

EasyBuy



Javier J. Cordero Álvarez



Índice

1	Explicación de la Aplicación	3
1.1	Descripción General	3
1.2	Funcionalidades Clave.....	3
1.3	Tecnologías y Recursos Utilizados.....	4
2	Capturas de Pantalla.....	5
2.1	Pantallas de la Aplicación	5
2.2	Detalles Importantes	9
3	Conclusiones y Resultados.....	15
3.1	Logros y Desafíos	15
3.2	Lecciones Aprendidas	16
3.3	Posibles Mejoras	17
4	Conclusión.....	17
5	Referencias.....	17
6	Repositorio de la aplicación.....	17



1 Explicación de la Aplicación

1.1 Descripción General

EasyBuy representa una aplicación diseñada para simplificar y agilizar la gestión de compras y productos. Desarrollada sobre el framework de Flutter, esta plataforma tiene como objetivo principal ofrecer a los usuarios una experiencia intuitiva y eficiente en la gestión de listas de compra, inventarios y selección de productos.

Concebida como una herramienta versátil, EasyBuy busca proporcionar una interfaz accesible y amigable, permitiendo a los usuarios crear listas de compras personalizadas, gestionar productos, y realizar operaciones asociadas a una base de datos en tiempo real de Firebase. Esta aplicación no solo enfatiza la simplicidad en la gestión de información, sino que también garantiza la seguridad y confiabilidad en las operaciones.

La combinación de tecnología avanzada con una interfaz cuidadosamente diseñada hace que EasyBuy sea más que una aplicación para la gestión de compras: es un asistente personalizado, adaptado para simplificar la experiencia de compra y gestión de inventarios.

1.2 Funcionalidades Clave

Las funcionalidades clave de la aplicación EasyBuy son:

Gestión de Listas de Compra: EasyBuy permite a los usuarios crear, editar y organizar listas de compra personalizadas. Los usuarios pueden agregar, eliminar o modificar elementos de la lista según sus necesidades, facilitando la planificación y seguimiento de compras.

Inventario de Productos: La aplicación ofrece un espacio dedicado para administrar el inventario de productos. Los usuarios pueden registrar y categorizar productos, incluyendo detalles como nombre, categoría, precio y cantidad disponible.

Base de Datos en Tiempo Real: Utilizando Firebase Realtime Database, EasyBuy sincroniza los datos de manera instantánea, permitiendo a múltiples usuarios acceder y actualizar información en tiempo real. Esto garantiza la coherencia de la información en diferentes dispositivos.

Interfaz Intuitiva y Navegación Sencilla: EasyBuy se enfoca en una interfaz amigable que permite una navegación intuitiva. Los usuarios pueden acceder rápidamente a las funciones principales, facilitando la gestión de compras y productos sin complicaciones.



Funciones de Edición y Eliminación: Además de añadir y visualizar productos, la aplicación ofrece capacidades de edición y eliminación. Los usuarios pueden modificar detalles como precios, cantidades y categorías, así como eliminar productos del inventario.

Seguridad y Confidencialidad: EasyBuy prioriza la seguridad de los datos de los usuarios. Implementa medidas para asegurar la confidencialidad de la información almacenada y la privacidad de las transacciones.

Estas funcionalidades clave representan las principales características que hacen de EasyBuy una herramienta integral para la gestión de compras y productos.

1.3 Tecnologías y Recursos Utilizados

Las principales tecnologías y recursos usados para crear el aplicativo son las siguientes:

Flutter: EasyBuy se desarrolló utilizando el framework Flutter de Google. Flutter ofrece una base para el desarrollo de aplicaciones multiplataforma con un solo código base, permitiendo una experiencia de usuario consistente en dispositivos Android y iOS.

Firestore: La aplicación aprovecha Firestore como plataforma integral para el backend. Firestore Realtime Database permite una sincronización instantánea de datos entre dispositivos, facilitando la gestión en tiempo real de listas de compras y productos.

Firestore Authentication: Para garantizar la seguridad de los usuarios, EasyBuy emplea Firestore Authentication para la autenticación de usuarios. Proporciona un sistema seguro de inicio de sesión y registro, protegiendo la información de manera confiable.

Firestore Database: La base de datos en tiempo real de Firestore es fundamental en EasyBuy. Permite almacenar y recuperar datos de manera eficiente, manteniendo la coherencia y actualización instantánea de la información en la aplicación.

Dart Programming Language: Dart es el lenguaje de programación principal utilizado en Flutter. Su estructura ágil y capacidad para desarrollar aplicaciones receptivas y eficientes contribuyen a la fluidez y rendimiento de EasyBuy.

Material Design: EasyBuy sigue las directrices de Material Design para ofrecer una interfaz de usuario intuitiva y atractiva. Se enfoca en la consistencia visual y la usabilidad para proporcionar una experiencia agradable al usuario.



2 Capturas de Pantalla

2.1 Pantallas de la Aplicación

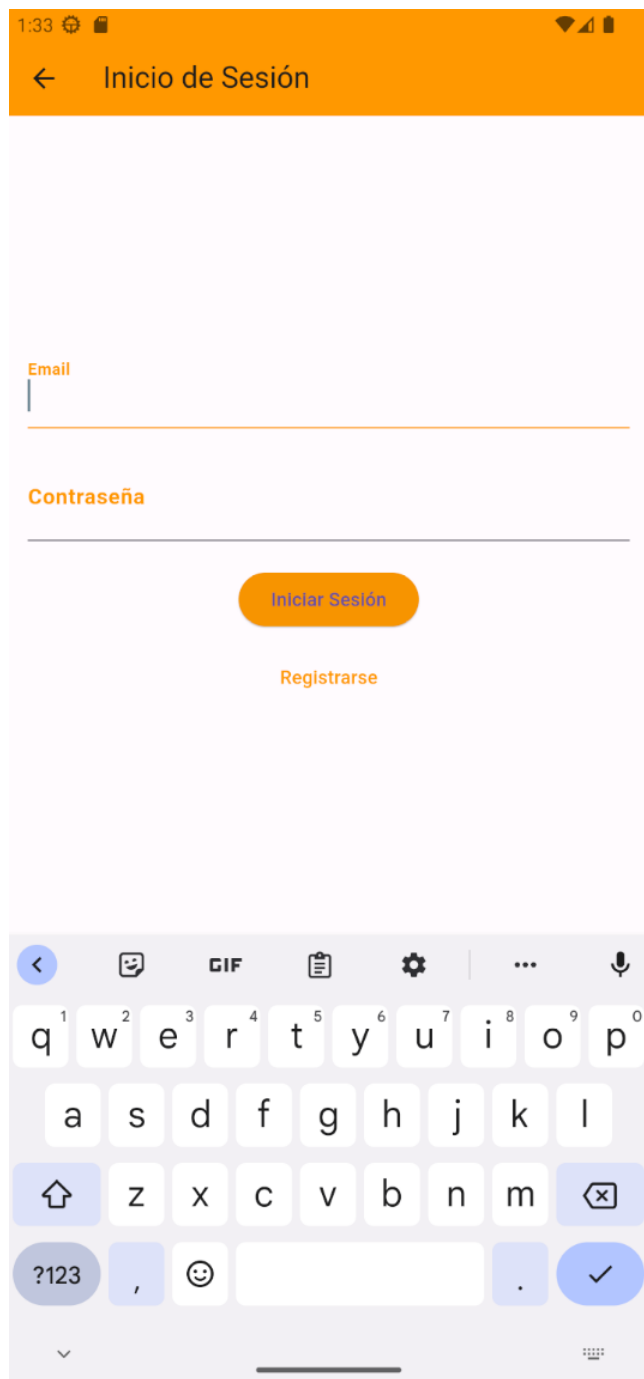
Pantalla Principal





Botón Login



Pantalla Login



1:33  

← Inicio de Sesión

Email

Contraseña

Iniciar Sesión

Registrarse

q¹ w² e³ r⁴ t⁵ y⁶ u⁷ i⁸ o⁹ p⁰

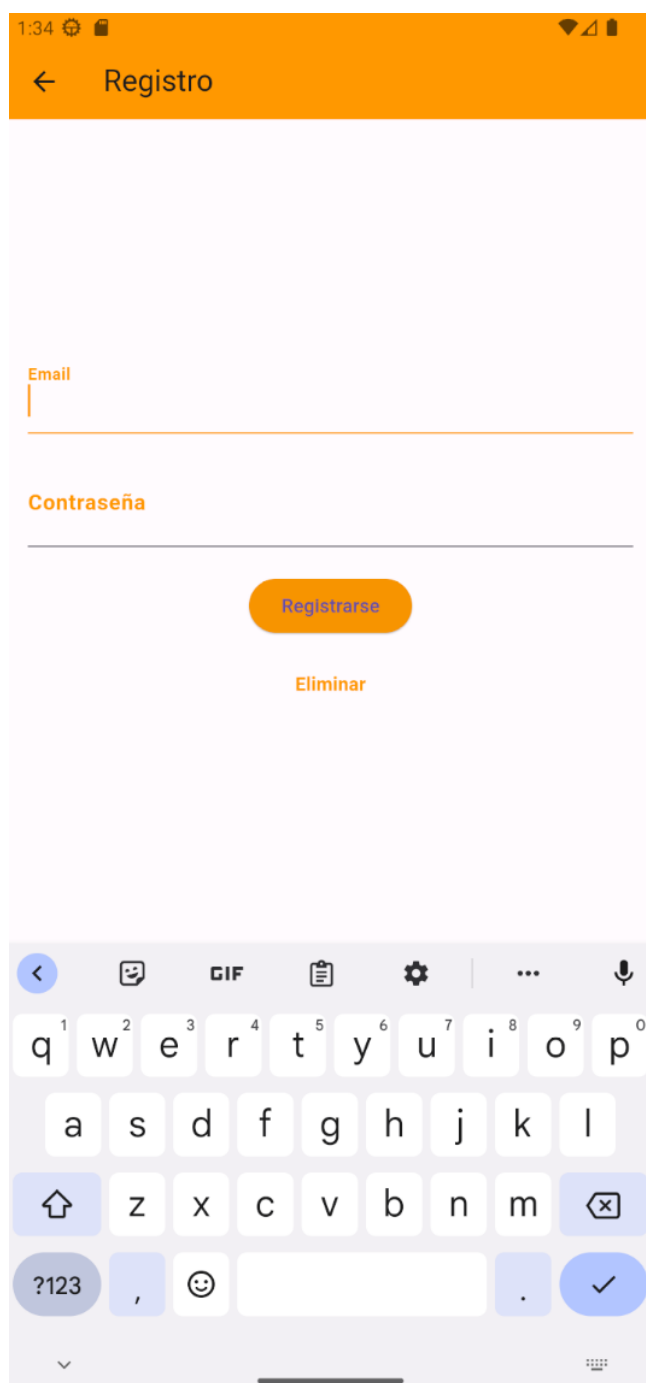
a s d f g h j k l

⬆ z x c v b n m ⬇

?123 , 😊 . ✓



Pantalla Registro



The screenshot shows a mobile application interface for a registration screen. At the top, there is an orange header bar with a back arrow icon and the text "Registro". Below the header, the main area has a light pink background. It contains two input fields: "Email" and "Contraseña" (Password). The "Email" field has a small orange vertical line on its left side. Below the "Contraseña" field, there is an orange button labeled "Registrarse" and a link labeled "Eliminar" in orange text. At the bottom of the screen, a standard Android keyboard is visible, showing keys for letters, numbers, and symbols.



Pantalla Privada

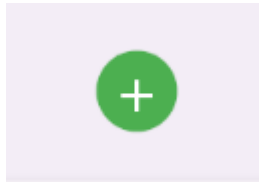


Botones para cambiar de pantalla

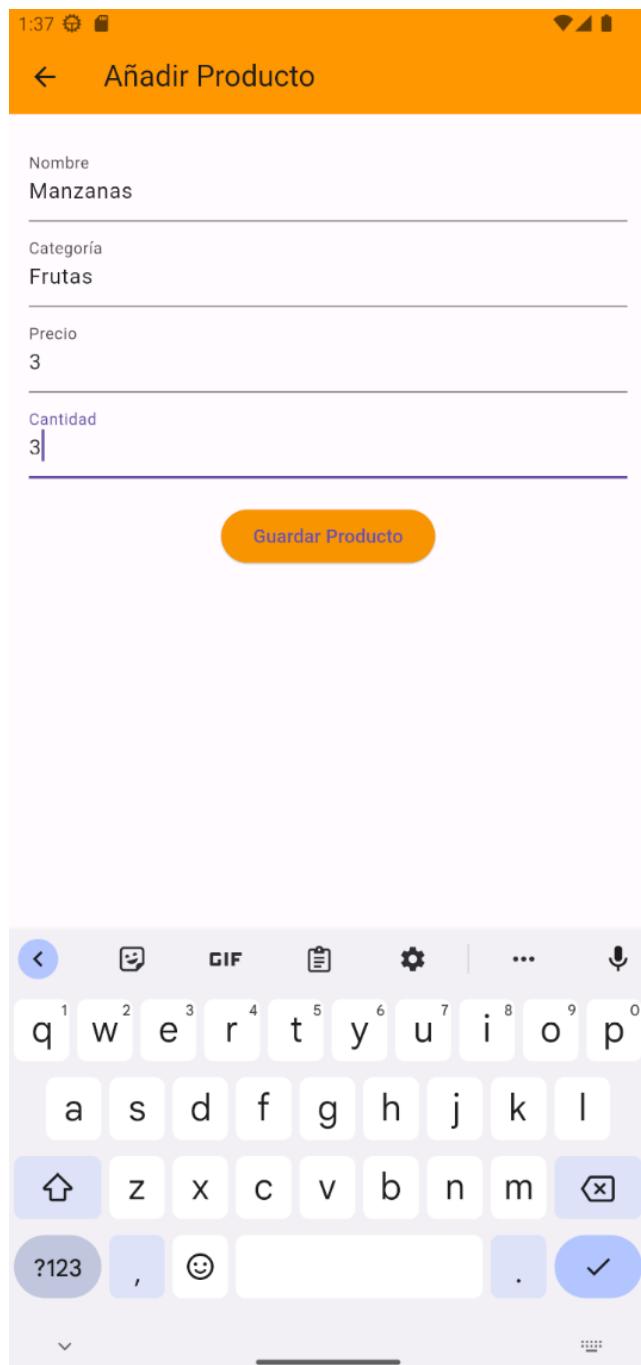


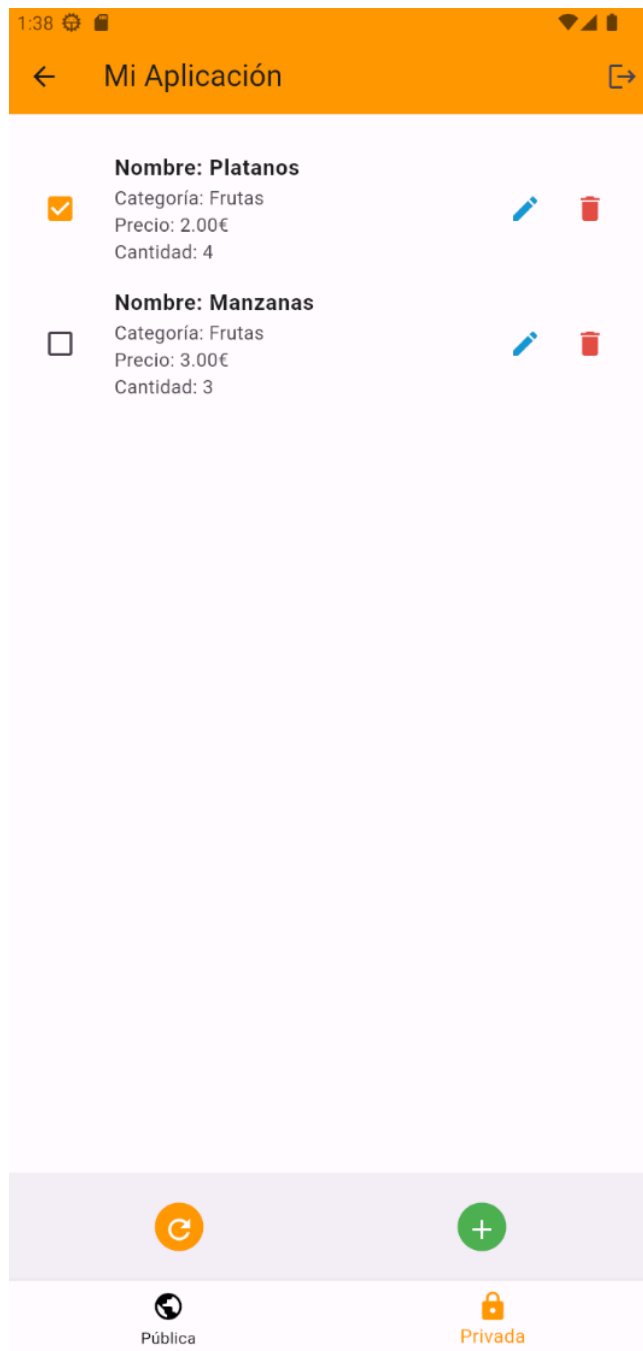
2.2 Detalles Importantes

Botón añadir producto



Añadir producto

Una captura de pantalla de una aplicación móvil. La barra superior es naranja con un botón de retroceso y el título 'Añadir Producto'. El formulario tiene campos para 'Nombre' (Manzanas), 'Categoría' (Frutas), 'Precio' (3) y 'Cantidad' (3). Debajo del formulario hay un botón naranja 'Guardar Producto'. En la parte inferior se ve un teclado virtual con un botón de confirmación azul.



Botón editar producto



Editorial product

←

Editar Producto

Nombre

Manzanassss

Categoría

Frutassss

Precio

4

Cantidad

5

Guardar Cambios

1

2

3

-

4

5

6

_

7

8

9

⌫

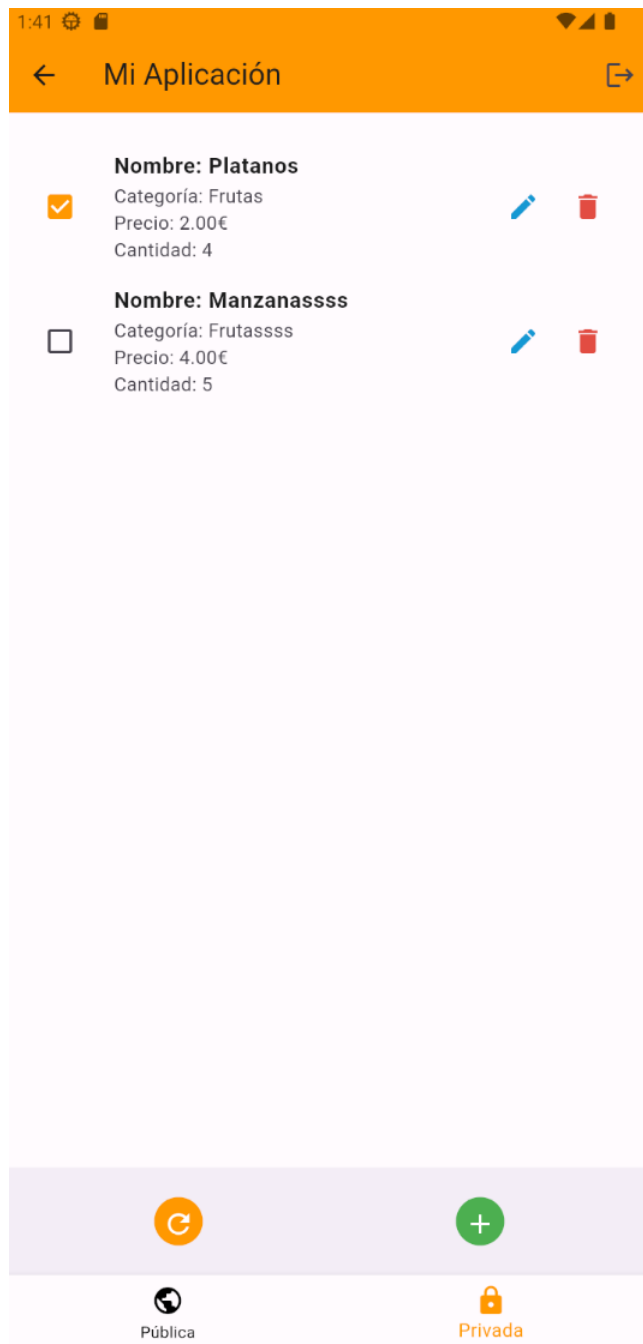
,

0

.

✓

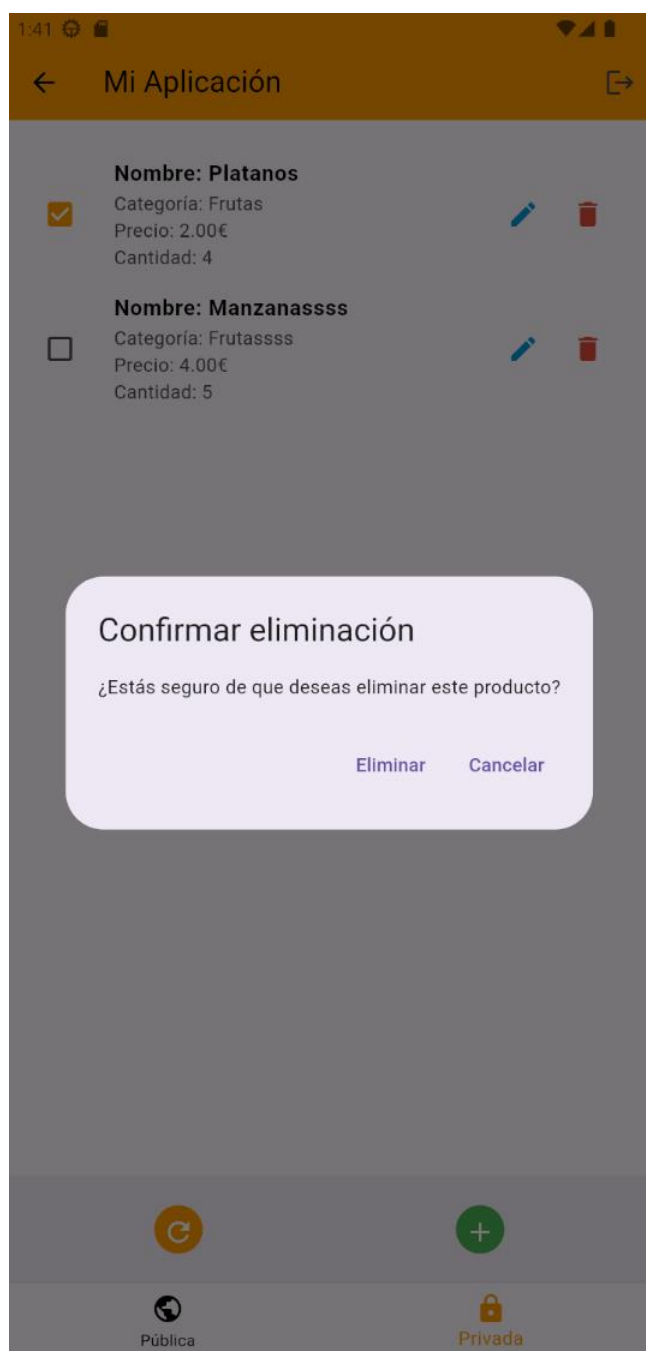




Botón eliminar producto



Eliminar producto

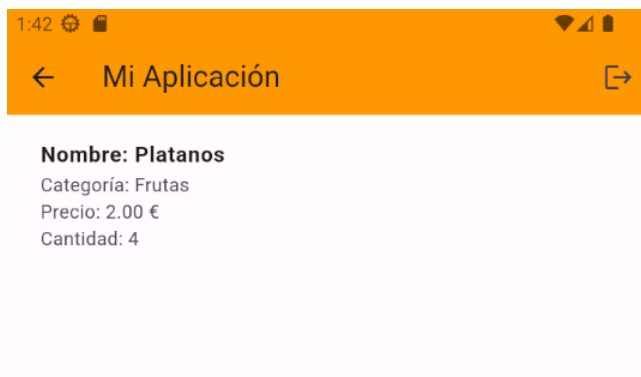




Checkbox para seleccionar producto



Producto Seleccionado mostrado en pantalla principal



Botón logout



3 Conclusiones y Resultados

3.1 Logros y Desafíos

Logros Alcanzados

1. **Experiencia de Usuario Consistente:** EasyBuy logra ofrecer una experiencia de usuario uniforme en plataformas iOS y Android gracias a Flutter. La interfaz intuitiva y la navegación fluida fueron aspectos destacados en el desarrollo.
2. **Gestión en Tiempo Real:** El uso de Firebase Database permitió implementar una gestión en tiempo real de las listas de compras y productos. Los cambios se reflejan instantáneamente en todos los dispositivos conectados, proporcionando una experiencia de usuario ágil y colaborativa.
3. **Seguridad Robusta:** La integración de Firebase Authentication aseguró un sistema de inicio de sesión y registro seguro para los usuarios, protegiendo la información personal y garantizando la confidencialidad de los datos.

Desafíos Superados

1. **Sincronización en Tiempo Real:** Garantizar la sincronización instantánea de datos en tiempo real a través de Firebase Database fue un desafío clave. La gestión eficiente de cambios y actualizaciones en la base de datos para reflejarlos instantáneamente en la aplicación requirió un enfoque cuidadoso.
2. **Compatibilidad Multiplataforma:** Asegurar una experiencia uniforme en dispositivos Android e iOS fue un desafío técnico. Se abordó mediante



pruebas exhaustivas y optimizaciones para garantizar el rendimiento y la apariencia consistentes en ambas plataformas.

3. **Manejo de la Seguridad:** Implementar un sistema de autenticación seguro y robusto con Firebase Authentication requirió una cuidadosa configuración y validación para garantizar la protección de la información del usuario.
4. **Navegación Compleja:** Se resolvieron desafíos en la navegación entre pantallas y en la gestión del estado de la aplicación al realizar el inicio y cierre de sesión.
5. **Depuración de Errores:** Se abordaron diversos problemas, desde errores en la autenticación hasta la gestión de errores al eliminar productos.
6. **Gestión de Firebase:** La configuración e instalación de Firebase Database, Core y Auth presentaron el desafío más significativo. Requirió la descarga de un programa especializado de Firebase con una terminal dedicada para ejecutar comandos, ya que los scripts no funcionaban desde la terminal estándar de VS Code o Windows.

Estos logros y desafíos superados fueron fundamentales en el desarrollo de EasyBuy, contribuyendo a su calidad, seguridad y funcionalidad.

3.2 Lecciones Aprendidas

Configuración Detallada: Entender la configuración detallada de servicios como Firebase resultó crucial. Aprendí que ciertas plataformas requieren pasos específicos y herramientas dedicadas para una integración exitosa.

Resolución de Problemas: Enfrentar obstáculos técnicos reforzó la importancia de la resolución de problemas. Aprendí a utilizar recursos alternativos y adaptar enfoques para solucionar problemas técnicos complejos.

Pruebas Locales: Experimentar con distintas formas de realizar pruebas locales me enseñó la relevancia de probar la funcionalidad en un entorno controlado antes de la implementación.

Documentación y Comunidad: Valorar la documentación detallada y la participación activa en comunidades técnicas para resolver problemas y encontrar soluciones eficientes.

Persistencia y Paciencia: La persistencia y la paciencia son clave en el desarrollo, sobre todo al enfrentar desafíos técnicos que pueden requerir múltiples intentos y enfoques diferentes para encontrar una solución.



3.3 Posibles Mejoras

Gestión de Usuarios: Permitir a cada usuario registrado acceder y gestionar sus propios datos de manera individual. Esto implica separar la base de datos por usuarios, garantizando que cada uno tenga un espacio dedicado para sus registros personales.

Eliminación de Usuarios: Facilitar a los administradores o a los propios usuarios registrados la capacidad de eliminar cuentas desde la lista de usuarios. Esta función garantiza una gestión eficiente y segura de la base de datos, permitiendo la eliminación de cuentas inactivas o no deseadas.

Funcionalidades Avanzadas: Considerar la adición de funcionalidades avanzadas, como filtros de búsqueda más elaborados, recomendaciones personalizadas, integración de métodos de pago adicionales, entre otros.

4 Conclusión

EasyBuy representa una solución integral para la gestión eficiente de inventarios y compras, ofreciendo a los usuarios una plataforma intuitiva y versátil. Esta aplicación simplifica el control de productos, brindando funcionalidades como la adición, edición y eliminación de artículos, así como la visualización detallada de datos relevantes.

El desarrollo de EasyBuy ha sido un proceso educativo y productivo. Ha proporcionado una experiencia valiosa en el ámbito de la gestión de bases de datos y el diseño de interfaces amigables. Este proyecto ha demostrado la importancia de la planificación detallada, la implementación eficiente de tecnologías y la capacidad de adaptación para superar desafíos técnicos. Además, ha resaltado la necesidad de la escalabilidad y la mejora continua para ofrecer una experiencia de usuario óptima.

5 Referencias

<https://pub.dev/>

<https://firebase.google.com/docs/database?hl=es-419>

<https://stackoverflow.com/>

<https://chat.openai.com/>

<https://docs.flutter.dev/>

6 Repositorio de la aplicación

<https://github.com/javi-dev-79/Shopping-List-Flutter-Firebase>

