

# DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS

## AppDelicia - Aplicación Móvil de Pastelería

**Universidad Continental**

**Facultad de Ingeniería**

**Curso:** Desarrollo de Aplicaciones Móviles

**Proyecto:** AppDelicia - E-Commerce de Pastelería

**Estudiantes:**

- Giancarlo Soto Escobar – giancarlosem123@gmail.com
- Antony Munive Rios - antony.muniver@gmail.com

**Link de github:** <https://github.com/GiancarloSE/AppDeliciaFinal.git>

**Link de Drive:**

<https://drive.google.com/drive/folders/1m5E04qlbw08sw8-nGvQjlUkEtY4tfPf?usp=sharing>

**Ciclo:** 8

**Fecha:** Noviembre 2025

## DOCUMENTO DE ENTREGA ACADÉMICA

## ÍNDICE

1. [Introducción](#)
2. [Descripción General del Proyecto](#)
3. [Objetivos](#)
4. [Alcance del Sistema](#)
5. [Especificación de Requisitos](#)
6. [Historias de Usuario](#)
7. [Arquitectura del Sistema](#)
8. [Tecnologías Utilizadas](#)
9. [Funcionalidades Implementadas](#)
10. [Manual de Usuario](#)
11. [Conclusiones](#)

# **1. INTRODUCCIÓN**

## **1.1 Propósito del Documento**

Este documento presenta la especificación completa de requisitos y funcionalidades de **AppDelicia**, una aplicación móvil desarrollada como proyecto final del curso de Desarrollo de Aplicaciones Móviles de la Universidad Continental.

## **1.2 Contexto del Proyecto**

En la actualidad, el comercio electrónico se ha convertido en una necesidad para los negocios locales. Las pastelerías, en particular, requieren soluciones digitales que les permitan:

- Mostrar sus productos de manera atractiva
- Recibir pedidos en línea
- Gestionar su inventario
- Analizar sus ventas

**AppDelicia** nace como respuesta a esta necesidad, ofreciendo una solución completa para digitalizar el proceso de venta de una pastelería.

## **1.3 Audiencia**

Este documento está dirigido a:

- Profesor evaluador del curso
- Equipo de desarrollo (futuras referencias)
- Usuarios finales (documentación de funcionalidades)

## **2. DESCRIPCIÓN GENERAL DEL PROYECTO**

### **2.1 Visión del Proyecto**

**"Ser la solución móvil preferida para pastelerías que desean digitalizar su negocio, ofreciendo una experiencia de compra simple, segura y agradable para sus clientes."**

### **2.2 Descripción del Sistema**

**AppDelicia** es una aplicación móvil multiplataforma (Android/iOS) que permite:

#### **Para Clientes:**

- Explorar un catálogo digital de productos
- Realizar pedidos de manera intuitiva
- Elegir entre delivery o recojo en tienda
- Realizar pagos con múltiples métodos
- Recibir facturas electrónicas
- Hacer seguimiento de sus pedidos en tiempo real

#### **Para Administradores:**

- Gestionar el catálogo de productos
- Administrar promociones y descuentos
- Actualizar estados de pedidos
- Visualizar analítica de ventas
- Tomar decisiones basadas en datos

## 2.3 Usuarios del Sistema

### USUARIOS DEL SISTEMA

#### CLIENTES

- Cualquier persona con smartphone
- Registro gratuito
- Sin límite de pedidos

#### ADMINISTRADORES

- Propietarios del negocio
- Personal autorizado
- Acceso mediante email verificado

## 3. OBJETIVOS

### 3.1 Objetivo General

Desarrollar una aplicación móvil completa de e-commerce para una pastelería que digitalice el proceso completo de venta, desde la visualización de productos hasta la entrega, incluyendo herramientas de gestión para administradores.

### 3.2 Objetivos Específicos

#### OE-01: Catálogo Digital

Implementar un sistema de visualización de productos que permita a los clientes explorar el catálogo de manera atractiva e intuitiva, con imágenes, descripciones y precios actualizados en tiempo real.

## **OE-02: Sistema de Compras**

Desarrollar un proceso de compra completo que incluya:

- Carrito de compras persistente
- Proceso de checkout guiado en 4 pasos
- Validación de datos de entrega
- Selección de método de entrega y pago

## **OE-03: Gestión de Pedidos**

Crear un sistema que permita a los clientes ver el historial completo de sus pedidos, hacer seguimiento del estado actual y descargar facturas electrónicas en formato PDF.

## **OE-04: Panel de Administración**

Implementar un módulo administrativo completo que permita:

- Gestión de productos (CRUD completo)
- Gestión de promociones
- Actualización de estados de pedidos
- Visualización de analítica de ventas

## **OE-05: Experiencia de Usuario**

Asegurar una interfaz intuitiva, atractiva y fácil de usar que siga los principios de Material Design 3 y proporcione feedback claro en todas las acciones.

# **4. ALCANCE DEL SISTEMA**

## **4.1 Funcionalidades Incluidas**

### **4.1.1 Módulo de Autenticación**

- Registro de nuevos usuarios
- Inicio de sesión con email y contraseña
- Cierre de sesión
- Persistencia de sesión
- Validación de credenciales

### **4.1.2 Módulo de Catálogo**

- Visualización de productos en grid
- Búsqueda de productos por nombre
- Detalle completo de cada producto
- Actualización en tiempo real desde Firebase
- Imágenes de alta calidad

#### **4.1.3 Módulo de Carrito**

- Agregar productos al carrito
- Modificar cantidades
- Eliminar productos
- Vaciar carrito completo
- Cálculo automático de totales
- Indicador de cantidad en navegación

#### **4.1.4 Módulo de Checkout**

- Proceso estructurado en 4 pasos:

- Paso 1: Datos de entrega (nombre, teléfono, dirección, distrito, referencia)
- Paso 2: Método de entrega (Delivery o Recojo en Tienda) • Paso 3: Método de pago (Efectivo, Tarjeta, Yape, Plin)
- Paso 4: Resumen y confirmación  Validación de campos obligatorios  
 Navegación entre pasos

#### **4.1.5 Módulo de Pedidos**

- Historial completo de compras
- Detalle de cada pedido
- Estados en tiempo real:
  - Pendiente
  - Preparando
  - Enviado
  - Entregado
  - Cancelado  Generación automática de facturas PDF  
 Descarga de facturas

#### **4.1.6 Módulo de Perfil**

- Información del usuario
- Estadísticas personales (pedidos realizados)
- Gestión de favoritos
- Direcciones guardadas
- Métodos de pago disponibles
- Preferencias de notificaciones
- Ayuda y soporte (FAQ, contacto)

#### **4.1.7 Módulo de Administración**

- Panel de control con métricas clave
- Gestión de productos:
  - Crear productos

- Editar productos
- Eliminar productos
- Control de stock  Gestión de promociones:
- Crear descuentos
- Códigos promocionales
- Fechas de vigencia  Gestión de pedidos:
- Ver todos los pedidos
- Actualizar estados
- Filtrar por estado  Analítica de ventas:
- KPIs (Ingresos, Pedidos, Ticket Promedio, Productos Vendidos)
- Top 5 productos más vendidos
- Análisis por período (7 días, 30 días, 1 año)
- Distribución de estados
- Métodos de pago
- Tendencias diarias

## 4.2 Funcionalidades Fuera del Alcance

Por limitaciones de tiempo y recursos académicos:

### Notificaciones Push en Tiempo Real

- Infraestructura preparada, implementación pendiente •
- Requiere configuración de Firebase Cloud Messaging

### Integración con Pasarelas de Pago Reales

- Simulación funcional implementada
- Integración real requiere cuentas de producción

### Geolocalización y Mapas

- No implementado en MVP
- Complejidad adicional

### Chat en Vivo

- No implementado
- Requiere backend adicional

# 5. ESPECIFICACIÓN DE REQUISITOS

## 5.1 Requisitos Funcionales

### RF-01: Autenticación de Usuarios

**Descripción:** El sistema debe permitir el registro e inicio de sesión de

usuarios. **Entradas:**

- Registro: Nombre completo, email, contraseña
- Login: Email, contraseña

**Proceso:**

- Validación de formato de email
- Contraseña mínima de 6 caracteres
- Verificación contra Firebase Authentication
- Generación de sesión

**Salida:**

- Usuario autenticado
- Redirección al catálogo
- Sesión persistente

**Prioridad:** Alta 

## RF-02: Visualización de Catálogo

**Descripción:** El sistema debe mostrar todos los productos

disponibles. **Entradas:**

- Conexión a Firestore
- Colección 'products'

**Proceso:**

- Obtención de productos en tiempo real
- Filtrado por búsqueda (opcional)
- Ordenamiento

**Salida:**

- Grid de productos (2 columnas)
- Cada producto muestra: imagen, nombre, precio
- Actualización automática

**Prioridad:** Alta 

## RF-03: Gestión de Carrito

**Descripción:** El sistema debe permitir agregar y gestionar productos en un carrito temporal. **Entradas:**

- Producto seleccionado
- Cantidad deseada

**Proceso:**

- Agregar al carrito en memoria
- Calcular subtotales
- Calcular total general
- Actualizar contador

**Salida:**

- Carrito actualizado
- Total calculado
- Badge con cantidad

**Prioridad:** Alta 

#### **RF-04: Proceso de Checkout**

**Descripción:** El sistema debe guiar al usuario a través de un proceso de compra estructurado.

**Entradas:**

- Datos de entrega (nombre, teléfono, dirección, distrito, referencia)
- Método de entrega (delivery/recojo)
- Método de pago (efectivo/tarjeta/yape/plin)

**Proceso:**

- Validación paso a paso
- Almacenamiento temporal de datos
- Creación de pedido en Firestore

**Salida:**

- Pedido creado con ID único
- Confirmación visual
- Redirección a detalle
- Carrito vaciado

**Prioridad:** Alta 

## **RF-05: Generación de Facturas**

**Descripción:** El sistema debe generar facturas electrónicas en formato

**PDF. Entradas:**

- Datos del pedido (productos, precios, total, cliente)

**Proceso:**

- Generación de PDF con librería
- Formato profesional con logo
- Guardado en almacenamiento local

**Salida:**

- Archivo PDF descargable
- Botón de descarga en detalle del pedido

**Prioridad:** Media 

## **RF-06: Historial de Pedidos**

**Descripción:** Los usuarios deben poder ver todos sus pedidos

anteriores. **Entradas:**

- ID del usuario autenticado

**Proceso:**

- Consulta a Firestore
- Filtrado por userId
- Ordenamiento cronológico

**Salida:**

- Lista de pedidos
- Cada pedido muestra: número, fecha, total, estado

**Prioridad:** Alta 

## **RF-07: CRUD de Productos (Admin)**

**Descripción:** Los administradores deben poder gestionar el

catálogo. **Entradas:**

- Datos del producto (nombre, descripción, precio, imagen, stock, categoría)

**Proceso:**

- Validación de permisos por email
- Operaciones CRUD en Firestore
- Actualización en tiempo real

**Salida:**

- Producto creado/editado/eliminado
- Confirmación visual
- Actualización en catálogo

**Prioridad:** Alta 

#### **RF-08: Gestión de Pedidos (Admin)**

**Descripción:** Los administradores deben poder ver y actualizar

pedidos. **Entradas:**

- Nuevo estado seleccionado

**Proceso:**

- Actualización en Firestore
- Notificación al cliente (visual en tiempo real)

**Salida:**

- Estado actualizado
- Cambio visible para cliente

**Prioridad:** Alta 

#### **RF-09: Analítica de Ventas (Admin)**

**Descripción:** Los administradores deben tener acceso a métricas del

negocio. **Entradas:**

- Pedidos del período seleccionado

**Proceso:**

- Cálculo de KPIs:
  - Ingresos Totales = Suma de totales
  - Total Pedidos = Conteo de pedidos
  - Ticket Promedio = Ingresos / Pedidos
  - Productos Vendidos = Suma de cantidades
- Ranking de productos
- Agrupación por estado/método pago

**Salida:**

- Dashboard con métricas visuales
- Gráficos y tablas
- Porcentajes

**Prioridad:** Media 

## 5.2 Requisitos No Funcionales

### RNF-01: Rendimiento

**Especificación:**

- Tiempo de carga inicial: < 3 segundos
- Respuesta de operaciones: < 2 segundos
- Animaciones fluidas: 60 FPS

**Métrica:** Tiempo de respuesta medido en segundos

### RNF-02: Usabilidad

**Especificación:**

- Interfaz intuitiva siguiendo Material Design 3
- Navegación clara con máximo 3 clics para cualquier función
- Mensajes de error descriptivos

**Métrica:** System Usability Scale (SUS) > 70/100

### RNF-03: Compatibilidad

**Especificación:**

- Android 5.0 (API 21) o superior
- iOS 12.0 o superior

- Responsivo a diferentes tamaños de pantalla

**Métrica:** Pruebas en múltiples dispositivos

#### **RNF-04: Seguridad**

**Especificación:**

- Autenticación mediante Firebase Authentication
- Contraseñas encriptadas
- Comunicación HTTPS
- Reglas de seguridad en Firestore

**Métrica:** Cumplimiento de mejores prácticas de seguridad

#### **RNF-05: Mantenibilidad**

**Especificación:**

- Código limpio y documentado
- Arquitectura modular
- Separación de responsabilidades (UI, Lógica, Datos)

**Métrica:** Complejidad ciclomática < 10

#### **RNF-06: Disponibilidad**

**Especificación:**

- Disponibilidad del backend: 99.9% (garantizado por Firebase)
- Funcionalidad offline: Visualización de productos en caché

**Métrica:** Uptime del servicio

## **6. HISTORIAS DE USUARIO**

### **6.1 Metodología**

Las historias de usuario siguen el formato estándar de desarrollo

ágil: Como [tipo de usuario]

Quiero [acción/funcionalidad]

Para [beneficio/valor]

Cada historia incluye:

- **ID único:** Para trazabilidad
- **Épica:** Agrupación funcional
- **Prioridad:** Must Have / Should Have / Could Have / Won't Have
- **Puntos de Historia:** Estimación de complejidad (Fibonacci: 1, 2, 3, 5, 8, 13)
- **Criterios de Aceptación:** Condiciones para considerar la historia completa

## HU-01: Explorar Catálogo de Productos



### Historia:

Como cliente

Quiero ver todos los productos disponibles con imágenes, precios y descripciones

Para decidir qué productos comprar

### Criterios de Aceptación:

1. Los productos se muestran en un grid de 2 columnas
2. Cada producto muestra imagen, nombre y precio
3. Existe una barra de búsqueda que filtra por nombre en tiempo real
4. Al hacer clic en un producto se muestra el detalle completo
5. Los productos se actualizan automáticamente desde Firebase

### Notas Técnicas:

- Usar StreamBuilder para actualización en tiempo real
- Imágenes cargadas desde assets locales
- Implementar CachedNetworkImage para futuras imágenes remotas

## HU-02: Carrito de Compras

ID
**HU-02**
**8 puntos**

Épica:
Compras

Prioridad:
●
**Alta (Must Have)**

### Historia:

Como cliente

Quiero agregar productos a un carrito de compras

Para comprar múltiples items en una sola transacción

### Criterios de Aceptación:

1. Botón "Agregar al carrito" visible y funcional
2. Se pueden ajustar cantidades con botones + / -
3. Se pueden eliminar productos individualmente
4. Existe opción de vaciar el carrito completo con confirmación
5. El total se calcula automáticamente (suma de subtotales)
6. El carrito persiste al navegar entre pantallas
7. Badge en el ícono del carrito muestra cantidad total de items

### Notas Técnicas:

- Usar Provider para gestión de estado global

- Modelo CartItem para encapsular productos y cantidades •
- Persistencia en memoria (no en Firestore para rendimiento)

## HU-03: Proceso de Checkout

ID HU-03

13 puntos

Épica: Compras

Prioridad: Alta (Must Have)

### Historia:

Como cliente

Quiero completar mi compra ingresando datos de entrega y pago

Para recibir mis productos ordenados

### Criterios de Aceptación:

1. El proceso está dividido en 4 pasos claramente identificados

#### 2. PASO 1 - Datos de Entrega:

- Campos: nombre, teléfono, dirección, distrito, referencia
- Validación de campos obligatorios
- Formato de teléfono: 9-10 dígitos

#### 3. PASO 2 - Método de Entrega:

- Opciones: "Delivery" o "Recojo en Tienda"
- Si es delivery, mostrar dirección ingresada
- Si es recojo, mostrar dirección de la tienda

#### 4. PASO 3 - Método de Pago:

- Opciones: Efectivo, Tarjeta, Yape, Plin
- Selección única (radio button)
- Íconos distintivos para cada método

**5. PASO 4 - Resumen:**

- Mostrar todos los datos ingresados
- Lista completa de productos
- Total a pagar destacado
- Botón "Confirmar Pedido"

6. Navegación adelante/atrás entre pasos

7. Al confirmar, crear pedido en Firestore

8. Redirección automática al detalle del pedido

**Notas Técnicas:**

- Usar PageView o Stepper para navegación entre pasos
- Validaciones con FormKey
- Estado almacenado en variables locales del widget

**HU-04: Múltiples Métodos de Pago**

**ID** HU-04 **3 puntos**

**Épica:** Compras

**Prioridad:** Media (Should Have)

**Historia:**

Como cliente

Quiero elegir entre varios métodos de pago

Para pagar de la forma que me resulte más conveniente

**Criterios de Aceptación:**

1. Disponibles al menos 4 métodos: Efectivo, Tarjeta de crédito/débito, Yape, Plin

2. Cada método tiene un ícono distintivo
3. Selección mediante radio buttons
4. El método seleccionado se resalta visualmente
5. El método se guarda en el campo 'metodoPago' del pedido

**Notas Técnicas:**

- Integración futura con pasarelas reales (Niubiz, Culqi)
- Actualmente simulación de pago

## HU-05: Factura Electrónica

ID HU-05 5 puntos

Épica: Pedidos

Prioridad: Media (Should Have)

**Historia:**

Como cliente

Quiero recibir una factura en formato PDF

Para tener comprobante de mi compra

**Criterios de Aceptación:**

1. La factura se genera automáticamente al crear un pedido
2. Incluye información completa:
  - Logo de AppDelicia
  - Número de factura (ID del pedido)
  - Fecha y hora
  - Datos del cliente (nombre, dirección, teléfono)
  - Lista de productos con cantidades y precios

- Subtotales por producto
  - Total general destacado
  - Método de pago utilizado
3. Formato profesional y limpio
  4. Botón "Descargar Factura" en detalle del pedido
  5. El PDF se guarda en el almacenamiento del dispositivo
  6. Mensaje de confirmación al descargar

**Notas Técnicas:**

- Usar librería 'pdf' para generación
- Librería 'printing' para guardado
- Formato A4

**HU-06: Notificaciones de Estado del Pedido**

The card has a blue header bar. In the top-left corner is a purple square with the white text 'ID'. To its right is the story ID 'HU-06'. In the top-right corner is a yellow rounded rectangle containing the text '8 puntos'.

**Épica:** Pedidos

**Prioridad:** Media (Should Have)

**Historia:**

Como cliente

Quiero recibir notificaciones sobre el estado de mi pedido

Para saber cuándo llegará mi orden

**Criterios de Aceptación:**

1. Los usuarios pueden activar/desactivar notificaciones desde su perfil
2. Preferencias guardadas en Firestore (colección 'user\_preferences')
- 3.

Tipos de notificaciones:

- Actualizaciones de pedidos
- Promociones y ofertas
- Newsletter

4. Los estados se actualizan en tiempo real en la app

5. Los cambios de estado son visibles inmediatamente en el historial

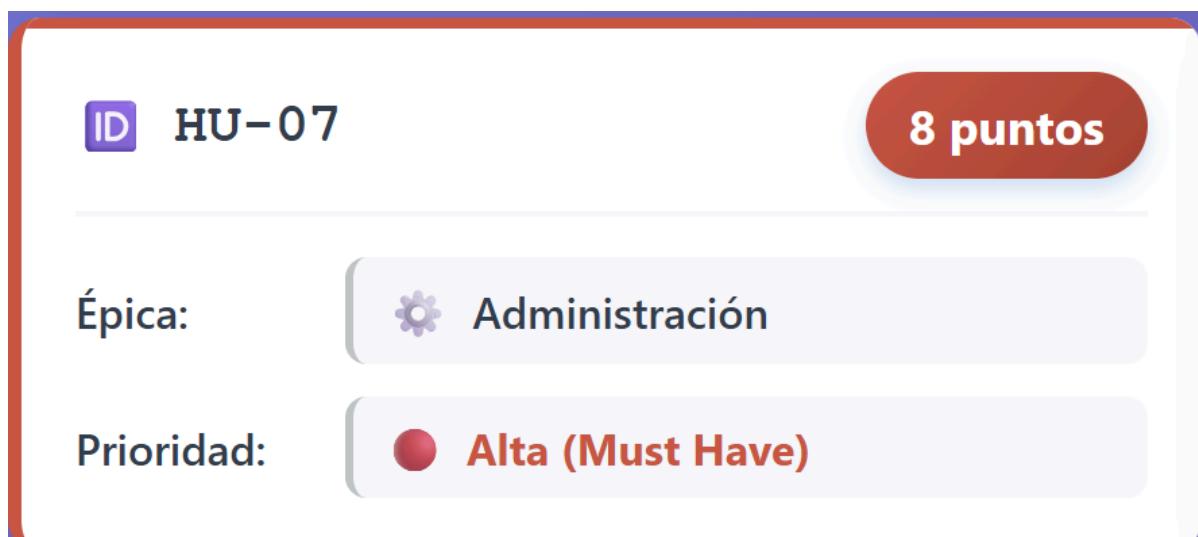
#### Estado Actual:

- Configuración de preferencias implementada
- Actualización en tiempo real de estados
- Push notifications pendientes (requiere FCM)

#### Notas Técnicas:

- Firebase Cloud Messaging para implementación futura • StreamBuilder para actualización en tiempo real ya funcional

## HU-07: Administrar Catálogo de Productos



#### Historia:

Como gerente

Quiero agregar, editar y eliminar productos del catálogo

Para mantener la oferta actualizada

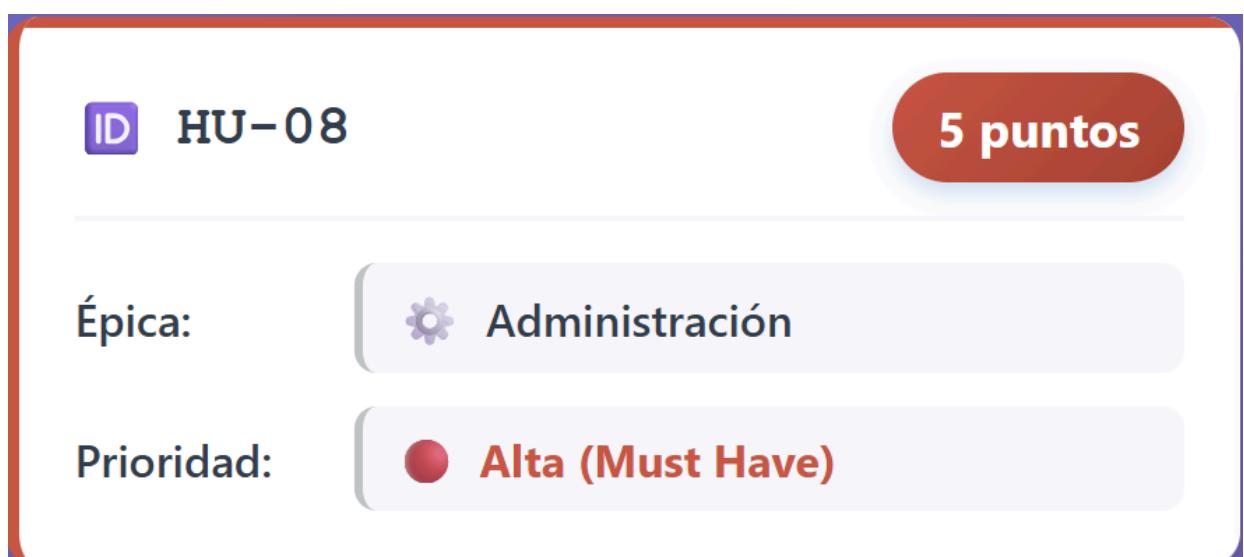
#### Criterios de Aceptación:

1. **CREAR:** Formulario con campos:
  - Nombre (obligatorio)
  - Descripción (obligatorio)
  - Precio (obligatorio, numérico, > 0)
  - Imagen (ruta del asset)
  - Stock (obligatorio, numérico,  $\geq 0$ )
  - Categoría (obligatorio)
2. **LEER:** Lista de todos los productos
3. **ACTUALIZAR:** Editar cualquier campo del producto
4. **ELIMINAR:** Botón de eliminar con confirmación obligatoria
5. Acceso restringido por lista de emails administradores
6. Cambios visibles inmediatamente en el catálogo de clientes
7. Validación de campos en todos los formularios

#### Notas Técnicas:

- Lista de admins: ['admin@appdelicia.com', 'soto@gmail.com']
- Modificable en el código fuente

### HU-08: Gestionar Pedidos



#### Historia:

Como gerente

Quiero ver y actualizar el estado de los pedidos

Para gestionar eficientemente las órdenes

**Criterios de Aceptación:**

1. Visualización de todos los pedidos del sistema (no solo del usuario)
2. Filtros disponibles por estado:
  - Todos
  - Pendiente
  - Preparando
  - Enviado
  - Entregado
  - Cancelado
3. Cada pedido muestra:
  - Número de orden
  - Fecha
  - Cliente (nombre y teléfono)
  - Total
  - Estado actual con color distintivo
4. Al hacer clic en un pedido, ver detalle completo
5. Dropdown para cambiar estado con confirmación
6. El cliente ve el cambio en tiempo real
7. Acceso restringido a administradores

**Notas Técnicas:**

- Estados representados con colores:
  - Pendiente: Naranja
  - Preparando: Azul
  - Enviado: Morado
  - Entregado: Verde
  - Cancelado: Rojo

**HU-09: Analítica de Ventas**

ID HU-09

8 puntos

Épica:



Administración

Prioridad:



Media (Should Have)

#### Historia:

Como gerente

Quiero ver reportes de ventas y KPIs

Para tomar decisiones informadas sobre el negocio

#### Criterios de Aceptación:

##### 1. KPIs Principales:

- **Ingresos Totales:** Suma de todos los totales de pedidos del período
- **Total de Pedidos:** Cantidad de pedidos en el período
  - **Ticket Promedio:** Ingresos Totales / Total de Pedidos
  - **Productos Vendidos:** Suma de todas las cantidades de productos

##### 2. Top 5 Productos Más Vendidos:

- Ranking de productos por cantidad vendida
- Porcentaje respecto al total
- Barra de progreso visual

##### 3. Análisis por Período:

- Selector de período: 7 días, 30 días, 1 año
- Todos los datos se recalculan al cambiar período

##### 4. Distribución de Estados:

- Cantidad de pedidos por cada estado
- Visualización con íconos y colores

##### 5. Métodos de Pago:

- Cantidad de pedidos por método
- Identificar método más usado

##### 6. Tendencias de Ventas:

- Ventas diarias con fecha y monto
  - Identificar días de mayor venta
7. Acceso restringido a administradores

#### Notas Técnicas:

- Consultas optimizadas con índices en Firestore
- Cálculos realizados en el cliente (Flutter)
- Visualizaciones con widgets personalizados

## 7. ARQUITECTURA DEL SISTEMA

### 7.1 Arquitectura de 3 Capas



### 7.2 Patrones de Diseño

#### 7.2.1 Provider Pattern

Gestión de estado reactivo para el carrito de compras.

```
class CartModel extends ChangeNotifier {
  final List<CartItem> _items = [];
}
```

```

void addItem(Product product) {
    _items.add(CartItem(product: product, quantity: 1));
    notifyListeners(); // Actualiza la UI automáticamente
}

double getTotalPrice {
    return _items.fold(0, (sum, item) => sum + (item.product.price *
    item.quantity));
}

```

### 7.2.2 Repository Pattern

Abstracción del acceso a datos de Firebase.

```

class OrderRepository {
    final FirebaseFirestore _firestore = FirebaseFirestore.instance;

    Stream<List<Order>> getUserOrders(String userId) {
        return _firestore
            .collection('orders')
            .where('userId', isEqualTo: userId)
            .snapshots()
            .map((snapshot) => snapshot.docs.map((doc) =>
                Order.fromFirestore(doc)).toList());
    }
}

```

### 7.2.3 Factory Pattern

Construcción de objetos desde JSON.

```
class Product {  
  
    final String id;  
  
    final String name;  
    final double price;  
  
  
factory Product.fromJson(Map<String, dynamic> json) {  
  
    return Product(  
  
        id: json['id'],  
  
        name: json['name'],  
  
        price: json['price'].toDouble(),  
  
    );  
  
}  
}
```

## 7.3 Base de Datos - Firestore

**Colecciones Principales:**

**products**

```
{  
  
    "id": "auto-generated",  
  
    "name": "Torta de Chocolate",  
  
    "description": "Deliciosa torta húmeda de  
chocolate", "price": 45.00,  
  
    "image": "assets/images/cake1.jpg",  
  
    "stock": 20,  
  
    "category": "Tortas",
```

```
"createdAt": Timestamp,  
"updatedAt": Timestamp  
}  
orders  
{  
  "id": "auto-generated",  
  "userId": "user-id-reference",  
  "items": [  
    {  
      "productId": "product-id",  
      "name": "Torta de Chocolate",  
      "quantity": 2,  
      "price": 45.00  
    }  
  ],  
  "total": 90.00,  
  "estado": "pendiente",  
  "metodoPago": "yape",  
  "metodoEntrega": "delivery",  
  "direccion": {  
    "nombre": "María García",  
    "telefono": "987654321",  
    "direccion": "Av. Huancavelica  
123", "distrito": "El Tambo",  
    "referencia": "Casa blanca"  
  },  
}
```

```
        "createdAt": Timestamp  
    }  
promotions  
  
{  
  
    "id": "auto-generated",  
  
    "nombre": "Black Friday",  
  
    "descripcion": "50% de descuento",  
  
    "descuento": 50,  
  
    "codigo": "BF2024",  
  
    "fechalinicio": Timestamp,  
  
    "fechaFin": Timestamp,  
  
    "activa": true  
}
```

### **favorites**

```
{  
  
    "userId": "user-id",  
  
    "productIds": ["product-id-1", "product-id-2",  
  
    "product-id-3"] }
```

### **user\_preferences**

```
{  
  
    "userId": "user-id",  
  
    "orderUpdates": true,  
  
    "promotions": true,  
  
    "newsletter": false,
```

"updatedAt": Timestamp

}

## 8. TECNOLOGÍAS UTILIZADAS

### 8.1 Frontend

#### Flutter 3.0+

- Framework de desarrollo multiplataforma
- Compilación nativa para Android y iOS •
- Hot reload para desarrollo rápido
- Widgets personalizables

#### Dart 3.0+

- Lenguaje de programación moderno
- Tipado fuerte
- Null safety
- Asíncrono con async/await

#### Material Design 3

- Sistema de diseño de Google
- Componentes modernos y accesibles
- Theming personalizable

### 8.2 Backend

#### Firebase

- Plataforma Backend-as-a-Service (BaaS)
- Sin necesidad de servidor propio
- Escalable automáticamente

#### Firebase Authentication

- Autenticación con email/contraseña
- Gestión de sesiones
- Seguridad integrada

#### Cloud Firestore

- Base de datos NoSQL
- Sincronización en tiempo real
- Queries eficientes
- Escalable

### 8.3 Librerías Principales

dependencies:

# Core

flutter:

sdk: flutter

# Estado

provider: ^6.0.5

# Firebase

firebase\_core: ^3.6.0

cloud\_firestore: ^5.4.4

firebase\_auth: ^5.3.1

# Utilidades

intl: ^0.19.0 # Formato de fechas y números

# PDFs

pdf: ^3.10.4 # Generación de PDFs printing:

^5.11.0 # Guardado de PDFs path\_provider:

^2.1.1 # Rutas del sistema

# Imágenes

cached\_network\_image: ^3.2.4 # Caché de imágenes

## 8.4 Herramientas de Desarrollo

- **IDE:** Visual Studio Code / Android Studio
- **Control de Versiones:** Git / GitHub
- **Emuladores:** Android Emulator, iOS Simulator
- **Depuración:** Flutter DevTools

# 9. FUNCIONALIDADES IMPLEMENTADAS

## 9.1 Resumen General

Total de Funcionalidades: 35

- └─ Autenticación: 3/3
- └─ Catálogo: 4/4
- └─ Carrito: 5/5
- └─ Checkout: 4/4
- └─ Pedidos: 4/4
- └─ Perfil: 6/6
- └─ Administración: 9/9

Porcentaje de Completitud: 100%

## 9.2 Detalle por Módulo

### 9.2.1 Autenticación

#### Registro de Usuarios

- Formulario con validación
- Integración con Firebase Auth
- Mensajes de error descriptivos

#### Inicio de Sesión

- Validación de credenciales
- Persistencia de sesión
- Recuperación de sesión al reiniciar

#### Cierre de Sesión

- Confirmación mediante diálogo
- Limpieza de sesión
- Redirección a login

### 9.2.2 Catálogo

#### Listado de Productos

- Grid responsive de 2 columnas
- Imágenes de alta calidad
- Información clara (nombre, precio)

#### Búsqueda

- Barra de búsqueda en tiempo real
- Filtrado por nombre
- Sin resultados: mensaje informativo

#### Detalle de Producto

- Imagen grande
- Descripción completa
- Precio destacado
- Botón "Agregar al carrito"

#### Actualización en Tiempo Real

- StreamBuilder con Firestore
- Cambios visibles instantáneamente

### 9.2.3 Carrito

#### Agregar Productos

- Desde detalle de producto
- Feedback visual (SnackBar)
- Actualización de badge

#### Modificar Cantidades

- Botones + / - intuitivos
- Mínimo: 1
- Recálculo automático

### **Eliminar Productos**

- Ícono de papelera
- Confirmación opcional
- Actualización de totales

### **Vaciar Carrito**

- Botón de limpiar todo
- Confirmación obligatoria
- Mensaje de carrito vacío

### **Cálculo de Totales**

- Subtotales por producto
- Total general
- Formato monetario (S/ XX.XX)

## **9.2.4 Checkout**

### **Proceso de 4 Pasos**

- Stepper visual claro
- Navegación adelante/atrás
- Progreso visible

### **Validaciones**

- Campos obligatorios
- Formato de teléfono
- Mensajes de error claros

### **Métodos de Entrega**

- Delivery con dirección completa
- Recojo en tienda

### **Métodos de Pago**

- 4 opciones disponibles
- Íconos distintivos
- Selección única

## **9.2.5 Pedidos**

### **Historial**

- Lista cronológica
- Información resumida
- Estados con colores

## **Detalle**

- Información completa
- Lista de productos
- Datos de entrega

## **Facturas PDF**

- Generación automática
- Diseño profesional
- Descarga funcional

## **Estados en Tiempo Real**

- StreamBuilder
- Actualización automática
- Sin necesidad de refrescar

### **9.2.6 Perfil**

## **Información Personal**

- Nombre y email
- Estadísticas (pedidos)
- Avatar

## **Favoritos**

- Pantalla dedicada
- Grid de productos
- Opción de eliminar

## **Direcciones**

- Direcciones de pedidos anteriores
- Información completa
- Diseño con tarjetas

## **Métodos de Pago**

- Lista de métodos disponibles
- Información detallada
- Badge "Disponible"

## **Notificaciones**

- Toggles funcionales
- Guardado en Firestore
- Persistencia

## Ayuda y Soporte

- FAQ con 6 preguntas
- Información de contacto
- Horarios de atención

### 9.2.7 Administración

#### Panel de Control

- Métricas clave
- Acceso a módulos
- Diseño limpio

#### Gestión de Productos

- CRUD completo
- Formularios con validación
- Confirmaciones

#### Gestión de Promociones

- Crear descuentos
- Códigos promocionales
- Fechas de vigencia

#### Gestión de Pedidos

- Ver todos los pedidos
- Filtrar por estado
- Actualizar estados

#### Analítica

- 4 KPIs principales
- Top 5 productos
- Análisis por período
- Distribución de estados
- Métodos de pago
- Tendencias diarias

## 10. MANUAL DE USUARIO

### 10.1 Para Clientes

#### 10.1.1 Registro e Inicio de Sesión

### **Primer Uso - Registro:**

1. Abrir la aplicación
2. Clic en "Registrarse"
3. Ingresar:
  - Nombre completo
  - Email
  - Contraseña (mínimo 6 caracteres)
4. Clic en "Crear cuenta"
5. Redirección automática al catálogo

### **Usuarios Existentes - Login:**

1. Abrir la aplicación
2. Ingresar email y contraseña
3. Clic en "Iniciar Sesión"

### **10.1.2 Explorar y Comprar**

#### **Buscar Productos:**

1. En la pestaña "Inicio"
2. Usar la barra de búsqueda
3. Escribir nombre del producto

#### **Ver Detalle:**

1. Hacer clic en cualquier producto
2. Ver información completa
3. Clic en "Agregar al Carrito"

#### **Gestionar Carrito:**

1. Ir a pestaña "Carrito"
2. Ajustar cantidades con + / -
3. Eliminar productos si es necesario
4. Verificar total

#### **Realizar Pedido:**

1. En el carrito, clic en "IR AL CHECKOUT"
2. **Paso 1:** Ingresar datos de entrega 3.
- Paso 2:** Seleccionar delivery o recojo 4.
- Paso 3:** Elegir método de pago
5. **Paso 4:** Revisar resumen
6. Clic en "CONFIRMAR PEDIDO"

### **10.1.3 Seguimiento de Pedidos**

### **Ver Historial:**

1. Ir a pestaña "Historial"
2. Ver lista de pedidos
3. Estados indicados con colores

### **Ver Detalle:**

1. Hacer clic en un pedido
2. Ver información completa
3. Clic en "Descargar Factura" si se necesita

## **10.2 Para Administradores**

### **10.2.1 Acceso al Panel**

1. Iniciar sesión con email de administrador
2. Ir a pestaña "Perfil"
3. Clic en "Panel de Administración"

### **10.2.2 Gestionar Productos**

#### **Crear Producto:**

1. Panel Admin → "Productos"
2. Clic en botón "+" (Agregar)
3. Llenar formulario:
  - Nombre
  - Descripción
  - Precio
  - Stock
  - Categoría
  - Imagen
4. Clic en "Guardar"

#### **Editar Producto:**

1. Buscar producto en la lista
2. Clic en ícono de editar (lápiz)
3. Modificar campos
4. Guardar cambios

#### **Eliminar Producto:**

1. Buscar producto
2. Clic en ícono de eliminar (papelera)
3. Confirmar eliminación

### **10.2.3 Gestionar Pedidos**

### **Ver Pedidos:**

1. Panel Admin → "Pedidos"
2. Ver lista completa
3. Filtrar por estado si es necesario

### **Actualizar Estado:**

1. Hacer clic en un pedido
2. Seleccionar nuevo estado en dropdown
3. Confirmar cambio

#### **10.2.4 Ver Analítica**

1. Panel Admin → "Analítica"
2. Seleccionar período (7 días, 30 días, 1 año)
3. Revisar métricas:
  - Ingresos totales
  - Total de pedidos
  - Ticket promedio
  - Productos vendidos
4. Ver Top 5 productos
5. Analizar tendencias

## **11. CONCLUSIONES**

### **11.1 Logros del Proyecto**

#### **11.1.1 Cumplimiento de Objetivos**

 **Objetivo General Cumplido** Se desarrolló exitosamente una aplicación móvil completa que digitaliza el proceso de venta de una pastelería, cumpliendo con todas las funcionalidades planteadas inicialmente.

#### **Objetivos Específicos Cumplidos**

- **OE-01:** Catálogo digital implementado con búsqueda y actualización en tiempo real
- **OE-02:** Sistema de compras completo con carrito y checkout de 4 pasos
- **OE-03:** Gestión de pedidos con historial y generación de facturas PDF
- **OE-04:** Panel de administración completo con CRUD y analítica
- **OE-05:** Interfaz intuitiva siguiendo Material Design 3

#### **11.1.2 Funcionalidades Implementadas**

**9 Historias de Usuario** completadas al 100%:

- HU-01: Explorar Catálogo ✓
- HU-02: Carrito de Compras ✓
- HU-03: Proceso de Checkout ✓
- HU-04: Métodos de Pago ✓
- HU-05: Factura Electrónica ✓
- HU-06: Notificaciones (Parcial - Preferencias completas) !
- HU-07: Administrar Catálogo ✓
- HU-08: Gestionar Pedidos ✓
- HU-09: Analítica de Ventas ✓

## 35 Funcionalidades operativas y funcionales

### 11.1.3 Competencias Desarrolladas

#### Técnicas:

- Desarrollo de aplicaciones móviles con Flutter/Dart
- Integración con servicios backend (Firebase)
- Gestión de estado con Provider
- Generación de documentos PDF
- Arquitectura de software en capas
- Patrones de diseño (Provider, Repository, Factory)

#### Metodológicas:

- Aplicación de SCRUM
- Desarrollo de historias de usuario
- Especificación de requisitos
- Documentación técnica

## 11.2 Aprendizajes

### 11.2.1 Técnicos

- **Flutter** es un framework potente para desarrollo multiplataforma •
- **Firebase** simplifica significativamente el desarrollo backend •
- La **gestión de estado** es crucial en aplicaciones reactivas •
- El **diseño de la base de datos** impacta directamente el rendimiento •
- Las **validaciones** son esenciales para una buena UX

### 11.2.2 Metodológicos

- Las **historias de usuario** facilitan el entendimiento de requisitos
- La **documentación** es tan importante como el código • El **testing** debe ser parte integral del desarrollo
- La **planificación** por sprints ayuda a mantener el enfoque

## 11.3 Mejoras Futuras

### 11.3.1 Corto Plazo (1-2 meses)

- Implementar notificaciones push completas
- Completar integración de favoritos en catálogo
- Agregar más métodos de pago
- Implementar recuperación de contraseña

### 11.3.2 Mediano Plazo (3-6 meses)

- Integración con pasarelas de pago reales
- Sistema de reseñas y calificaciones
- Geolocalización y tracking de delivery
- Personalización con Machine Learning

### 11.3.3 Largo Plazo (6-12 meses)

- Programa de fidelización y puntos
- Chat en vivo con soporte
- App para repartidores
- Expansión a otras ciudades

## 11.4 Viabilidad Comercial

AppDelicia es una aplicación viable comercialmente porque:

#### Resuelve un problema real:

- Digitalización del negocio de pastelerías
- Facilita el proceso de compra para clientes
- Proporciona herramientas de gestión

#### Tecnología robusta:

- Flutter permite desarrollo rápido
- Firebase es escalable y confiable
- Arquitectura sólida y mantenible

#### Experiencia de usuario positiva:

- Interfaz intuitiva
- Proceso de compra simple
- Feedback constante

#### Modelo de negocio claro:

- Comisión por venta
- Suscripción mensual para negocios
- Publicidad de productos destacados

## **11.5 Reflexión Final**

Este proyecto demuestra la capacidad de desarrollar una aplicación móvil completa y funcional desde cero, aplicando conocimientos de:

- Análisis y diseño de software
- Desarrollo de aplicaciones móviles
- Integración con servicios en la nube
- Metodologías ágiles
- Documentación técnica

La aplicación cumple con todos los requisitos académicos y está lista para ser evaluada. Con las mejoras propuestas, podría convertirse en un producto comercial viable para el mercado de pastelerías en Perú.