

APLICACION MOVIL FLUTTER

# FashionMarket

Plataforma de comercio electronico — App movil multiplataforma

---

ALUMNO

Jose Manuel Ortega Soto

ASIGNATURA

Sistema de Gestion  
Empresarial

PROFESOR

David Soto

CENTRO

Victoriafp

REPOSITORIO GITHUB

[github.com/JoseManuelOS/FashionMarketFlutter](https://github.com/JoseManuelOS/FashionMarketFlutter)

# Indice de contenidos



01	Resumen ejecutivo
02	Objetivos del proyecto
03	Problema que resuelve
04	Publico objetivo
05	Descripcion general
06	Funcionalidades principales
07	Flujo de compra
08	Pedidos, devoluciones y facturacion
09	Panel de administracion
10	Seguridad y confianza
11	Arquitectura y tecnologias
12	Integraciones externas
13	Modelos de datos
14	Navegacion y rutas
15	Estado actual y operacion

**16** Impacto y beneficios

---

**17** Mejoras futuras

---

# 1. Resumen ejecutivo

---

**FashionMarket Flutter** es la aplicación móvil multiplataforma del ecosistema FashionMarket, desarrollada con **Flutter 3** y **Dart 3.8**. Complementa la tienda web (Astro) proporcionando una experiencia nativa optimizada para iOS y Android, con soporte adicional para Web.

La aplicación implementa el ciclo completo de comercio electrónico: catálogo de productos con búsqueda y filtrado avanzado, carrito persistente offline, pasarela de pago Stripe, gestión de pedidos, favoritos, devoluciones, y un panel de administración integrado para gestionar toda la operativa del negocio desde el dispositivo móvil.

Construida siguiendo la arquitectura **Clean Architecture por Features**, utiliza **Riverpod** para la gestión de estado, **Freezed** para modelos inmutables, **GoRouter** para navegación declarativa, **Hive** para persistencia local y **Supabase** como backend completo con autenticación, base de datos PostgreSQL y suscripciones en tiempo real.

---

## 2. Objetivos del proyecto

---

- **Experiencia móvil nativa:** ofrecer una app fluida y optimizada para dispositivos móviles con rendimiento nativo en iOS y Android.
  - **Comercio electrónico completo:** cubrir todo el ciclo de compra, desde el descubrimiento del producto hasta la entrega y posible devolución.
  - **Administración móvil:** permitir al administrador gestionar productos, pedidos, devoluciones, descuentos y configuración desde cualquier lugar.
  - **Persistencia offline:** mantener el carrito y preferencias del usuario disponibles sin conexión mediante almacenamiento local con Hive.
  - **Tiempo real:** reflejar cambios en ofertas, stock y configuración de forma instantánea mediante Supabase Realtime.
  - **Arquitectura mantenible:** seguir principios de Clean Architecture con separación de capas y tipado fuerte para facilitar el desarrollo y mantenimiento.
  - **Manejo robusto de errores:** utilizar programación funcional con `Either<Failure, T>` (fpdart) y una jerarquía de excepciones tipadas.
-

### 3. Problema que resuelve

---

PROBLEMA	SOLUCION EN LA APP
Los usuarios prefieren apps nativas para compras recurrentes	App Flutter multiplataforma con rendimiento nativo y navegacion fluida
Perdida de carrito al cerrar la app o sin conexion	Carrito persistente con Hive — funciona offline y se mantiene entre sesiones
El administrador necesita gestionar desde el movil	Panel de administracion completo integrado en la misma app con 13 pantallas dedicadas
Actualizaciones lentas de ofertas y stock	Suscripciones Realtime de Supabase que reflejan cambios al instante
Experiencia de compra lenta en web movil	Rendimiento nativo con animaciones Hero, carga de imagenes cacheadas y scroll infinito
Gestion manual de devoluciones y cancelaciones	Flujos automatizados con reembolso Stripe, reposicion de stock y notificaciones por email
Busqueda y filtrado limitados	Sistema completo de filtros por categoria, talla, color, precio, ofertas y ordenacion multiple
Falta de metricas de negocio en movil	Dashboard con KPIs, graficos de ventas (fl_chart) y alertas de stock bajo

---

### 4. Publico objetivo

---

- **Compradores habituales:** usuarios que prefieren una app nativa para realizar compras recurrentes con carrito persistente y notificaciones.
- **Usuarios moviles:** personas que navegan y compran principalmente desde su smartphone, buscando una experiencia rapida y fluida.
- **Administradores en movimiento:** gestores de la tienda que necesitan controlar pedidos, stock, devoluciones y configuracion desde cualquier lugar.

- **Profesorado y evaluadores:** el proyecto demuestra el dominio de Flutter, arquitectura limpia, gestion de estado avanzada y desarrollo multiplataforma.
- 

## 5. Descripcion general

---

La aplicacion se divide en dos grandes areas funcionales: la **zona publica** (cliente) y la **zona de administracion**.

### 5.1 Zona publica (cliente)

- Splash animado con logo SVG animado por paths.
- Pantalla de inicio con carrusel hero configurable, categorias, productos destacados y ofertas en tiempo real.
- Catalogo de productos con grid escalonado, busqueda por texto, filtros avanzados (categoria, talla, color, precio, ofertas) y ordenacion multiple.
- Detalle de producto con galeria de imagenes, selector de color y talla, selector de cantidad con validacion de stock, boton de añadir al carrito con feedback visual, y productos relacionados.
- Carrito persistente con Hive (funciona offline), slide-over lateral, control de cantidades con validacion de stock real, y calculo automatico de totales y ahorros.
- Checkout multistep: datos de envio, metodo de envio, codigo de descuento y resumen final con redireccion a Stripe Checkout.
- Gestion de pedidos: listado con filtros por estado, detalle con linea temporal de estados, cancelacion con motivo, solicitud de devolucion y descarga de factura.
- Favoritos sincronizados con Supabase, toggle optimista con rollback ante error.
- Perfil de usuario con edicion de datos personales, direcciones y gestion de cuenta.
- Autenticacion completa: login con email, registro con bienvenida y newsletter, OAuth con Google y Apple, y recuperacion de contraseña.

### 5.2 Zona de administracion

- Dashboard con KPIs en tiempo real: ventas del dia/mes, pedidos pendientes, stock bajo, producto mas vendido.
- Graficos de ventas interactivos con periodos configurables (fl\_chart).

- CRUD completo de productos: nombre, descripcion, precio, categoria, variantes (talla/color/stock), imagenes via Cloudinary, estado activo/inactivo y ofertas.
  - Gestion de pedidos: listado filtrable, cambio de estado, envio de tracking, cancelacion con reembolso, aceptacion/rechazo de devoluciones (totales y parciales).
  - Codigos de descuento: creacion, activacion/desactivacion, porcentaje/fijo, compra minima, limites de uso, un solo uso por cliente, fecha de expiracion.
  - Carrusel de la home: CRUD de slides con titulo, subtítulo, imagen, enlace CTA, duracion y orden.
  - Gestion de categorias: CRUD con nombre, slug, descripcion, imagen y orden de visualizacion.
  - Gestion de usuarios: listado de clientes con estadísticas de pedidos y gasto total.
  - Centro de notificaciones: pedidos nuevos (24h), variantes con stock bajo, productos agotados, pedidos pendientes.
  - Newsletter: envio de campañas masivas con asunto, contenido, imagen, código promocional y botón CTA.
  - Facturas: gestion y envio de facturas a clientes.
  - Configuración de animaciones del sitio.
- 

## 6. Funcionalidades principales

---

### 6.1 Catalogo de productos

- Grid escalonado con `flutter_staggered_grid_view` para una presentación visual atractiva.
- Carga de imágenes cacheadas con `cached_network_image` y placeholders shimmer.
- Animaciones Hero entre la tarjeta del producto y la página de detalle.
- Búsqueda por texto con debounce de 300ms y soporte para deep links (`/buscar?q=`).
- Filtros combinables: categoría, rango de tallas, rango de precios, solo ofertas, color.
- Ordenación: más recientes, precio ascendente/descendente, nombre A-Z/Z-A.
- Productos destacados como "Nuevo" si fueron creados en los últimos 7 días.
- Indicadores visuales de descuento con porcentaje y precio original tachado.

## 6.2 Carrito con persistencia local

- Almacenamiento en **Hive** con adaptador personalizado (@HiveType): el carrito sobrevive al cierre de la app y funciona sin conexion.
- Identificacion unica por combinacion de `productId + size + color`.
- Fusion automatica de cantidades al añadir un producto que ya esta en el carrito.
- Validacion de stock real antes de incrementar cantidad (consulta Supabase en tiempo real).
- Slide-over lateral accesible desde cualquier pantalla via el icono del carrito con badge de cantidad.
- Calculo automatico de subtotal, ahorros por descuento y total general.

## 6.3 Checkout y pagos con Stripe

- Proceso de checkout en 4 pasos con indicador de progreso visual.
- **Paso 1 — Datos de envio:** email, telefono, nombre, direccion completa con validacion.
- **Paso 2 — Metodo de envio:** seleccion entre metodos disponibles con precio y estimacion de entrega.
- **Paso 3 —Codigo de descuento:** validacion en tiempo real contra el backend (limites de uso, expiracion, uso unico por cliente).
- **Paso 4 — Resumen y pago:** desglose completo del pedido con redireccion a Stripe Checkout.
- Resolucion inteligente de variantes por color para enviar la informacion correcta a Stripe.
- Verificacion de pago post-redireccion con pantalla de exito y numero de pedido.
- Soporte para multiples metodos de pago en Stripe: tarjeta, Apple Pay, Google Pay, PayPal, Revolut Pay.

## 6.4 Cuenta del cliente

- Registro con creacion automatica de perfil en tabla `customers`, email de bienvenida y suscripcion opcional a newsletter con codigo promocional.
- Login con email/contraseña, Google OAuth y Apple OAuth con URLs de redireccion especificas por plataforma.
- Recuperacion de contraseña por email.
- Edicion de perfil: nombre, telefono, avatar.



- Historial de pedidos completo con filtro por estado.
- Favoritos sincronizados en la nube con toggle optimista.
- Descarga de facturas en PDF.

## 6.5 Ofertas y descuentos

- **Ofertas en tiempo real:** `StreamProvider` suscrito a la tabla `products` — los cambios del admin se reflejan instantaneamente.
- **Switch global de ofertas:** tabla `app_config` con flag `offers_enabled` monitorizado en tiempo real.
- Precios con descuento visibles en tarjetas y detalle del producto.
- Codigos de descuento aplicables en el checkout con validacion completa.
- Carrusel hero en la home con slides promocionales configurables por el admin.

## 6.6 Favoritos

- Almacenamiento en tabla `customer_favorites` de Supabase.
- Toggle optimista: la UI se actualiza inmediatamente, la sincronizacion con el servidor ocurre en segundo plano con rollback automatico ante errores.
- Cache global de IDs de productos favoritos para consultas reactivas eficientes.
- Pantalla dedicada con listado completo de productos favoritos.

---

## 7. Flujo de compra

---



1. El cliente navega por el catalogo, aplica filtros o busca un producto.
2. Accede al detalle del producto con galeria de imagenes, descripcion y seleccion de variante (talla y color).

- 3. Selecciona talla y color con validacion de stock en tiempo real. El selector muestra disponibilidad por variante.
- 4. Añade el producto al carrito. Si el mismo producto con la misma variante ya existe, se fusionan las cantidades.
- 5. Procede al checkout: introduce datos de envio, selecciona metodo de envio, aplica codigo de descuento opcional y revisa el resumen.
- 6. Se crea una sesion de Stripe Checkout y se redirige al navegador externo para completar el pago de forma segura.
- 7. Tras el pago, la app verifica la sesion de Stripe y muestra la pantalla de exito con el numero de pedido.
- 8. Se vacia el carrito automaticamente y el pedido queda registrado en el historial del cliente.

## 8. Pedidos, devoluciones y facturacion

### 8.1 Ciclo de vida del pedido



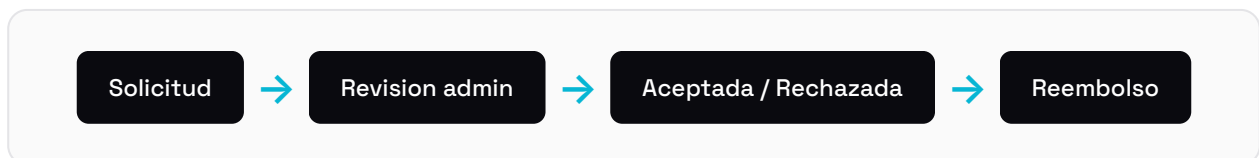
ESTADO	DESCRIPCION	ACCIONES DISPONIBLES
Pendiente	Pedido creado, esperando confirmacion de pago	Cancelar (cliente/admin)
Pagado	Pago verificado via webhook de Stripe	Cancelar (cliente/admin), preparar envio (admin)
Enviado	Pedido enviado con numero de seguimiento	Consultar tracking (SEUR, Correos, MRW)
Entregado	Pedido recibido por el cliente	Solicitar devolucion (cliente)
Cancelado		Solo lectura — motivo visible

ESTADO	DESCRIPCION	ACCIONES DISPONIBLES
	Pedido cancelado con reembolso procesado	
Devolucion solicitada	Cliente ha iniciado una devolucion	Aceptar/Rechazar (admin)
Devuelto	Devolucion aceptada, reembolso procesado	Solo lectura
Devolucion parcial	Parte de los articulos devueltos	Solicitar nueva devolucion (cliente)
Devolucion rechazada	Admin ha denegado la devolucion	Solo lectura

## 8.2 Cancelaciones

- El cliente puede cancelar pedidos en estado **pendiente** o **pagado**, indicando un motivo.
- El administrador puede cancelar cualquier pedido no entregado. El motivo "Cancelado por el administrador" se registra automaticamente.
- La cancelacion procesa el reembolso via Stripe, repone el stock y envia notificaciones por email a ambas partes.
- Los pedidos cancelados quedan bloqueados: el admin no puede modificar su estado ni datos de envio. El motivo de cancelacion se muestra en el detalle.

## 8.3 Devoluciones



- El cliente inicia la devolucion desde el detalle del pedido, indicando el motivo.
- El administrador revisa la solicitud y puede aceptarla (total o parcial) o rechazarla.
- Si se acepta, se procesa el reembolso via Stripe, se repone el stock y se genera una factura rectificativa en PDF.

- Las devoluciones parciales permiten devolver solo algunos articulos del pedido, manteniendo el estado `partial_return` para futuras devoluciones de los articulos restantes.
- El motivo de la devolucion se persiste en la base de datos y es visible tanto para el cliente como para el administrador.

## 8.4 Facturacion

- **Factura de venta:** generada automaticamente al confirmar el pedido, descargable y enviable por email.
  - **Factura rectificativa:** generada automaticamente ante cancelacion o devolucion aceptada, con desglose de importes reembolsados.
  - Generacion de PDF directamente en el Edge Function con `pdf-lib`, enviados como adjunto por email via Resend.
- 

# 9. Panel de administracion

---

El panel de administracion esta integrado en la propia app, protegido por un sistema de autenticacion independiente basado en una tabla `admins` con verificacion de credenciales via RPC. Consta de **13 pantallas** con navegacion lateral mediante drawer.

## 9.1 Dashboard

- KPIs del negocio: total de productos, total de clientes, pedidos pendientes, ventas del mes, productos con stock bajo, producto mas vendido.
- Listado rapido de los ultimos pedidos y productos registrados.
- Acceso directo a todas las secciones del panel desde tarjetas interactivas.

## 9.2 Graficos de ventas (KPIs)

- Graficos interactivos con `fl_chart` mostrando la evolucion diaria de pedidos e ingresos.
- Periodo configurable (ultimos N dias) con actualizacion en tiempo real.
- Visualizacion de tendencias de venta para toma de decisiones.

### 9.3 Gestion de productos

ACCION	DESCRIPCION
Crear producto	Nombre, slug, descripcion, precio, categoria, etiquetas y estado (activo / inactivo)
Variantes	Definicion de tallas y colores con stock independiente por variante, SKU y precio opcional
Imagenes	Subida multiple a Cloudinary con compresion automatica (max 1200px, calidad 80), gestion de colores asociados
Ofertas	Precio con descuento, porcentaje de ahorro, switch de oferta visible en catalogo y home
Filtro activo/ inactivo	Visualizacion separada de productos activos e inactivos con contadores

### 9.4 Gestion de pedidos

- Listado completo con filtros por estado.
- Vista de detalle con productos comprados, direccion, metodo de envio y datos de pago.
- Cambio de estado: preparar envio con numero de tracking, marcar como entregado.
- Cancelacion con reembolso Stripe automatico y registro del motivo.
- Gestion de devoluciones: aceptar (total o parcial), rechazar, con reposicion de stock y factura rectificativa.
- Envio automatico de emails al cliente en cada cambio de estado (envio, entrega, cancelacion, devolucion).
- Pedidos cancelados bloqueados: panel de solo lectura con motivo visible.

### 9.5 Configuracion de la tienda

- **Codigos de descuento:** creacion con porcentaje o cantidad fija, compra minima, limites de uso, uso unico por cliente, fechas de expiracion, activacion/desactivacion.
- **Carrusel hero:** CRUD de slides con titulo, subtítulo, imagen, enlace CTA, duracion de visualizacion y orden.

- **Categorías:** CRUD completo con nombre, slug, descripción, imagen y orden de visualización.
  - **Newsletter:** envío de campañas masivas con contenido personalizado, imagen de encabezado, código promocional y botón de acción.
  - **Notificaciones:** centro con pedidos nuevos (24h), alertas de stock bajo/agotado y pedidos pendientes.
  - **Usuarios:** listado de clientes con estadísticas de pedidos realizados y gasto total acumulado.
  - **Facturas:** gestión y envío de facturas a clientes.
  - **Animaciones:** configuración de animaciones del frontend.
- 

## 10. Seguridad y confianza

---

MECANISMO	IMPLEMENTACION
Autenticación de clientes	Supabase Auth con email/contraseña, Google OAuth y Apple OAuth, con confirmación por email y recuperación de contraseña
Autenticación de admin	Sistema independiente con tabla <code>admins</code> y verificación de credenciales vía función RPC — aislado de Supabase Auth
Autorización	Row Level Security (RLS) en todas las tablas: cada cliente solo accede a sus propios datos
Funciones RPC admin	Funciones PostgreSQL con <code>SECURITY DEFINER</code> que operan como superusuario para bypass controlado de RLS en operaciones administrativas
Pagos seguros	Stripe Checkout: los datos de la tarjeta nunca pasan por la app ni por nuestro servidor
Verificación de pagos	Webhook con firma HMAC — los pedidos solo se confirman tras validación criptográfica en el servidor
Edge Function relay	Las peticiones Web pasan por un Supabase Edge Function que actúa como proxy CORS, evitando exponer el backend directamente al navegador

---

MECANISMO	IMPLEMENTACION
Variables de entorno	Claves sensibles almacenadas exclusivamente en variables de entorno del servidor y Edge Functions
Manejo de errores	Jerarquia de excepciones tipadas (AppException) con Either<Failure, T> de fpdart para propagacion funcional de errores

**Principio de minimo privilegio:** la app utiliza exclusivamente la clave anonima de Supabase. Las operaciones administrativas se ejecutan a traves de funciones RPC con SECURITY DEFINER y el Edge Function relay, nunca con la clave de servicio en el cliente.

## 11. Arquitectura y tecnologias

### Flutter 3 + Dart 3.8

Framework multiplataforma para iOS, Android y Web con compilacion nativa y hot reload para desarrollo agil

### Riverpod

Gestion de estado reactiva con generacion de codigo, StreamProvider para tiempo real y FutureProvider para datos asincronos

### Freezed

Modelos de datos inmutables con union types, serializacion JSON automatica y metodo copyWith generado

### GoRouter

Navegacion declarativa con StatefulShellRoute para pestañas, deep links, rutas tipadas y soporte para redirecciones

### Supabase

Base de datos PostgreSQL, autenticacion, Realtime subscriptions, Edge Functions y Row Level Security como backend completo

### Stripe

Pasarela de pagos con Checkout Sessions, redireccion a navegador externo, verificacion post-pago y reembolsos automaticos

### Hive

Base de datos local NoSQL para persistencia del carrito offline con adaptadores personalizados y acceso ultrarapido

### Cloudinary

Almacenamiento y optimizacion de imagenes con upload sin firma, compresion automatica y transformaciones por URL

### fpdart

Programacion funcional con Either<Failure, T> para manejo robusto de errores sin excepciones no controladas

### fl\_chart

Graficos interactivos y personalizables para el dashboard de ventas con visualizacion de tendencias diarias

## Patron arquitectonico

La aplicacion sigue el patron **Clean Architecture por Features** (Feature-first). Cada funcionalidad se organiza en su propia carpeta dentro de `lib/features/` con tres capas independientes:

CAPA	RESPONSABILIDAD	CONTENIDO
Data	Acceso a datos y serializacion	Modelos Freezed, datasources remotos, repositorios de implementacion
Domain	Logica de negocio pura	Repositorios abstractos, entidades, casos de uso
Presentation	Interfaz de usuario y estado	Screens, widgets, providers Riverpod

## Estructura de carpetas

CARPETA	DESCRIPCION
<code>lib/config/</code>	Configuracion global: tema, router, constantes, almacenamiento Hive
<code>lib/shared/</code>	



CARPETA	DESCRIPCION
	Codigo compartido: excepciones, extensiones, servicios, widgets reutilizables
lib/features/admin/	Panel de administracion (13 pantallas, dashboard, CRUD, notificaciones)
lib/features/auth/	Autenticacion (login, registro, recuperacion, OAuth)
lib/features/cart/	Carrito con persistencia Hive
lib/features/ checkout/	Proceso de checkout y pagos Stripe
lib/features/ favorites/	Lista de favoritos sincronizada
lib/features/home/	Pantalla de inicio con carrusel y ofertas en tiempo real
lib/features/orders/	Gestion de pedidos del cliente
lib/features/ products/	Catalogo, detalle, busqueda y filtrado de productos
lib/features/ profile/	Perfil y datos personales del usuario
lib/features/splash/	Pantalla de carga con logo animado

## 12. Integraciones externas

SERVICIO	FUNCION	COMUNICACION
Supabase	Base de datos, autenticacion, Realtime y politicas de acceso RLS	SDK oficial de Flutter (supabase_flutter)

SERVICIO	FUNCION	COMUNICACION
Stripe	Cobros online y reembolsos automaticos via Checkout Sessions	API REST a traves del backend FashionStore + Edge Function relay
Cloudinary	Almacenamiento, compresion y entrega optimizada de imagenes	Upload HTTP sin firma + URLs de transformacion automatica
FashionStore API	Backend Astro que gestiona checkout, emails, stock, facturas y descuentos	HTTP client directo (movil) o Edge Function relay (Web/CORS)
Resend	Envio de emails transaccionales (confirmaciones, envio, devolucion, newsletter)	API REST invocada desde el backend FashionStore y Edge Functions

La app movil se comunica directamente con el backend FashionStore via HTTP. En Web, las peticiones pasan por un **Supabase Edge Function** (orders-relay) que actua como proxy CORS para evitar restricciones del navegador.

## 13. Modelos de datos

Todos los modelos principales utilizan **Freezed** para inmutabilidad, serializacion JSON automatica y metodo copyWith. El carrito utiliza ademas **Hive** para persistencia local.

### 13.1 Modelos del dominio

MODELO	CAMPOS PRINCIPALES	PROPIEDADES COMPUTADAS
ProductModel	id, name, slug, description, price, stock, categoryId, isOffer, originalPrice, discountPercent, sizes, active, images, tags	mainImage, hasDiscount, formattedPrice, availableSizes, colors, isNew, isSoldOut, canBePurchased

MODELO	CAMPOS PRINCIPALES	PROPIEDADES COMPUTADAS
ProductImageModel	id, productId, imageUrl, sortOrder, color, colorHex, altText	thumbnailUrl (300×400), detailUrl (800×1000)
ProductVariantModel	id, productId, size, stock, sku, price, isOffer	isOutOfStock, isLowStock (3 o menos)
CategoryModel	id, name, slug, description, imageUrl, parentId, sortOrder, active	—
OrderModel	id, orderNumber, totalPrice, status, customerId, customerEmail, shippingAddress, trackingNumber, cancellationReason, items	formattedOrderNumber, canCancel, canRequestReturn, trackingUrl
CartItemModel	id, productId, name, slug, price, quantity, size, imageUrl, color, originalPrice, discountPercent	uniqueId (productId_size_color), subtotal, savings, hasDiscount
CustomerModel	id, email, fullName, phone, avatarUrl, defaultAddressId, newsletter	displayName, initials, isProfileComplete
CarouselSlideModel	id, title, subtitle, imageUrl, ctaText, ctaLink, duration, sortOrder, isActive, discountCode	—

## 13.2 Modelos de soporte

MODELO	DESCRIPCION
AdminModel	id, email, fullName, role — con isSuperAdmin y displayName

MODELO	DESCRIPCION
AdminNotification / NotificationSummary	Tipos: newOrder, lowStock, outOfStock, pendingOrders — con contadores y totalUnread
ShippingMethodModel	id, name, description, price, estimatedDaysMin/Max — con formattedPrice y estimatedDelivery
OrderItemModel	id, orderId, productId, productName, productImage, quantity, size, color, priceAtPurchase
CustomerAddressModel	street, city, postalCode, province, country — para envio y facturacion
ProductFilters	search, categorySlug, sizes, priceMin/Max, offersOnly, color, sortBy

## 14. Navegacion y rutas

La navegacion utiliza **GoRouter** con `StatefulShellRoute` para mantener el estado de las pestañas principales. La app soporta deep links tanto en Web como en nativo.

### 14.1 Navegacion principal (pestañas)

PESTAÑA	RUTA	ICONO	PANTALLA
Inicio	/home	Icons.home	HomeScreen — carrusel, categorias, productos destacados, ofertas
Productos	/productos	Icons.grid_view	ProductListPage — catalogo con filtros y busqueda
Carrito	/carrito	Icons.shopping_bag + badge	CartScreen — productos añadidos con totales

PESTAÑA	RUTA	ICONO	PANTALLA
Cuenta	/cuenta	Icons.person	ProfileScreen — perfil, pedidos, favoritos

## 14.2 Rutas de cliente

RUTA	PANTALLA
/productos/:slug	Detalle de producto con galeria, variantes y carrito
/categoria/:slug	Listado filtrado por categoria
/ofertas	Listado filtrado solo ofertas
/buscar?q=texto	Busqueda por texto con deep link
/cuenta/pedidos	Historial de pedidos del cliente
/cuenta/pedidos/:orderId	Detalle de pedido con timeline de estados
/cuenta/favoritos	Productos favoritos
/checkout	Proceso de compra (4 pasos)
/checkout/success	Confirmacion de compra exitosa

## 14.3 Rutas de autenticacion

RUTA	PANTALLA
/auth/login	Inicio de sesion (email, Google, Apple)
/auth/registro	Registro de nuevo cliente
/auth/recuperar	Recuperacion de contraseña

## 14.4 Rutas de administracion

RUTA	PANTALLA
/admin/dashboard	Dashboard con KPIs y resumen
/admin/kpis	Graficos de ventas interactivos
/admin/productos	Gestion CRUD de productos
/admin/pedidos	Gestion de pedidos y devoluciones
/admin/codigos	Codigos de descuento
/admin/carrusel	Configuracion del carrusel hero
/admin/categorias	Gestion de categorias
/admin/usuarios	Listado de clientes
/admin/notificaciones	Centro de notificaciones
/admin/newsletter	Envio de campañas
/admin/facturas	Gestion de facturas
/admin/animaciones	Configuracion de animaciones

## 15. Estado actual y operacion

ASPECTO	ESTADO
App cliente (iOS / Android)	Operativa — todas las funcionalidades implementadas y probadas
App cliente (Web)	Operativa — con Edge Function relay para CORS

ASPECTO	ESTADO
Panel de administracion	Operativo — 13 pantallas con gestion completa
Pasarela de pago	Integrada y funcional con Stripe Checkout
Persistencia local	Carrito offline con Hive, preferencias con SharedPreferences
Tiempo real	Activo — ofertas y configuracion se actualizan instantaneamente via Supabase Realtime
Backend compartido	Conectado al mismo backend FashionStore (Astro) y base de datos Supabase que la web
Base de datos	18+ tablas con politicas RLS, funciones RPC, triggers y Realtime habilitado

## Servicios compartidos implementados

SERVICIO	ARCHIVO	FUNCION
<b>SupabaseService</b>	shared/services/ supabase_service.dart	Singleton de Supabase: auth, database, realtime, storage
<b>FashionStoreApiService</b>	shared/services/ fashion_store_api_service.dart	Cliente HTTP (637 lineas) para todo el backend: checkout, stock, descuentos, emails, pedidos, facturas
<b>CloudinaryService</b>	shared/services/ cloudinary_service.dart	Upload sin firma, compresion (1200px, calidad 80), carpetas por tipo (products, carousel, categories)
<b>StripeService</b>	checkout/presentation/services/ stripe_service.dart	Resolucion de variantes, creacion de sesion Stripe,

SERVICIO	ARCHIVO	FUNCION
		apertura en navegador externo, verificacion
LocalStorageService	shared/services/local_storage_service.dart	Wrapper de SharedPreferences con soporte para tipos primitivos, listas y JSON

## Edge Function: orders-relay

Funcion de Supabase Edge Functions (703 lineas TypeScript) que actua como proxy CORS entre Flutter Web y el backend FashionStore. Soporta 17 acciones diferentes incluyendo checkout, verificacion de pago, gestion de descuentos, emails, devoluciones y cancelaciones. Incluye generacion directa de PDFs de facturas rectificativas con pdf-lib.

## 16. Impacto y beneficios

BENEFICIO	DESCRIPCION
Experiencia nativa	Rendimiento nativo en iOS y Android con animaciones fluidas, gestos naturales y carga instantanea
Acceso offline	El carrito funciona sin conexion gracias a Hive — el usuario nunca pierde sus selecciones
Administracion movil	El gestor de la tienda controla pedidos, stock, devoluciones y configuracion desde cualquier lugar
Actualización instantanea	Los cambios en ofertas y configuracion se reflejan en tiempo real en todos los dispositivos conectados
Ecosistema unificado	Comparte backend, base de datos y logica de negocio con la tienda web, eliminando duplicacion



BENEFICIO	DESCRIPCION
Escalabilidad	Arquitectura Clean Architecture por Features preparada para crecer en funcionalidades sin rediseñar
Mantenibilidad	Modelos inmutables (Freezed), tipado fuerte (Dart), estado reactivo (Riverpod) y errores funcionales (fpdart) reducen bugs
Multiplataforma	Un unico codebase para iOS, Android y Web — reduccion significativa de tiempo y coste de desarrollo

## 17. Mejoras futuras

MEJORA	DESCRIPCION
Notificaciones push	Firebase Cloud Messaging para alertas de estado de pedido, ofertas y recordatorios de carrito abandonado
Chat en vivo	Atencion al cliente en tiempo real integrada en la app con soporte de Supabase Realtime
Programa de fidelizacion	Puntos por compra, descuentos acumulables y niveles de cliente VIP
Internacionalizacion	Soporte multidioma (i18n) y multidivisa para expansion internacional
Opiniones de clientes	Sistema de valoraciones y reseñas verificadas en los productos con fotos
Modo oscuro / claro	Implementacion completa del tema claro (actualmente dark-only) con switch de preferencia del usuario
Biometria	Login con huella dactilar y Face ID para acceso rapido y seguro
Widget de inicio	Widget nativo para iOS y Android mostrando ofertas destacadas en la pantalla de inicio del dispositivo

---

**FashionMarket Flutter** — Documentacion de entrega del proyecto

Jose Manuel Ortega Soto · Sistema de Gestion Empresarial · Victoriafp · 2026

[github.com/JoseManuelOS/FashionMarketFlutter](https://github.com/JoseManuelOS/FashionMarketFlutter)