**MEMORIA DEL PROYECTO INTEGRADO**

**“LUNA FASHION”**

**Realizado por**

DANIEL ARANDA MAESTRO

**Tutor**

JOSÉ ANTONIO LAR SÁNCHEZ

**Grado Superior “Desarrollo de aplicaciones multiplataforma”**

**Cesur (Málaga Este)**

Índice

[Resumen del proyecto 5](#_Toc102320664)

[Introducción 7](#_Toc102320665)

[Objetivos 9](#_Toc102320666)

[Características del proyecto 9](#_Toc102320667)

[1. Framework y lenguaje de programación 9](#_Toc102320668)

[2. Funcionalidades principales 10](#_Toc102320669)

[3. Infraestructura y servicios 10](#_Toc102320670)

[4. Tipo de arquitectura 12](#_Toc102320671)

[5. Packages externos 13](#_Toc102320672)

[Finalidad 15](#_Toc102320673)

[Medios materiales usados 17](#_Toc102320674)

[Humanos 17](#_Toc102320675)

[Hardware 18](#_Toc102320676)

[Software 19](#_Toc102320677)

[1. Flutter 19](#_Toc102320678)

[2. Visual Studio Code 20](#_Toc102320679)

[3. Android Studio / Xcode 21](#_Toc102320680)

[4. Firebase 22](#_Toc102320681)

[5. GitHub 23](#_Toc102320683)

[Planificación del proyecto 25](#_Toc102320684)

[Fase de pruebas 26](#_Toc102320685)

[Conclusiones y trabajos futuros o posibles mejoras 26](#_Toc102320686)

[Apéndices 27](#_Toc102320687)

[Referencias bibliográficas 28](#_Toc102320688)

# Resumen del proyecto

Tras la pandemia, una tienda de barrio ha sufrido mucho y sus ventas han caído considerablemente, necesitan un cambio urgente para poder aumentar sus ventas y continuar con el negocio. Se pretende realizar una aplicación en el que los clientes puedan realizar sus pedidos a través de la aplicación y poder gestionar dichos pedidos por los usuarios administradores. De esta forma, los usuarios sólo tendrán que ir a la tienda cuando sus pedidos estén listos para recoger.

Como la aplicación se necesita que esté disponible para todo el mundo, es necesario realizar una aplicación en Android y otra en iOS para poder subirlo a las plataformas de Google Play y App Store. Para ello vamos a realizar la aplicación en Flutter, que nos permitirá exportar la aplicación en Android e iOS sin ningún tipo de problemas desde un mismo código para ambos.

Además, la aplicación necesitará de un backend donde se pueda gestionar las autenticaciones de los usuarios, una base de datos para almacenar la información de los productos, pedidos y usuarios, y un sistema de mensajería para las notificaciones push. Por ello, como solución, hemos optado por la plataforma Firebase que nos provee de todos estos y otros servicios de forma sencilla y rápida.

# Introducción

Este proyecto consiste en una aplicación para una tienda de barrio, enfocado para el comercio menor.

Las funcionalidades principales de la aplicación serán las de seleccionar productos escogidos por el usuarios, agregarlos al carrito y realizar posteriormente la compra de dichos productos. Además, la aplicación distinguirá de uno o varios usuarios con el rol de Administrador para poder visualizar y gestionar los pedidos realizados a través de la aplicación.

La intención es que los usuarios compren los productos a través de la aplicación, estos pedidos pasan a un estado pendiente, donde el administrador preparará los pedidos y cuando estén listos cambia el estado del pedido a completado para que el usuario final pueda ir a recoger su compra en la tienda física.

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.

# Objetivos

El objetivo principal de la aplicación es poder aumentar las ventas a través de la aplicación y facilitar los estados de los pedidos entre el administrador y el usuario final.

## Características del proyecto

Las características del proyecto podemos dividirlas en Framework a desarrollar, funcionalidades principales de la aplicación, infraestructura y servicios, el tipo de arquitectura y packages externos.

Cada uno de estos puntos es muy importante para conocer bien el proyecto y así entender todas las partes que lo componen.

### Framework y lenguaje de programación

* Flutter: Es el kit de herramientas de *UI de Google* para realizar hermosas aplicaciones, compiladas nativamente, para móvil, web y escritorio desde una única base de código.
* Dart: Es un lenguaje de programación que usa *Flutter* y es *open source*, creado por Google en 2011. Se puede utilizar para aplicaciones web, servidores, aplicaciones de escritorio y aplicaciones móviles.

### Funcionalidades principales

* Autenticación: El proceso de autenticación consiste en un ingreso sencillo con *email* y *contraseña*. Este ingreso se auto conectará con la plataforma *Firebase* donde nos proporcionará las credenciales y el token del usuario para determinar si está o no autenticado. Además cuenta con una página de registro con un formulario básico.
* Onboarding: Es un pequeño proceso por el que se ayudará al usuario a entender cómo funciona y para qué sirve la aplicación.
* Productos: Una de las funcionalidades principales de la aplicación en el que podremos obtener un listado de los productos y sus detalles.
* Carrito: Una de las funcionalidades principales de la aplicación en el que gestionará los productos que tiene almacenado el usuario en su carrito.
* Pedidos: Una de las funcionalidades principales de la aplicación en el que se obtendrá un listado de los pedidos tanto para el administrador que lo pueda gestionar como para el usuario para que sepa el estado actual de su pedido.
* Notificaciones Push: Funcionalidad extra para el administrador, con las notificaciones push podrá ser avisado de cuando se ha realizado un pedido, de quien lo ha hecho y de cuantos productos constará dicho pedido.
* Internalización (i18n): Funcionalidad extra que permite manejar varios idiomas dentro de la aplicación.

### Infraestructura y servicios

* Firebase: Es una plataforma diseñada y creada por Google para desarrollar y facilitar la creación de aplicaciones, obteniendo así aplicaciones de manera rápida y de gran calidad. Además cuenta con muchísimos servicios. Esta plataforma tiene una pequeña capa gratuita y en cuanto se sobrepasa comienza a ser de pago.
* Firebase core: Permite la conexión y el manejo directo con la plataforma Firebase.
* Firebase auth: Servicio de autenticación, permite poder autenticarse con numerosos métodos como: por Email, por Google, por Facebook, por Teléfono, por Apple, por Anónimo, por Twitter, etc…
* Firebase storage: Servicio de almacenamiento, permite almacenar cualquier tipo de archivo, obtener una url y la posibilidad de poder descargar dichos archivos.
* Cloud firestore: Servicio de base de datos, esta base de datos es una base de datos documental muy similar a las No-SQL, permite almacenar la información a través de colecciones y documentos. También tiene la característica de agregar reglas de seguridad para que sólo pueda tener acceso tanto de escritura y/o de lectura de los usuarios que se deseen. Además cuenta con la posibilidad de crear índices de búsquedas para mayor rapidez.
* Firebase crashlytics: Servicio de fallos, es un servicio muy útil para recopilar y detectar errores que se producen en la aplicación, de esta manera podremos corregir estos fallos sin necesidad de que sea el usuario que nos reporte dichos errores.
* Firebase analytics: Servicio de métricas, nos permite a través de eventos conocer más al usuario final para poder estudiar así mejor los comportamientos que ocurren dentro de la aplicación.
* Firebase messaging: Servicio para poder utilizar las notificaciones push de los dispositivos móviles, se puede usar para enviar mensajes masivos a grupos específicos o mensajes específicos a un usuario en particular. Se maneja a través de *topics* o *token del usuario*.
* Firebase remote config: Es un poderoso servicio que permite almacenar variables de entornos y poder obtenerlo en la aplicación para no tener que almacenar datos sensibles dentro de la aplicación.

### Tipo de arquitectura

* Clean Arquitecture: Es un tipo de arquitectura en el que consiste separar los elementos principales de un proyecto en diferentes capas en forma de anillos. De fuera a dentro podemos encontrar la capa de Infraestructura, donde encontramos las conexiones con todo lo exterior como bases de datos, servicios, interfaces externas, webs, etc. Una capa más a dentro encontramos la de Aplicación donde se definen todas las reglas de negocio y más a dentro la capa de Dominio donde se definen los modelos, las entidades y otras reglas de negocio entre otro. Al separar por capas conseguimos que nada esté completamente acoplado en especial con la capa de Presentación, si hubiera que cambiar algo relativo a la infraestructura como la implementación de una nueva base de datos no se verían afectadas todas las demás capas.
* Principios Solid: Son unas pequeñas reglas que ayudarán a desarrollar software de calidad. Cada letra tiene una característica especial, la primera S de single responsibility principle, trata de convertir algo complejo en muchas cosas simples y definir así métodos que sólo hagan lo que menciona el nombre del método. El segundo principio es la O de open/closed, abierto para escalar el proyecto y cerrado para modificaciones de cambios. De esta forma hacemos más robusto y estable la aplicación. El tercer principio la L de Liskov substitution, consiste en que las clases derivadas deben poder sustituirse por sus clases base. El cuarto principio la I de Interface segregation, consiste en definir cada interfaz por caso de uso de esta forma queda mucho más independiente cada uso que se le vaya a dar a la aplicación. Por último queda la D de Dependency inversión, recomienda que los módulos de alto nivel no deberían depender de los módulos de bajo nivel, se refiere a que hay que crear clases abstractas para que sus implementaciones sean totalmente independientes.

### Packages externos

* Flutter\_navite\_splash: Permite crear un splash personalizado en nativo de forma sencilla y rápida. Gracias a esto se elimina el splash blanco que viene por defecto en Flutter.
* Flutter\_launcher\_icons: Permite crear el icono de la aplicación de forma sencilla y rápida para Android e iOS.
* Flutter\_riverpod: Es uno de los manejadores de estados más potentes que hay junto con Flutter Bloc y Provider.
* Dio: Es un poderoso cliente Http con el que podemos crear decoradores y configuraciones de forma sencilla.
* Logger: Permite configurar los logs de tu proyecto.
* Another\_flushbar: Instanciar toasts personalizados ya creados.
* Lottie: Renderiza efectos de animaciones con archivos específicos de Lottie.
* Flutter\_svg: Permite visualizar archivos en svg.
* Image\_picker: Permite capturar y seleccionar imágenes de tu dispositivo.
* Url\_launcher: Permite abrir un navegador.
* Flutter\_slidable: Widget personalizado listo para usarse, permite deslizar ítems para agregar funciones extras.

# Finalidad

La finalidad de esta aplicación es la de poder ayudar a pequeños comercios con dificultades para que puedan de una manera fácil y económica aumentar sus ventas. Con esto se consigue más participación en el uso de las aplicaciones móviles. Además que facilitamos a los usuarios finales el realizar pedidos sin la necesidad de ir a la tienda física.

Como se puede observar, todo es una cadena, que va desde el propio comercio hasta los usuarios finales. Con esto, indirectamente ganan todos.

# Medios materiales usados

El equipamiento técnico es uno de los puntos más importantes para un desarrollador, ya que es con lo que trabaja en el día a día. Tener un buen espacio de trabajo ayuda a enfocarse mejor y a ser más productivo.

Además, es necesario que cuente con las mejores herramientas y mantenerse actualizado con las últimas tecnologías, ya que es muy fácil quedarse desactualizado.

Para este proyecto, se definirán todos los medios que se va a utilizar, tanto humanos, hardware y software.

## Humanos



Daniel Aranda Maestro  
Desarrollador de aplicaciones móviles con 4 años de experiencia.  
Tengo amplios conocimientos en bases de datos, infraestructuras y servidores para el desarrollo de cualquier tipo de aplicación.

## Hardware



El espacio de trabajo cuenta con un monitor de 34 pulgadas ultrawide, portátil Acer, portátil MacBook Pro, móvil Huawei Honor 8x, móvil iPhone 11, teclado y ratón.

## Software

Para poder desarrollar un proyecto es necesario hacer uso de algunos programas informáticos que hacen posible la ejecución de tareas específicas dentro de un ordenador. Para nuestro proyecto los softwares necesarios son:

### Flutter

Flutter es un framework multiplataforma creado por Google y que compila en nativo. Permite crear aplicación más rápido, diseñar hermosas apps y publicarlas en cualquier tipo de plataforma.

Sus principales características son:

* Desarrollo Rápido: Trae tu aplicación a la vida en cuestión de milisegundos con Hot Reload. Utilice un completo conjunto de widgets totalmente personalizables para crear interfaces nativas en cuestión de minutos.
* UI Expresiva y Flexible: Monta rápidamente funcionalidades con el foco en la experiencia de usuario nativa. La arquitectura en capas permite una completa personalización, que resultan en un renderizado increíblemente rápido y diseños expresivos y flexibles.
* Rendimiento Nativo: Los widgets de Flutter incorporan todas las diferencias críticas entre plataformas, como el scrolling, navegación, iconos y fuentes para proporcionar un rendimiento totalmente nativo tanto en iOS como en Android.

### Visual Studio Code



Es uno de los editores de código más potentes que existe, pudiéndose utilizar en Windows, macOS y Linux. Otorga innumerables posibilidades gracias a la infinidad de extensiones que dispone.

Es una de las principales elecciones para desarrolladores web y javascript, pero para nuestro proyecto en Flutter sirve perfectamente. Se acopla perfectamente a cualquier lenguaje de programación y dispone de extensiones específicas para dichos lenguajes.

Estas son algunas extensiones que se utilizará en este proyecto:

* Flutter
* Dart
* Dart Data Class Generator
* Dart Union Class Generator
* Flutter Intl
* Awesome Flutter Snippets
* Material Icon Theme
* GitHub
* Better Comments
* Image preview

### Android Studio / Xcode



Android Studio y Xcode son dos entornos de desarrollo (IDE) para el desarrollo de aplicaciones móviles.

Android Studio se utiliza para el desarrollo de dispositivos Android en nativo con lenguajes de programación Java y Kotlin. Usa un sistema de compilación flexible basado en Gradle.

Xcode se utiliza para el desarrollo de dispositivos de Apple, sus lenguajes de programación son Swift y Objective-C entre otros.

En estos IDEs se suelen desarrollar las aplicaciones en nativo con dichos lenguajes, esto significa que si nuestro proyecto se desarrollara en nativo se tendrían que realizar dos proyectos uno para Android y otro para iOS. Sin embargo, este proyecto está realizado en Flutter y no necesitaremos desarrollar dos proyectos sino que con el mismo código podremos exportarlo para ambas plataformas.

Únicamente utilizaremos estos entornos para las configuraciones nativas de nuestro proyecto en Flutter ya que como se mencionó anteriormente compila en nativo y es por ello que requiere de ciertas configuraciones tanto en Android como iOS.

### Firebase

### 

### 

Firebase es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Su finalidad es facilitar el desarrollo, crecimiento, monetización y análisis de las aplicaciones.

Sus servicios más destacados son:

* Realtime Database: bases de datos en tiempo real, son No-SQL y almacena los datos como JSON. Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian.
* Autenticación de usuarios: ofrece un sistema de autenticación que permite tanto el registro como el acceso utilizando perfiles de otras plataformas externas.
* Almacenamiento en la nube: sistema de almacenamiento donde los desarrolladores pueden guardar los ficheros de sus aplicaciones y sincronizarlos. Este almacenamiento es de gran ayuda para tratar archivos de los usuarios (por ejemplo, fotografías que hayan subido).
* Crash Reporting: útil para mantener y mejorar la calidad de la app, hay que prestar especial atención a los fallos, por lo que los seguimientos de errores son clave para poder actuar y solucionarlos.
* Cloud Messaging: útil para envío de notificaciones y mensajes a diversos usuarios en tiempo real.

Y otros muchos más servicios. Resulta muy completa y con grandes y numerosos beneficios, aunque hay que mencionar que hay que tener cuidado con la escalabilidad de Firebase porque la capa gratuita es muy limitada.

### GitHub

GitHub es un servicio basado en la nube que aloja un sistema de control de versiones llamado Git.

Permite crear y descargar repositorios. Además de poder administrar los proyectos con el sistema de control Git, es muy útil para trabajar en equipo ya que cada uno puede ir trabajando en diferentes ramas del proyecto y luego hacer merge para fusionar las ramas.

Los comandos más comunes para trabajar con git son:

* Git init
* Git status
* Git branch
* Git checkout “rama”
* Git add .
* Git commit –m “mensaje”
* Git pull
* Git push

El link al repositorio de este proyecto es:  
<https://github.com/Anullos/luna_fashion>

# Planificación del proyecto

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.

# Fase de pruebas

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.

# Conclusiones y trabajos futuros o posibles mejoras

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.

# Apéndices

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.

# Referencias bibliográficas

Como funcionalidad adicional, para que la aplicación cobre más sentido, se agregará las notificaciones push para avisar tanto a los usuarios como a los administradores de los estados de los pedidos.