



**FORMACIÓN
PROFESIONAL**

PROYECTO DE DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Grupo 12

**Roberto Fernández
Javier Franz
(David Gómez)**

ÍNDICE

Introducción	3
Motivación y objetivos	4
Módulos formativos aplicados	5
Tecnologías y lenguajes utilizados	7
Componentes del equipo y aportación de cada uno de ellos	8
Fases del proyecto	9
Mockups	10
Modelo de datos	13
Funcionamiento. Pantalla de Splash y roles de usuario	14
Funcionamiento. Pantalla de login	15
Funcionamiento. Proceso de compra	16
Funcionamiento. Gestión de pedidos	18
Funcionamiento. Stock	19
Appbar	20
Relaciones lógicas de la base de datos	21
Flujo de datos	22
Conclusiones y mejoras del proyecto	23

INTRODUCCIÓN

Multiclick es una aplicación Android que ofrece una solución integral para una tienda de productos multimedia, abordando tanto las necesidades de los clientes como las del personal de tienda. Su principal objetivo es proporcionar a los clientes una plataforma intuitiva y fácil de usar para navegar a través de diferentes categorías y productos disponibles, y realizar sus compras de manera sencilla.

Para lograr esto, se ha prestado especial atención a la experiencia de usuario, asegurándose de que la interfaz sea atractiva, y fácil de navegar.

También se ha desarrollado un conjunto de funcionalidades específicas para el personal de la tienda. Esto incluye la capacidad de visualizar y gestionar los pedidos pendientes de envío, lo que les permite mantener un seguimiento actualizado del estado de cada pedido. Asimismo, se ha implementado una funcionalidad para visualizar el historial de envíos realizados.

También se ha abordado la gestión del stock. Se ha desarrollado una funcionalidad que permite llevar un control preciso de los productos disponibles en el inventario. Esto ayudaría al personal de tienda a asegurarse de que los productos estén disponibles.

Por ultimo, se presenta también una solución para automatizar el proceso de envío de los paquetes, facilitando la cumplimentación de los formularios pertinentes y agilizando el proceso a través de la utilización de datos y credenciales ya establecidos.

MOTIVACIÓN Y OBJETIVOS

La creciente demanda de compras online, especialmente de productos multimedia y la cada vez mayor necesidad de proporcionar una plataforma eficiente para que los clientes puedan realizar compras de manera cómoda y rápida, han motivado el desarrollo de esta aplicación. Además, el personal de Multiclick requiere de herramientas eficaces para gestionar los pedidos, el stock y el envío de productos. Con la aplicación Multiclick, se busca cubrir estas necesidades y ofrecer una experiencia mejorada tanto para los clientes como para el personal de la tienda.

El desarrollo de esta aplicación persigue, en esencia, tres objetivos:

1. Aplicación de los conocimientos adquiridos: El objetivo principal ha sido poner en práctica los conocimientos y habilidades adquiridos durante el ciclo de desarrollo de aplicaciones multiplataforma. Por otra parte también se aplican conocimientos aprendidos en las empresas en el periodo de prácticas que, en un principio, son conocimientos no aprendidos durante el curso, con el objetivo de aportar mas valor y variedad de soluciones.
2. Desarrollo de una aplicación Android funcional: Este proyecto también tiene como objetivo crear una aplicación Android completa y funcional que permita a los clientes navegar, obtener información de productos de tecnología y realizar pedidos de productos multimedia, y al personal de la tienda gestionar los pedidos, el stock y el envío.
3. Acercamiento al mundo laboral: Por último, uno de los objetivos clave de este proyecto ha sido simular una experiencia cercana al mundo laboral. A través del desarrollo de una aplicación completa y funcional como Multiclick, se ha buscado simular el tipo de proyecto que una empresa podría solicitar a sus desarrolladores.

MÓDULOS FORMATIVOS APLICADOS

Programación:

1. Se utilizan sentencias de control de flujo como if, for, while en los métodos onClick() y onComplete() para controlar el flujo de ejecución en base a condiciones.
2. Se utilizan estructuras de datos compuestas como Bundle, ArrayList, QuerySnapshot para almacenar y manipular conjuntos de datos.
3. Se observa la interrelación entre clases al utilizar herencia y polimorfismo en los métodos onCreate() y onComplete() que pertenecen a las clases AccountType, LoginComprador y VendedorPedidoInfo.
4. Se manejan excepciones mediante la utilización de try-catch en bloques como addOnCompleteListener() para capturar y manejar posibles errores durante la ejecución del código.
5. Se aplica la modificación de la clase ArrayAdapter para satisfacer las necesidades de funcionalidad de aplicación, enfocado a la creación de los ítems en todos los ListView y su funcionamiento con sus respectivos botones.
6. Se utiliza la llamada a las imágenes de los productos a través de la modificación del ImageView pertinente, obteniendo este Id desde los datos de una base ya declarada.
7. Para la creación de las listas de items se parte de un item con características específicas a cada función de cada Activity, usando este item como plantilla y representando los datos, extrayéndolos desde la base de datos.

Bases de Datos:

1. Se utiliza Firebase Firestore para almacenar y recuperar datos en la nube. Esto se evidencia en el código que hace referencia a db.collection("Pedidos") y doc.getString("Articulo"), que indican la consulta y extracción de datos de la base de datos.
2. En la base de datos, se da el caso de reinicio de ciertas colecciones para garantizar la limpieza de estas colecciones. Esto da lugar a que no queden datos residuales durante la lectura de datos que no se alteran dentro de la base de datos.

Lenguajes de Marcas:

1. Se utiliza XML para definir la estructura y los atributos de las interfaces de usuario. Por ejemplo, en los fragmentos de código, se definen vistas como TextView y se establecen sus atributos como android:id, android:layout_width, android:layout_height, etc.

Entornos de desarrollo:

1. Se utiliza Android Studio como entorno de desarrollo para desarrollar la aplicación. También se ha hecho uso de GitHub para el control de versiones y UML y Figma para el modelado UML.

Diseño de interfaces:

1. Se utilizan componentes de interfaz de usuario como TextView, Button y se definen atributos para personalizar su apariencia y comportamiento, como android:textColor, android:layout_width, etc.
2. Se hace uso de herramientas de diseño visual proporcionadas por Android Studio para construir y confeccionar las interfaces de usuario de manera gráfica.

Programación Multimedia y dispositivos móviles: El proyecto está desarrollado para dispositivos móviles Android, lo cual implica la utilización de numerosas tecnologías específicas para dicha plataforma, empezando ya propiamente por Android.

Conocimientos Externos: El proyecto incluye una solución de automatización hecha en la herramienta de RPA “UiPath”, agilizando el proceso cumplimentación de la documentación de envío del pedido a través de “Correos”.

TECNOLOGÍAS Y LENGUAJES UTILIZADOS

Android Studio: Es el entorno de desarrollo integrado (IDE) principal para el desarrollo de aplicaciones Android. Proporciona herramientas y recursos para la creación, depuración y prueba de aplicaciones Android.

Java: Es el lenguaje de programación principal utilizado en el desarrollo de aplicaciones Android. Se utiliza para escribir la lógica de la aplicación, las interacciones con la interfaz de usuario y la manipulación de datos.

XML (Extensible Markup Language): Se utiliza para definir la estructura y los atributos de las interfaces de usuario en Android. Los archivos XML se utilizan para crear diseños de pantalla, definir recursos y establecer propiedades de vistas.

Firebase: Es una plataforma de desarrollo de aplicaciones móviles de Google que proporciona una amplia gama de servicios y herramientas, como bases de datos en tiempo real, autenticación de usuarios, almacenamiento en la nube, notificaciones push, entre otros. En el proyecto, Firebase puede haber sido utilizado para funciones como la autenticación de usuarios y el almacenamiento de datos en la nube.

Figma: Es una herramienta de diseño de interfaces de usuario (UI) y prototipado. Puede haberse utilizado para crear y diseñar los wireframes, mockups o diseños de las pantallas de la aplicación, permitiendo visualizar y compartir las interfaces antes de implementarlas.

Git: Es un sistema de control de versiones ampliamente utilizado en el desarrollo de software. Permite el seguimiento de cambios en el código fuente y la colaboración en equipo de manera eficiente.

UiPath: Es una plataforma de automatización robótica de procesos (RPA) que permite diseñar, desarrollar y desplegar robots de software para automatizar tareas empresariales repetitivas y basadas en reglas, mejorando la eficiencia y productividad de los procesos.

Photoshop: Es una herramienta de edición de imagen. En este caso se ha usado para editar los logos para coincidencia de la paleta de colores.

COMPONENTES DEL EQUIPO Y APORTACIÓN DE CADA UNO DE ELLOS

Javier Franz:

1. Primeros pasos; creación del proyecto de Android Studio, de Firebase y del repositorio.
2. Redacción del anteproyecto.
3. Diseño y creación de logotipos.
4. Creación de la Landing Page.
5. Autenticación de usuarios.
6. Gestión de diferentes logins según el rol de usuario.
7. Documentación de las diferentes fases del proyecto.
8. Redacción de la memoria.

Roberto Fernández:

1. Creación de base de datos en Firebase Database y llenado de datos correspondientes.
2. Diseño y estructuración de la lógica para organizar los datos dentro de la base de datos.
3. Diseño completo de la interfaz de usuario, con paleta de colores, de la aplicación.
4. Creación de la interfaz de usuario (UI) de la sección del comprador de la aplicación,
5. Análisis y redacción de la lógica completa de todas las páginas de la sección del comprador.
6. Creación de la interfaz de usuario (UI) de la sección del vendedor de la aplicación.
7. Elaboración y redacción de la lógica completa de todas las páginas de la sección del vendedor.
8. Diseño y razonamiento del flujo de trabajo de la aplicación.
9. Desarrollo de una solución automatizada para el envío de pedidos a través de la empresa de paquetería “Correos”.
10. Implementación y codificación de la solución de automatización en UiPath.
11. Redacción de la memoria.
12. Incorporación de imágenes en la aplicación correspondientes a todos los productos mencionados en la base de datos, asegurando la correcta visualización y asociación de las imágenes con sus respectivos registros de la base de datos.

David Alejandro Gómez Toca:

1. Creación de UI de Stock.
2. Creación de la lógica de las paginas de stock.
3. Función de control de stock al enviar pedidos.
4. Integración de precio del carrito y visibilidad del mismo en los pedidos (no implementado).

FASES DEL PROYECTO

Planificación del proyecto:

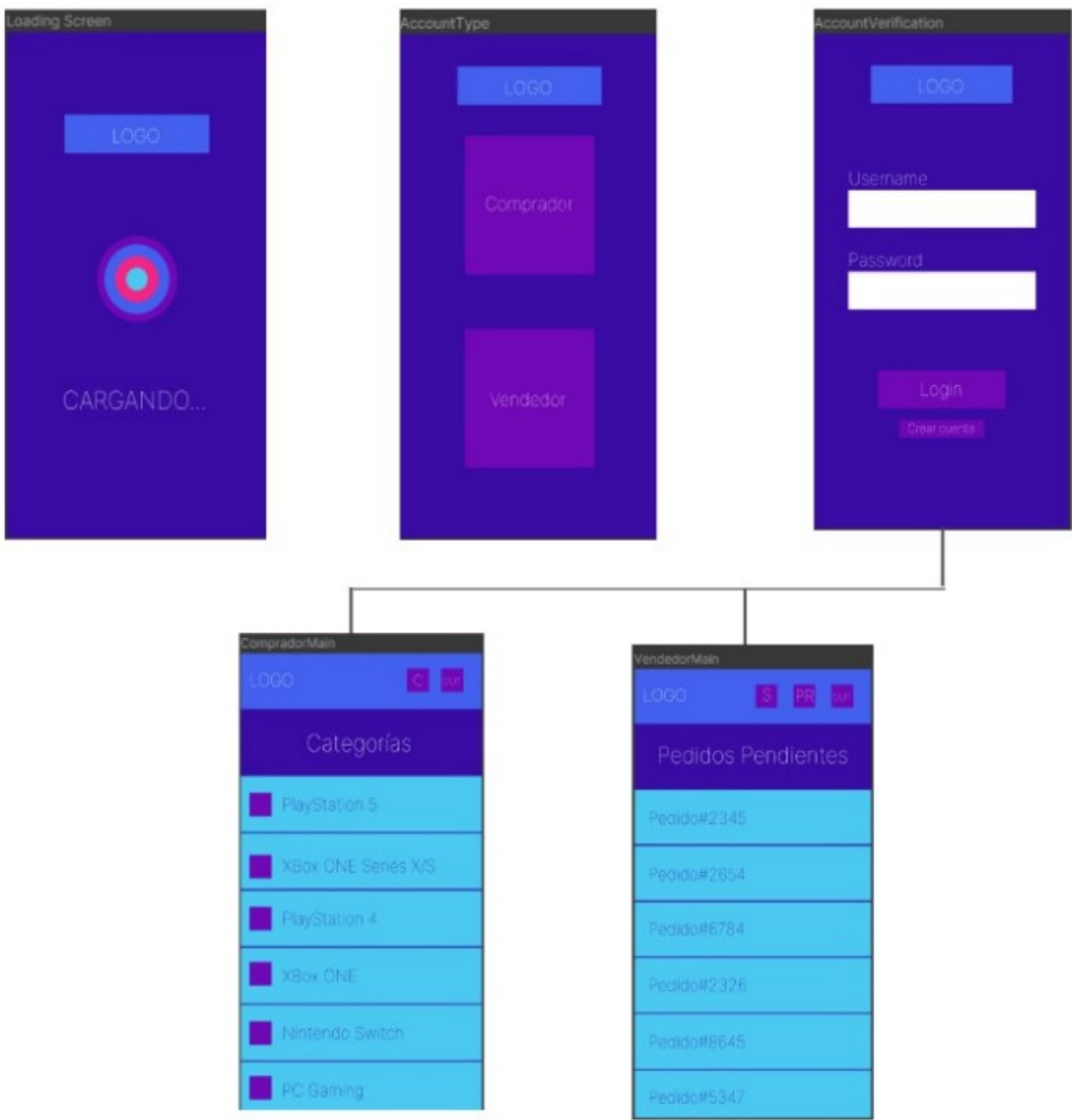
Tras poner en común las bases sobre las que desarrollar el proyecto, establecemos juntos los siguientes sprints, dejando una semana de margen al final para posibles imprevistos.

	FASE DEL PROYECTO	FECHA
	Reunión de planificación	21/3
	Planteamiento y reparto de tareas	2/4
	Anteproyecto definitivo	5/4
	Diseño gráfico	16/4
	Reparto tareas backend	16/4
	Desarrollo backend	7/5
	Unión parte gráfica y backend	14/5
	Fin de la fase de desarrollo	14/5
	Subsanación de errores	21/5
	Presentación de la aplicación	22/5

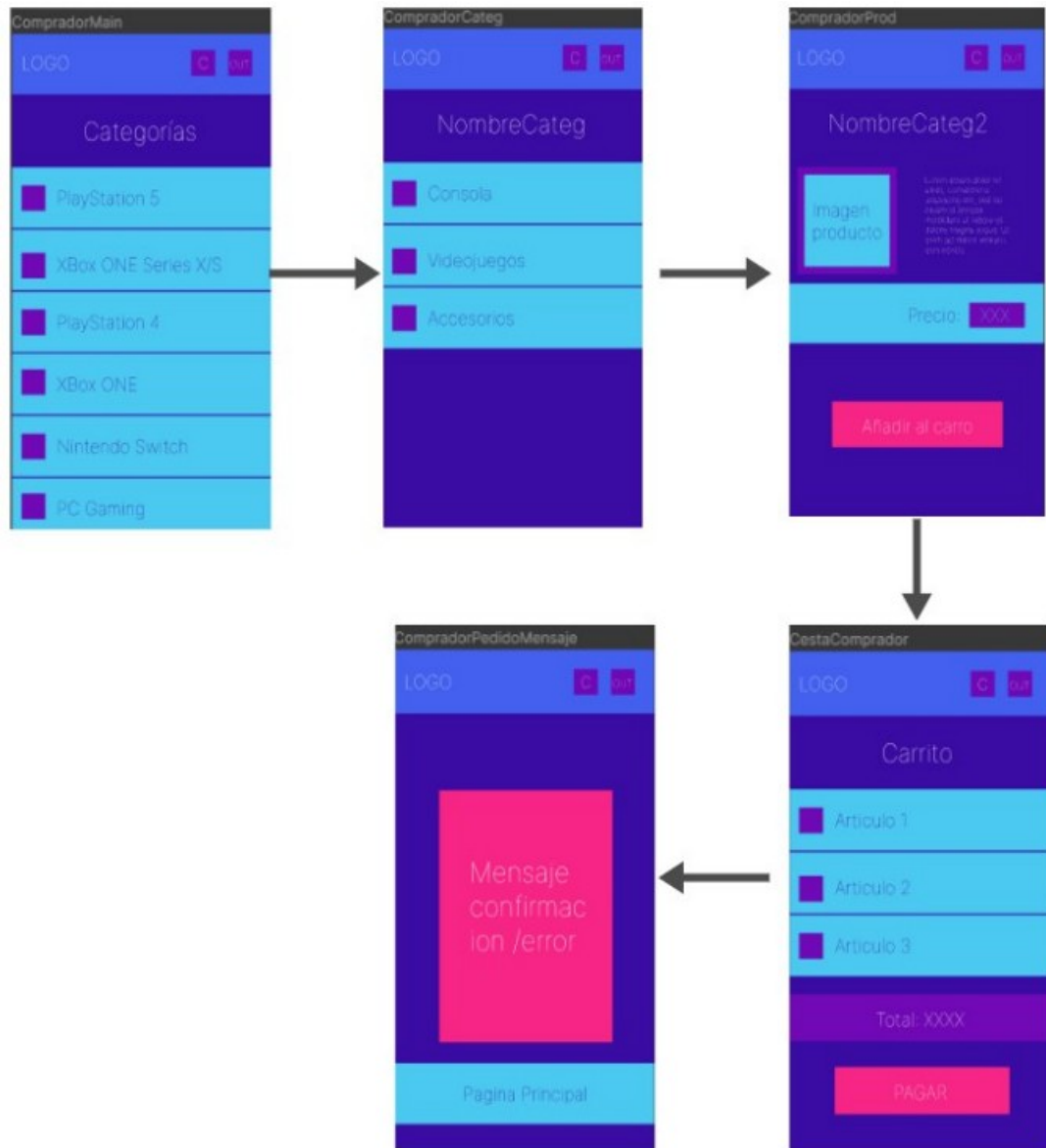
Aunque la planificación inicial no se ajustó completamente a los tiempos estimados, pensamos que ésta ha sido adecuada pues se ha trabajado hasta el último día pero se entrega el proyecto con todas las funcionalidades previstas dentro de las posibilidades del grupo que desarrolla este proyecto y además, se ha podido invertir algo de tiempo en mejorar el aspecto gráfico, lo que implica una buena gestión y compromiso por la mayor parte del equipo.

MOCKUPS:

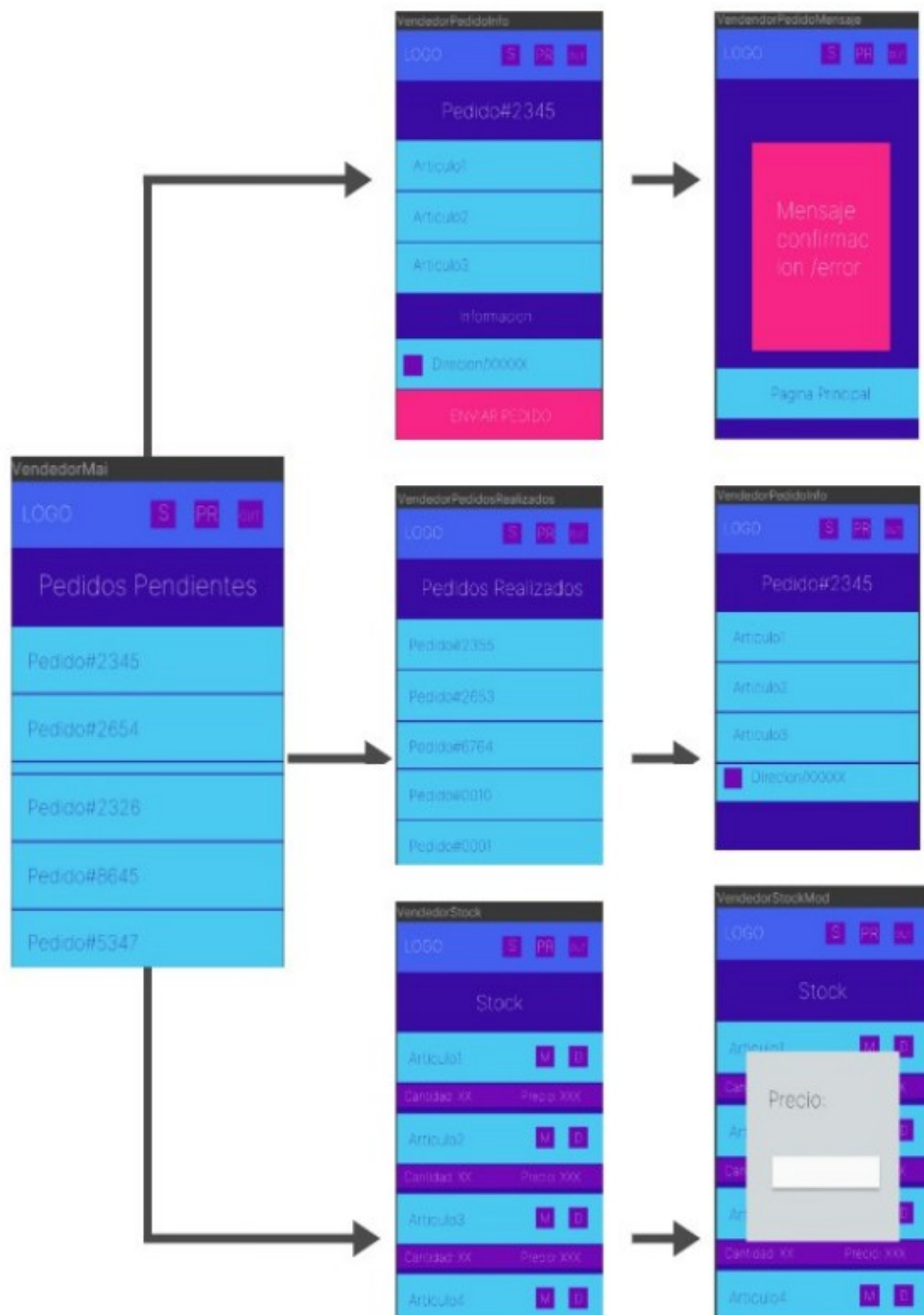
1. Login y roles de usuario:



2. Comprador:








3. Vendedor



MODELO DE DATOS:

Para el modelo de datos hemos utilizado Firestore, una base de datos NoSQL y un servicio de almacenamiento en la nube ofrecido por Firebase,

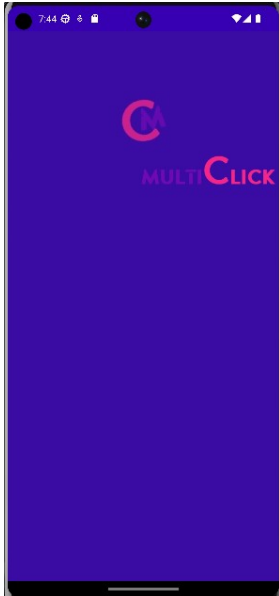
 multiclick-d51d1	 Articulos 	 0 
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Articulos >	0 >	+ Agregar campo
Carritos	1	Articulo: "Consola"
Categorias	2	
DireccionPedido		
InfodeArticulos		
Pedidos		
PedidosRealizados		
Stock		

En la siguiente tabla se detallan los campos existentes en cada colección:

Categorias	Articulos	InfodeArticulos	Carritos	DirecciónPedido	Stock
PS5	Consola	Artículo	Artículo	Email	Cantidad
PS4	Videojuego	Consola	Mail	NumPedido	Categoria
Xbox OneXS	Accesorio	Descripción		Apellidos	NombreArticulo
Xbox One		ImagenID		Código Postal	
Nintendo Switch		Precio		Direccion	
PC				Nombre	

FUNCIONAMIENTO:

1. Pantalla de Splash:



Se ha optado por dar especial importancia a la pantalla de Splash, puesto que es la primera toma de contacto del usuario con la aplicación. En ella podemos apreciar dos elementos animados por separado.

En primer lugar apreciamos el logo de la aplicación, que realiza una animación doble; por un lado rota 90° grados en sentido horario y por otro, al llegar a su posición, tiene un efecto rebote muy visual.

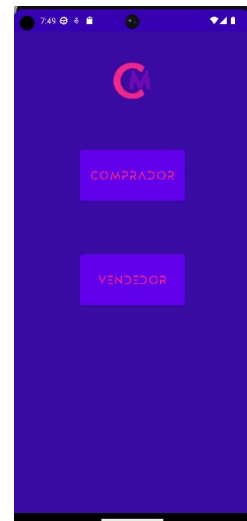
En segundo lugar, el nombre de la aplicación aparece desde la derecha.

Al finalizar esta pequeña animación, la aplicación de lugar a la pantalla en la que elige su rol de usuario.

2. Pantalla roles de usuario.

En la siguiente pantalla, el usuario debe elegir su rol; comprador o vendedor.

En un caso real se hubieran desarrollado dos aplicaciones diferentes, pero para el presente TFG se ha estimado que era más adecuado recoger ambos casos de uso en la misma aplicación.



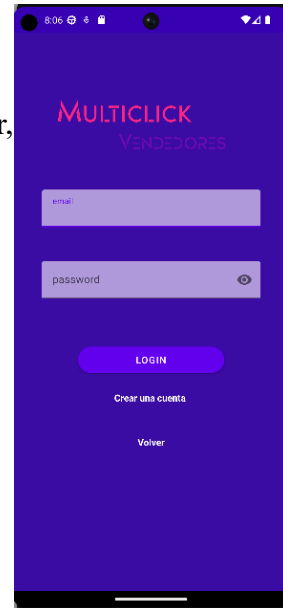
3. Pantalla de Login:

Cuando un usuario quiere utilizar la aplicación, debe crearse una cuenta. Si es un comprador, es necesaria para poder almacenar la información de sus pedidos, y si es un vendedor, es necesaria para poder enviarlos.

Desde esta pantalla se puede crear una cuenta o bien iniciar sesión.

La pantalla de Login consta de un email de usuario y una contraseña, sobre las cuales se han aplicado las siguientes restricciones:

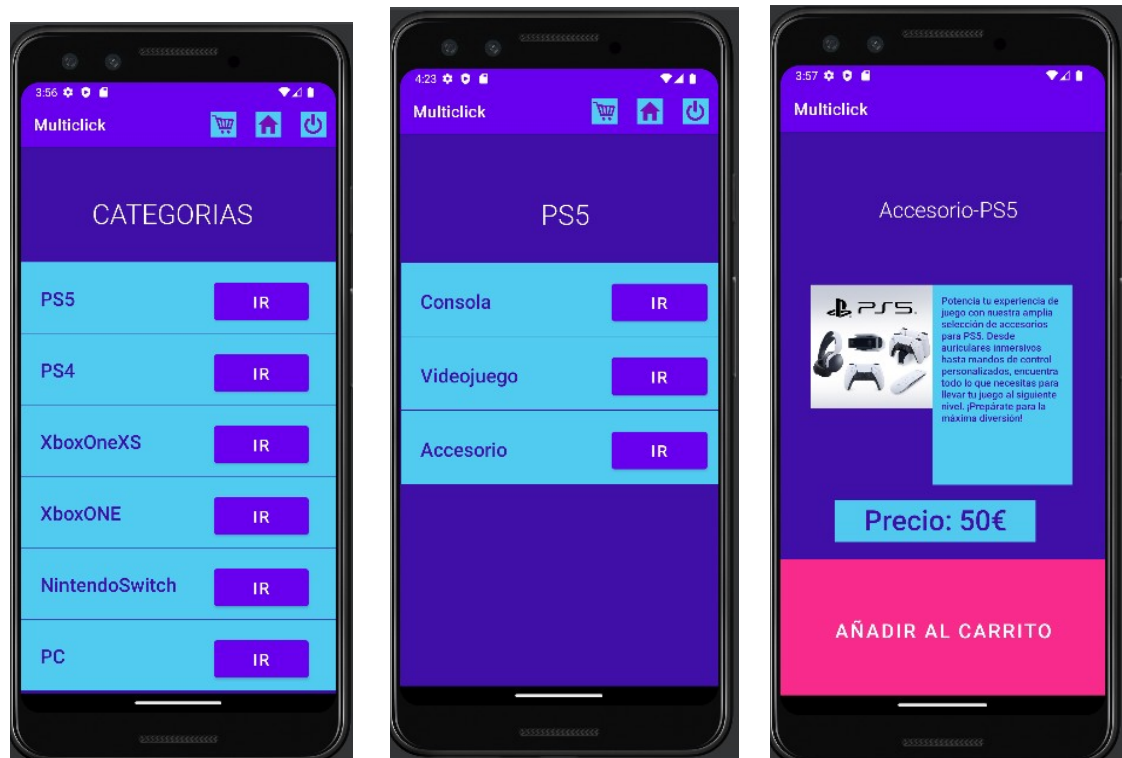
- Ambos campos deben estar completados para poder loguearse. De lo contrario, aparece un toast avisando del error.
- El email debe ser una dirección de correo válida.
- La contraseña debe contener al menos 6 caracteres
- Si uno de estas condiciones no se cumple, aparece un aviso sobre la propia caja de texto sugiriendo la manera correcta de hacerlo.
- Para utilizar la aplicación como vendedor, el usuario debe tener una dirección de correo de “@multiclick.com”.



Una vez que el usuario se ha creado una cuenta y ha iniciado sesión, puede comenzar el uso de la aplicación.

A continuación veremos el uso de la aplicación como usuario comprador y después como vendedor para ver cómo funciona todo el proceso.

4. Proceso de compra:



Nada más iniciar sesión como comprador, el usuario puede navegar entre las diferentes categorías de los productos de la tienda, y al hacer clic en cada una de ellas, se muestran las subcategorías.

Una vez hemos decidido qué producto queremos comprar, hacemos clic sobre él y la aplicación nos muestra una fotografía del producto, su descripción y el precio.

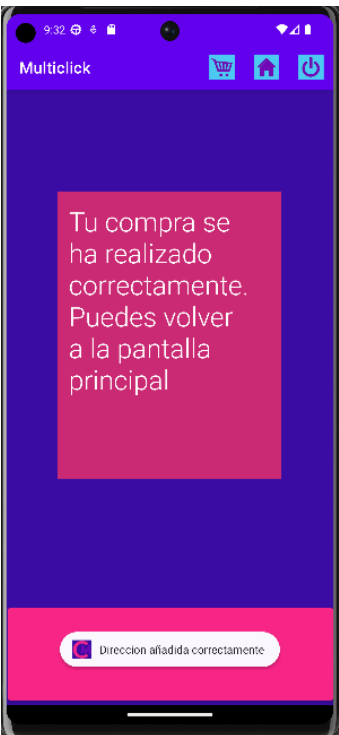
En este ejemplo, el usuario se ha dado un capricho por haber terminado con éxito su TFG y se ha comprado una PS5 y un videojuego para celebrarlo durante el fin de semana. También le ha comprado a su sobrino un videojuego para la Nintendo Switch por haber sacado buenas notas.

Para realizar el pedido, el usuario deberá rellenar de forma correcta su dirección para que el envío pueda realizarse correctamente.

Si alguno de los campos no está completo, la aplicación devuelve un mensaje de error para que el usuario rellene todos los campos.

Cuando el usuario ha introducido correctamente su dirección y pulsa el botón “Pagar”, la aplicación informa de que el pedido se ha realizado con éxito.

En este momento, el pedido se almacena en la colección “Pedidos” para que sea gestionado por un vendedor.

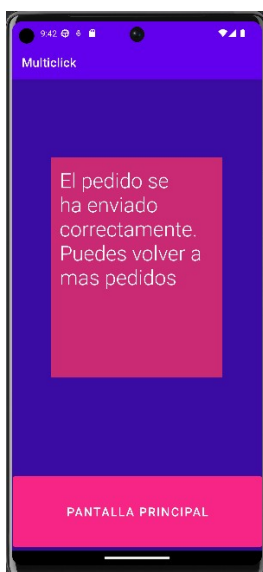


5. Gestión de pedidos:



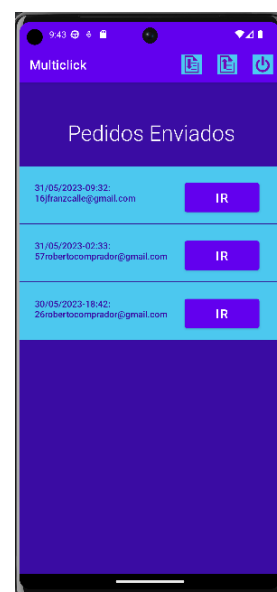
Cuando un vendedor abre la aplicación, se encuentra con la pantalla de pedidos pendientes de envío y su respectivo número de pedido.

Al hacer clic sobre cada uno de ellos, la aplicación nos muestra la información del pedido: El usuario que la ha comprado, su dirección y los productos que ha comprado.

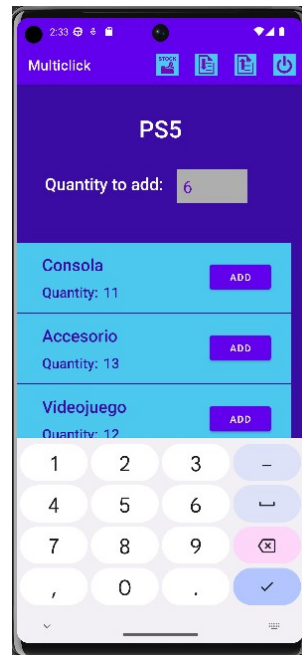
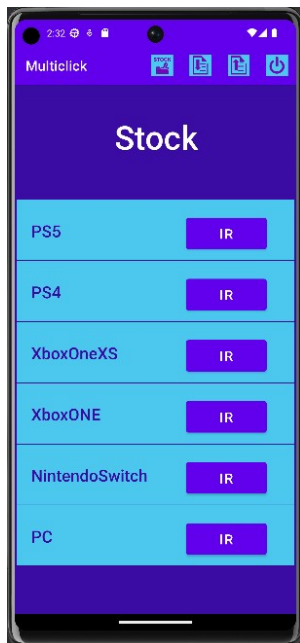


Si hacemos clic sobre el botón “Enviar pedido”, la aplicación nos informará de que se ha enviado correctamente.

También es posible visualizar los pedidos enviados y abrir los detalles de cada uno de ellos (Cliente, dirección y artículos)



6. Stock:



La pantalla de stock muestra primeramente las categorías. Al clicar sobre cada una de ellas se pasa a ver los diferentes artículos para esa categoría y la cantidad de stock disponible. También tiene la función de poder añadir más stock. Finalmente, si un pedido se envía, se restaran las cantidades de artículos correspondientes.

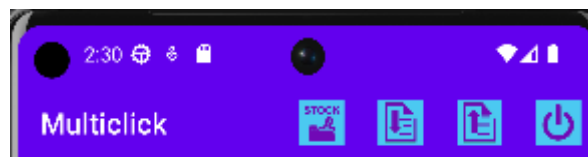
AppBar Comprador:

Está compuesta por tres iconos: El primero lleva al carrito, el segundo devuelve a la lista de categorías o “home” de la sección comprador y el tercero es el LogOut.



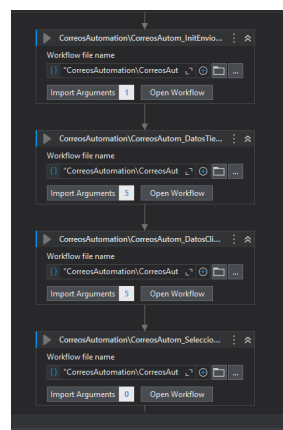
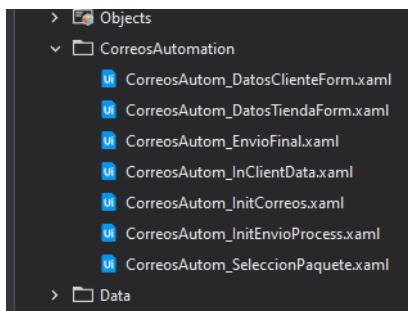
AppBar Vendedor:

Está compuesta por iconos: El primero lleva al control del stock, el segundo a los pedidos pendientes, el tercero a los pedidos enviados y el último es el LogOut



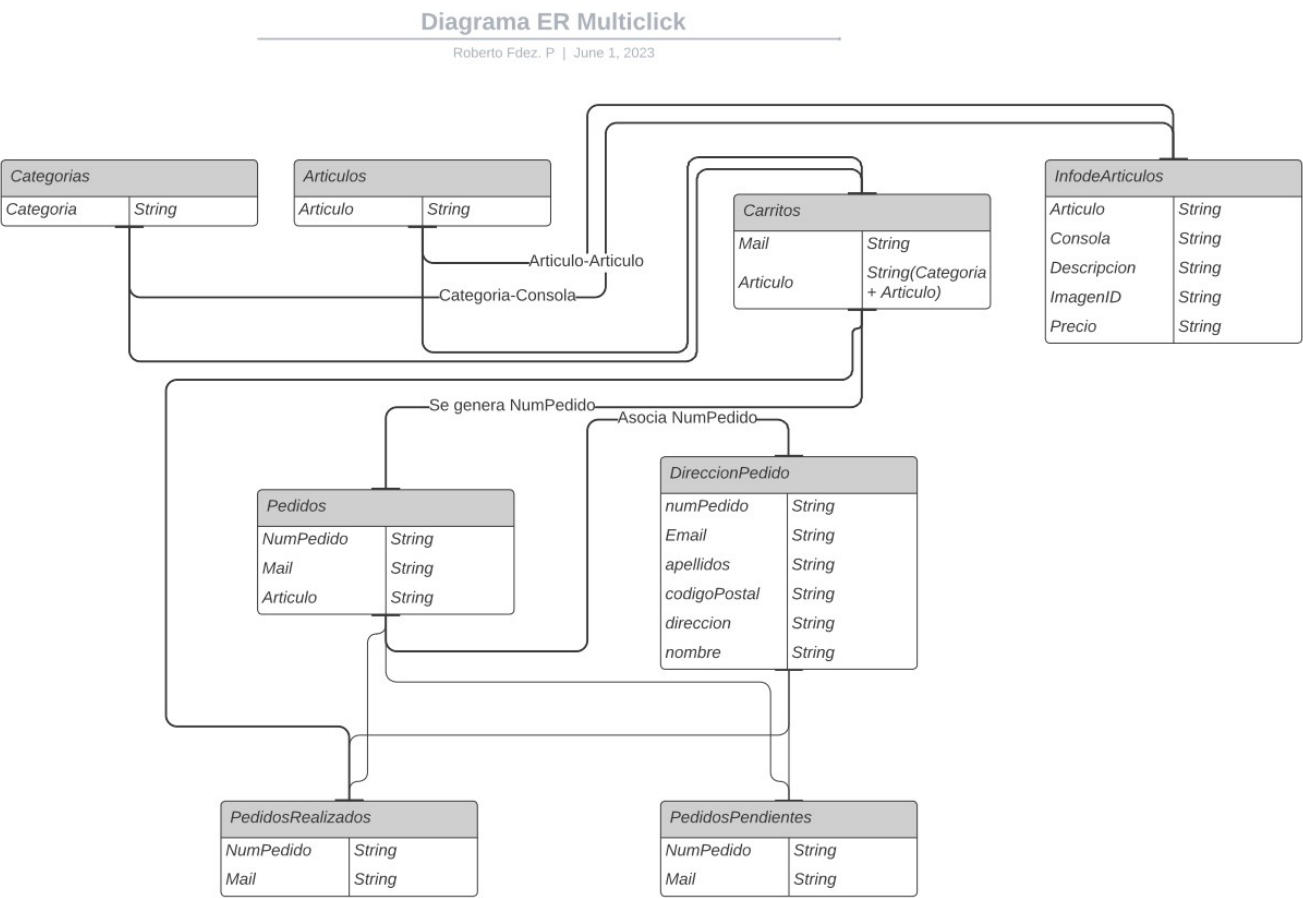
Automatización de formularios de envío de correos:

Se ha escogido las herramienta UiPath para abordar esta cuestión. Siendo una opción que se podría aplicar como solución en el mundo real. La automatización trata de pedir los datos necesarios a cerca del comprador y una vez introducidos los datos, la automatización rellena el formulario de forma completamente automática (como se puede ver en un video que adjuntamos). Los datos de la tienda (quien envía) ya están incluidos de forma predeterminada en el proyecto de UiPath. Al es un dato recurrente e invariable, no sera necesario introducirlo. En caso del comprador simplemente se piden los datos que aparecen en la sección de VendedorPedidoInfo.

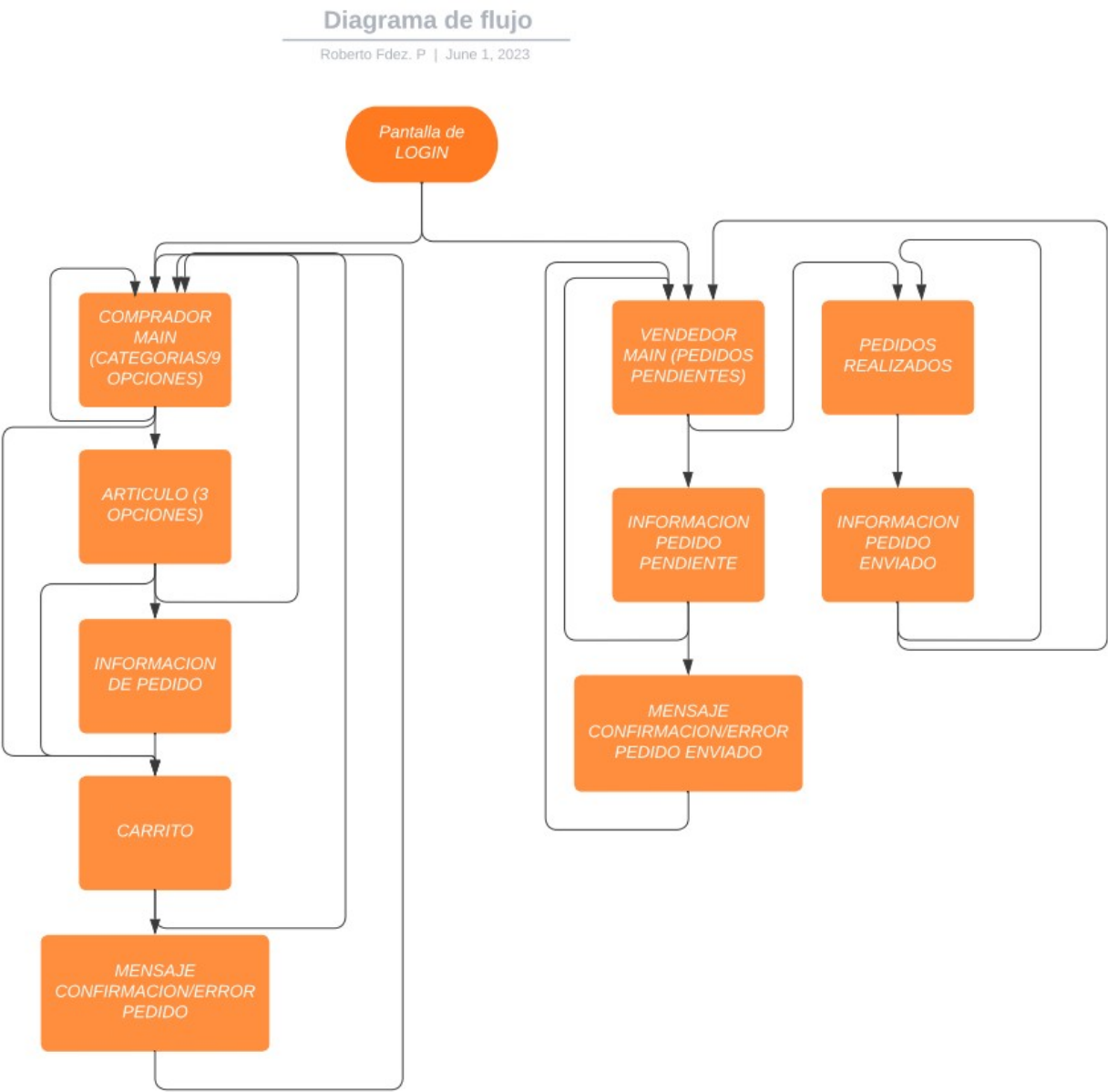


Relaciones lógicas de la BBDD (relacionándose a través del código)

La base de datos de Firebase es NoSQL (no relacional) pero se incluye un diagrama con las relaciones que se pretenden hacer desde la aplicación para el funcionamiento de la misma.



Flujo de datos



CONCLUSIONES Y MEJORAS DEL PROYECTO

Objetivos y resultados:

Durante el desarrollo de este proyecto, se lograron alcanzar los siguientes objetivos y resultados:

- Implementación de un sistema de autenticación: Se ha desarrollado un sistema de autenticación que permite a los usuarios iniciar sesión y registrarse en la aplicación utilizando sus correos electrónicos y contraseñas. Esto proporciona seguridad y privacidad a los usuarios.
- Interfaz de usuario intuitiva: Se ha diseñado una interfaz de usuario intuitiva y fácil de usar para mejorar la experiencia del usuario al interactuar con la aplicación. Se han utilizado elementos visuales y controles adecuados para facilitar la navegación y comprensión de las funcionalidades de la aplicación.
- Integración con Firebase: Se ha utilizado Firebase como plataforma backend para almacenar y gestionar datos relevantes de la aplicación, como la información de los usuarios y los productos. Esto permite un almacenamiento seguro y escalable de los datos, así como la implementación de funciones en tiempo real.
- Gestión efectiva del proceso de compra: Se ha logrado implementar un proceso de compra eficiente, donde los compradores pueden agregar productos al carrito, realizar pagos de forma segura y recibir confirmaciones de compra. Esto asegura una experiencia de compra fluida y satisfactoria para los usuarios.
- Gestión efectiva de los pedidos por parte del vendedor: Se ha implementado una funcionalidad que permite a los vendedores administrar sus pedidos de manera efectiva. Pueden ver y actualizar el estado de los pedidos, así como enviar información de envío a los compradores. Esto simplifica y agiliza el proceso de gestión de pedidos para los vendedores.
- Aportación de una solución cómoda para agilizar la cumplimentación de documentación de los pedidos para ser enviados al cliente a través de la empresa “Correos”.

Mejoras potenciales:

Aunque el proyecto ha alcanzado los objetivos establecidos, existen áreas de mejora que podrían considerarse en futuras versiones de la aplicación:

- Ampliación de funcionalidades: Se pueden agregar nuevas funcionalidades para mejorar la experiencia del usuario, como la implementación del sistema de envío mediante Correos para facilitar la entrega de productos, la integración de un sistema de pago con PayPal para ofrecer más opciones de pago a los usuarios y la implementación de una funcionalidad que permita a los vendedores agregar nuevos productos de manera sencilla.
- Mejoras en la interfaz de usuario: Se pueden realizar ajustes y mejoras en el diseño y la usabilidad de la interfaz de usuario para hacerla más atractiva y fácil de usar. Esto puede incluir la optimización de la disposición de elementos, el uso de animaciones sutiles y la mejora de la coherencia visual en toda la aplicación.

Estas mejoras potenciales pueden ayudar a enriquecer la funcionalidad y usabilidad de la aplicación, así como a ofrecer una experiencia mejorada para los usuarios.