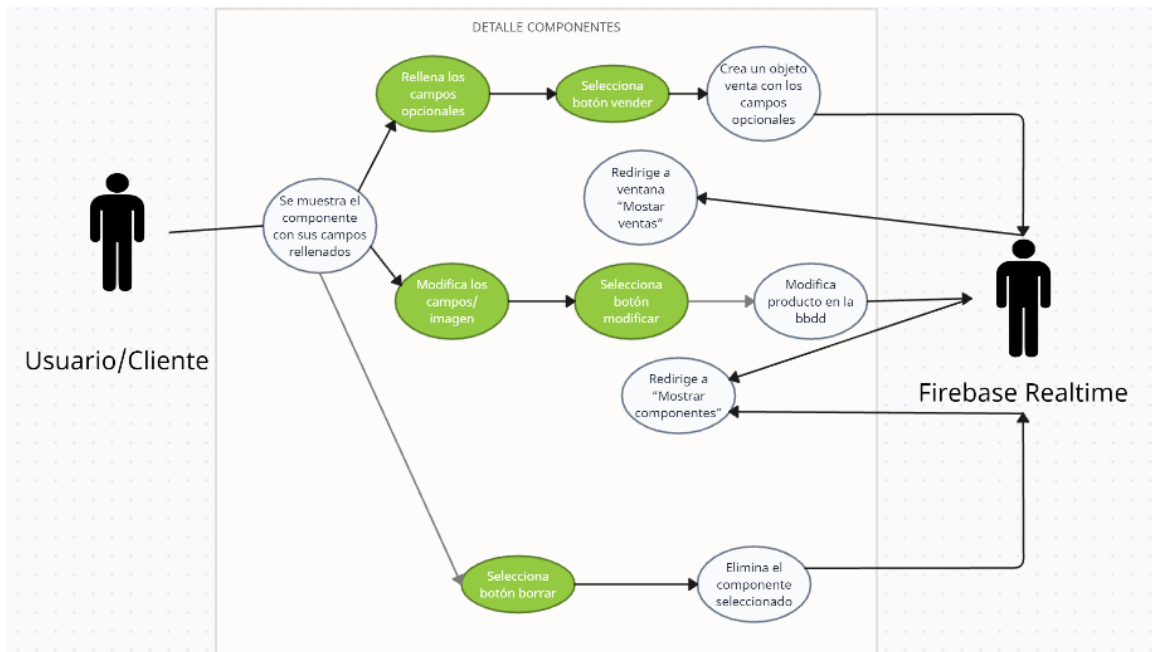


Diagrama de caso de uso ventana “Detalle componentes”

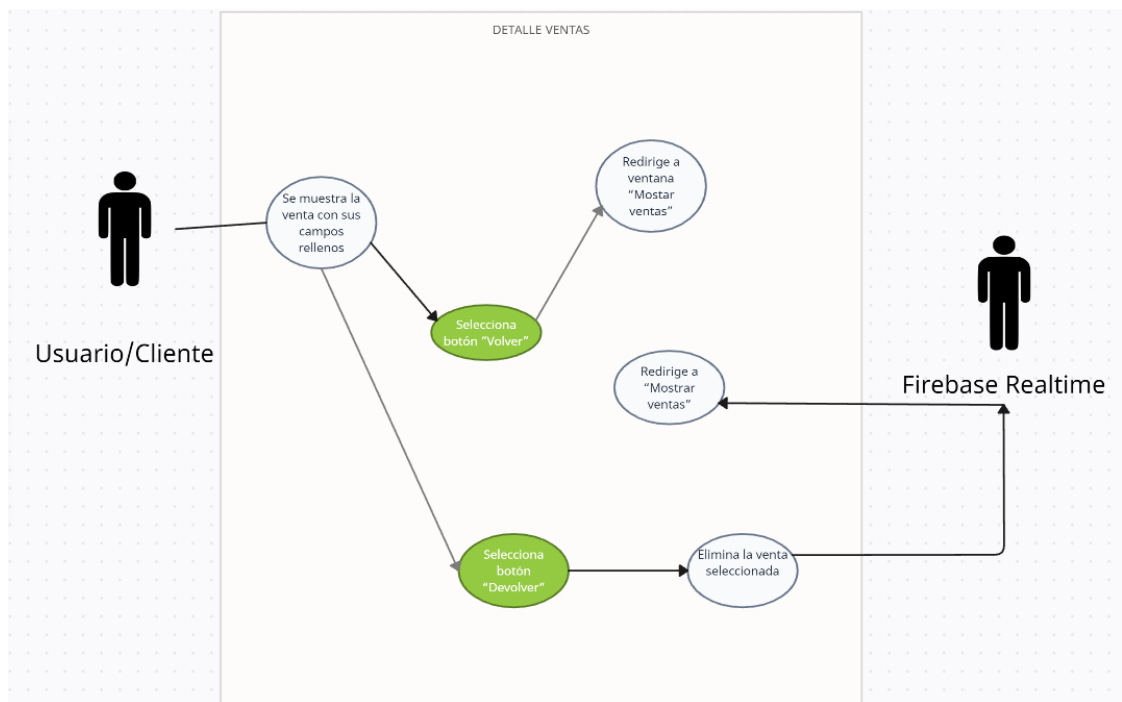


Diagrama de caso de uso ventana “Detalle ventas”

### 4.3 Vistas.

En cada una de las vistas se llevarán a cabo: Validaciones, Comandos, Métodos, etc. Las vistas son las siguientes:



### 4.3.1 MainActivity.

MainWindow será la ventana en la que están alojados las clases fragment de registro e inicio de sesión. Está formado por:

- Un ImageView que establece la imagen de fondo de la aplicación.
- Un LinearLayout vertical para colocar los elementos.
- Un tableLayout formado por dos Table Items que mostrará permitirá cambiar entre cada uno de los fragments de inicio de sesión y registro de usuario.
- Un ViewPager desde donde podremos interactuar con los fragments anteriormente mencionados.

Cada fragment a su vez está formado por los EditText necesarios para introducir los datos del usuario y un botón que será el que active el método de registro o de inicio de sesión.

### 4.3.2 HomeActivity.

Esta será la vista se iniciará inmediatamente después de que se autentifique el usuario. Esta ventana consta de cuatro botones, tres de ellos redireccionarán al usuario a las diferentes pantallas de las que consta la aplicación, el cuarto cerrará sesión y reconducirá al cliente a la pantalla “MainActivity”.

### 4.3.3 Nuevo\_producto\_activity.

Desde esta ventana se crea el componente a través de los cinco EditText donde el usuario deberá introducir los datos del producto. Podrá añadir una imagen desde el ImageView haciendo click sobre este y eligiendo la foto que desee.

Una vez rellenados los campos y habiendo seleccionado la imagen, llamamos al método new\_product() desde el botón “CREAR”. Antes de insertar el nuevo componente en la base de datos, el método comprobará que no quede ningún campo vacío y en caso de que lo haya, paralizará la operación y mostrará un error por pantalla. Una vez creado el componente, la aplicación nos redireccionará a la pantalla “Home”.

Si el usuario lo prefiere y decide volver a la pantalla anterior sin crear ningún producto, tendría que presionar el botón “VOLVER”.

#### 4.3.4 MostrarProductoActivity.

En esta vista se le mostrará al usuario todos los componentes que han sido creados y que están registrados actualmente en la base de datos.

La ventana está formada principalmente por:

- Un EditText que actuará como cuadro de búsqueda. El usuario podrá buscar un producto específico introduciendo el número de referencia o el nombre del mismo y pulsando el botón “BUSCAR” se le mostrará por pantalla el producto con esas credenciales.
- Un RecyclerView que mostrará los componentes de la base de datos. Dicho producto aparecerá con su imagen y características. Al hacer clic sobre uno de los productos, la aplicación reconducirá al usuario a otra ventana donde podrá ver los detalles del producto.
- Dos botones adicionales, uno de ellos conducirá al usuario a la ventana de “nuevo\_producto\_activity”, el otro le llevará a la ventana “HomeActivity”

### 4.3.5 DetallesProductoActivity.

Después de haber seleccionado un componente desde la vista de `MostrarProductoActivity`, nos redirige a esta ventana. Se mostrarán una serie de `EditText` con los datos del componente, que el usuario podrá modificar libremente y otros opcionales que solo se deben rellenar en caso de que dicho componente se vaya a vender. El usuario dentro de esta ventana tendrá tres cursos de acción: modificar los datos del producto como bien se ha mencionado cambiando a su parecer los datos del producto, borrar el producto seleccionado o venderlo habiendo rellenado previamente los campos opcionales.

### 4.3.6 VentasActivity.

Similar a la ventana “`MostrarProductoActivity`”, con la diferencia de que no hay un botón para volver a una ventana de “crear”.

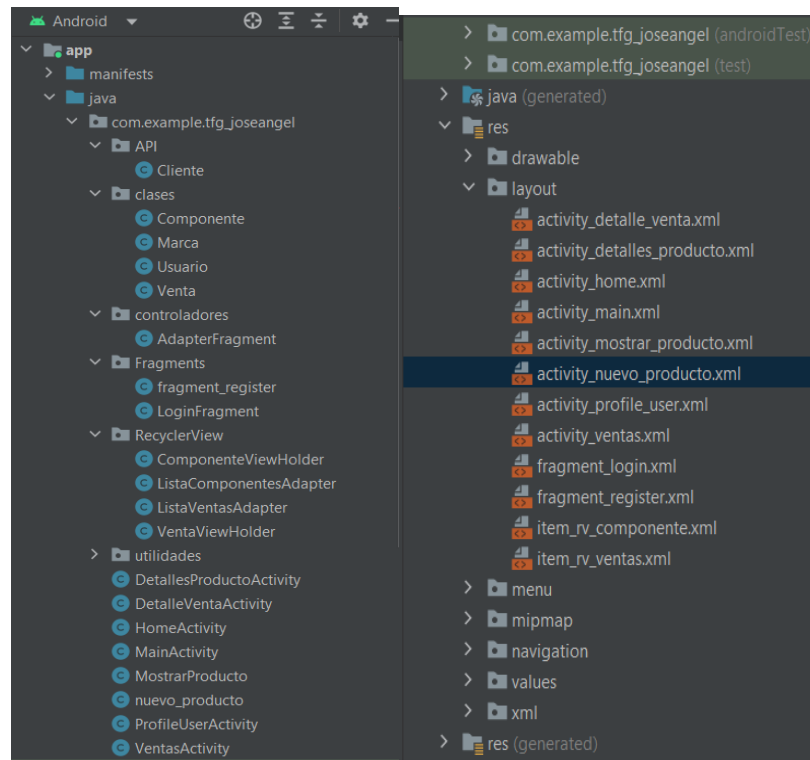
La ventana consta de un cuadro de búsqueda que filtra por referencia de la venta y un `RecyclerView` que mostrará todas las ventas actuales o en caso de haber buscado una específica, la venta que coincida con la referencia.

### 4.3.7 DetalleVentasActivity.

La función de esta ventana es la misma que la de `DetallesProductoActivity`. Una serie de `EditText` con los datos de la venta y dos botones, uno para devolver el producto (emplea el mismo procedimiento que el método borrar de la ventana antes mencionada) y otro de volver a vista de “`VentasActivity`”.

## 4.4 Estructura de carpetas.

El proyecto está estructurado por carpetas, para que resulte más fácil localizar el/los archivo/s que busquemos.



Vista de carpetas del proyecto

## 4.5 Ejecutable.

Se ha generado el ejecutable, que se encontrará en la carpeta API.

## 4.6 Despliegue y pruebas

Nº	ESPECIFICACIÓN DE PRUEBAS
1º	<p><b>Objetivo:</b> Enlazar los fragments a la vista MainActivity y comprobar que cambia la vista.</p> <p><b>Pruebas realizadas:</b> Al iniciar la aplicación se comprueba que lo primero que salga es el fragment de registro y que, al pulsar el TabItem, cambie al fragment del Log-in sin que lance ningún tipo de error por consola.</p>
2º	<p><b>Objetivo:</b> Comprobar que inicie o registre un usuario con éxito</p> <p><b>Pruebas realizadas:</b> Tras programar los métodos de inicio y registro, se pulsa el botón de ejecución y al introducir las credenciales del usuario, debe cambiar a la vista “HomeActivity”. Se comprueba en la consola de authentication de Firebase que el usuario haya quedado registrado.</p>

3º	<p><b>Objetivo:</b> Comprobar que cada botón de la vista “HomeActivity” sea funcional</p> <p><b>Pruebas realizadas:</b> Se pulsán todos los botones de la pantalla HomeActivity y se comprueba que cambie a su correspondiente vista.</p>
4º	<p><b>Objetivo:</b> Comprobar que el botón de “Cerrar sesión” de la vista “HomeActivity” sea funcional.</p> <p><b>Pruebas realizadas:</b> Al pulsar el botón, la sesión debe cerrarse y la aplicación debe redirigir al usuario a la vista “MainActivity”.</p>
5º	<p><b>Objetivo:</b> Comprobar que se crea un componente correctamente en la vista “nuevo_producto_activity”.</p> <p><b>Pruebas realizadas:</b> Se rellenan los campos, se introduce la imagen y se le da al botón de crear. La aplicación debe redirigir al usuario a la vista HomeActivity mientras muestra un mensaje por pantalla del éxito de la operación. En el apartado de Firebase “RealtimeDatabase” debe salir la instancia del componente, así como el componente creado. No debe aparecer ningún error en la consola de la aplicación.</p>

Nº	ESPECIFICACIÓN DE PRUEBAS
6º	<p><b>Objetivo:</b> Comprobar la inserción de imágenes</p> <p><b>Pruebas realizadas:</b> En la vista de “nuevo_producto”, al pulsar sobre el ImageView, debe aparecer una opción para adjuntar una imagen ya existente en el dispositivo emulado. Posteriormente, una vez creado el producto, se debe comprobar que en el apartado “Storage” de Firebase, aparece el componente con la imagen. Mismo procedimiento para comprobar el funcionamiento del botón vender de “DetallesProductosActivity”.</p>
7º	<p><b>Objetivo:</b> Comprobar que el funcionamiento de los RecyclerView</p> <p><b>Pruebas realizadas:</b> Se pulsán los botones que llevan a las vistas “VentasActivity” y “MostrarProductoActivity”, En el RecyclerView deberían aparecer los productos/ ventas que han sido creados. Al interactuar con el item, debe redireccionar al usuario a la ventana de “DetallesProductosActivity” o “DetalleVentaActivity”.</p>

8º	<p><b>Objetivo:</b> Comprobar que el botón de modificar funciona.</p> <p><b>Pruebas realizadas:</b> Se editan los editText de “DetallesProductosActivity” a gusto del usuario y al darle al botón, se deben de verse reflejados los cambios en la base de datos de Firebase Realtime Database.</p>
9º	<p><b>Objetivo:</b> Comprobar que el botón de borrar/ devolver funciona</p> <p><b>Pruebas realizadas:</b> Una vez seleccionado un producto o una venta, dentro de las ventanas de “DetallesProductosActivity” o “DetalleVentaActivity” se pulsa al botón eliminar/ devolver y tras un mensaje que confirme el éxito de la operación, se debe comprobar que, en la base de datos, dicho producto o venta han sido eliminados.</p>
10º	<p><b>Objetivo:</b> Comprobar que el botón volver de todas las vistas funciona.</p> <p><b>Pruebas realizadas:</b> Una vez estemos en la ventana correspondiente, al pulsar el botón volver, este nos debe llevar a una vista anterior programada. Dependiendo de la vista en la que esté el usuario, el método volver() del botón le llevara a una vista específica.</p>

## 5. Conclusiones.

El objetivo fue crear una aplicación para la gestión del inventario de un negocio, una vez finalizado el proyecto doy por cumplido los objetivos propuestos y que han ido surgiendo, así como los especificados en el apartado “Objetivos”, habiendo generado así el ejecutable para posterior instalación la aplicación en cualquier dispositivo inteligente.

He encontrado numerosas dificultades a la hora de compatibilizar las versiones de todas las librerías empleadas ya que, determinadas versiones no funcionaban correctamente, eran incompatibles con la versión de Android o con otras librerías o generaban fallos inexistentes por un bug sin depurar. Esto añadido a la gran variedad de variables con diversos nombres que a veces generaban errores debido a la similitud de sus IDs, dificultaban de sobremanera hallar el origen del fallo y por tanto su resolución.

Se deja abierta la posibilidad de continuar con el desarrollo de esta aplicación.

## 6. Bibliografía.

Las webs, repositorios, canales de Youtube, etc. visitados son:

- <https://stackoverflow.com/>
- <https://firebase.google.com/docs/firestore?hl=es-419>
- <https://app.createely.com/d/BHtAtI0g2pk/edit>
- <https://developer.android.com/guide>

## 7. Vías futuras:

Con un poco más de tiempo para aprender y desarrollar la aplicación me hubiera gustado introducir un rol de cliente que tuviera las siguientes funciones:

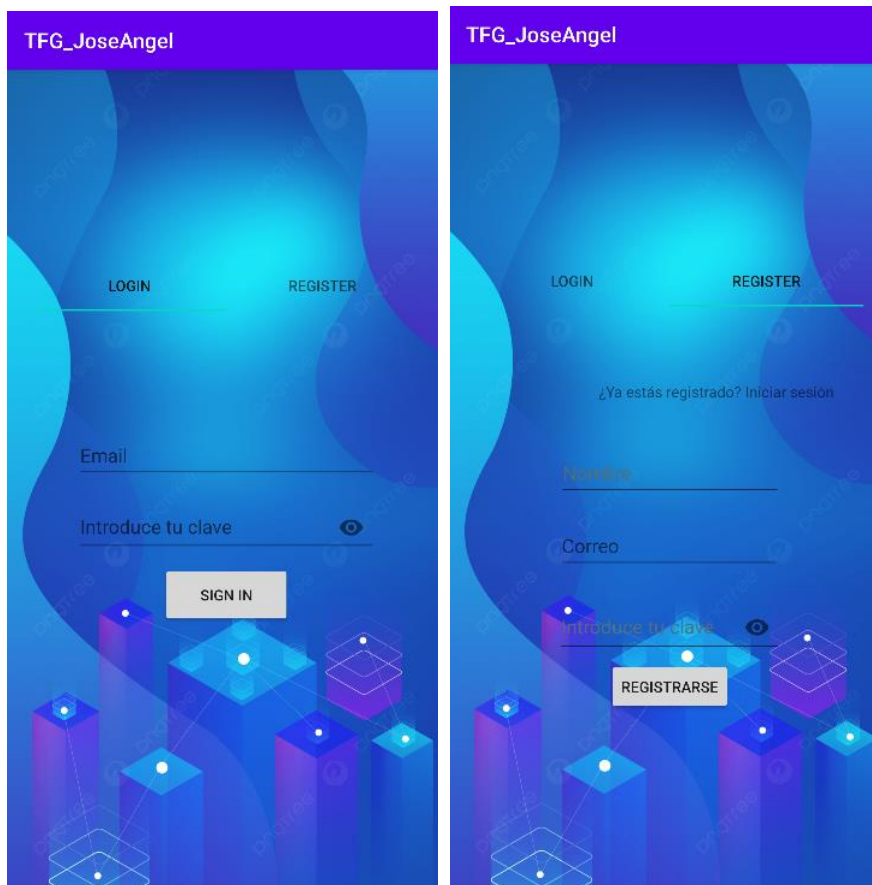
- Registrarse e iniciar sesión
- Tener acceso a un detallado catálogo de la tienda con filtro de búsqueda
- La opción de seleccionar los productos que desee, asignarlos a una lista de favoritos o “carrito” y desde ahí poder comprarlos, generando una factura que se le enviaría por correo.
- Un registro en el cual, el cliente pueda ver su historial de compras y que los empleados de la tienda pudieran ver el historial de compra de todos los clientes.

## 8. Glosario

- Java: Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems.
- IDE: Un entorno de desarrollo integrado o entorno de desarrollo interactivo, es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.
- BBDD: abreviación de base de datos. Una base de datos es Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.
- EditText: TextView cuya apariencia ha sido modificada para actuar como campo de texto, donde el usuario puede editar su contenido para especificar datos en una aplicación Android.

- TextView: un componente que se puede implementar en las vistas de la aplicación en la que estamos trabajando con el propósito de presentarle un texto al usuario.
- RecyclerView: ViewGroup que contiene las vistas correspondientes a tus datos. Se debe implementar como una vista propia y no solo como un componente.
- ViewGroup: es un contenedor invisible que define la estructura de diseño de la vista y otros objetos.
- LinearLayout: grupo de vistas que alinea todos los elementos secundarios en una única dirección, de manera vertical u horizontal. Se puede especificar la orientación.

## 9. Anexo I



Vista MainActivity, fragments de inicio de sesión y registro de usuario

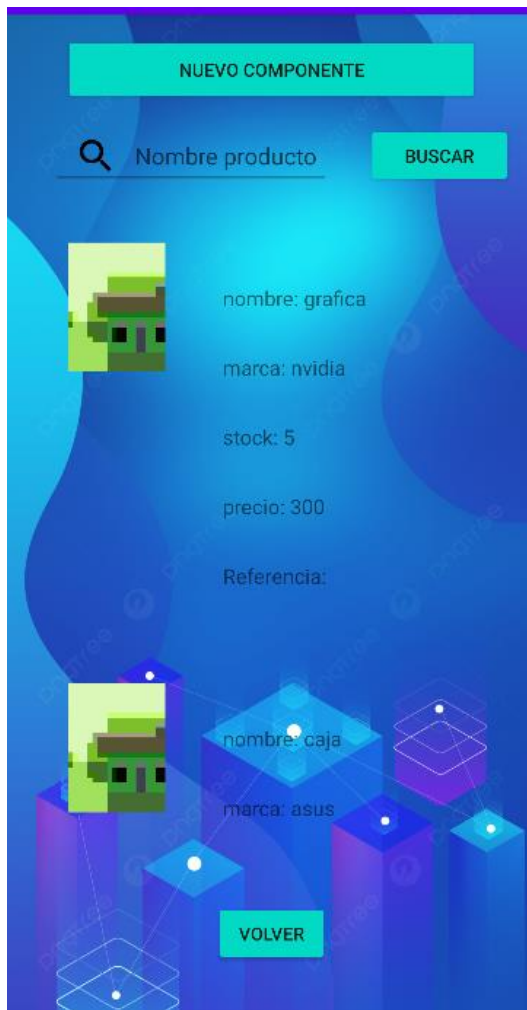




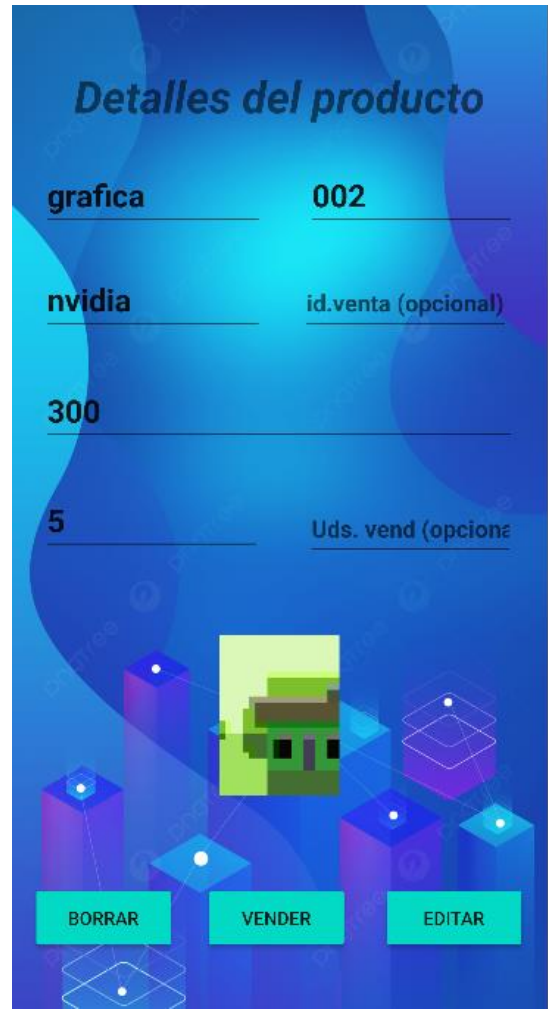
Vista ventana "HomeActivity"

The image shows a mobile application interface for creating a new component. The title at the top is 'CREAR NUEVO COMPONENTE'. Below the title are five input fields, each with a label and a horizontal line for text entry: 'Referencia', 'Nombre', 'Precio', 'Cantidad', and 'Marca'. At the bottom, there are two cyan buttons: 'VOLVER' on the left and 'CREAR' on the right. A large black checkmark icon is positioned in the center of the screen, overlapping the input fields. The background is blue and purple with a 3D bar chart at the bottom. A faint watermark 'proteus' is visible across the screen.

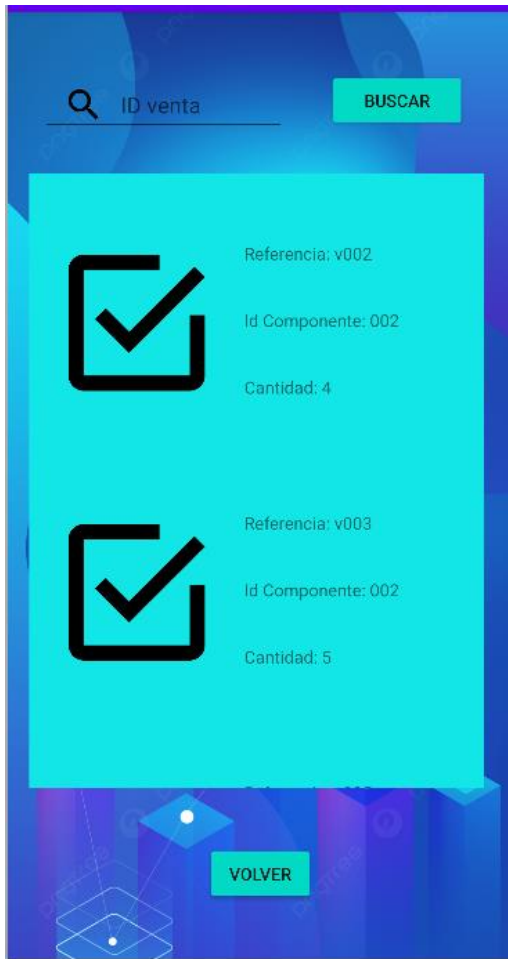
Vista ventana nuevo\_producto



Vista ventana MostrarProductoActivity



Vista ventana DetallesProductoActivity



Vista ventana MostrarVentasActivity



Vista ventana DetalleVentaActivity