# DOCUMENTO MAESTRO – MarketSales (Versión Actualizada 2025)

## 1. INTRODUCCIÓN Y CONTEXTO DEL PROYECTO

1.1 Origen del Proyecto  
MarketSales es la evolución de la aplicación 'Caja Mercadillos', creada para dar respuesta a una necesidad real detectada en vendedores y artesanos que trabajan en mercadillos. El objetivo principal es proporcionar una herramienta de control de caja adaptada al contexto real de este tipo de negocio, con especial énfasis en su uso sin conexión y la posterior sincronización cuando haya internet.  
  
1.2 Filosofía de Desarrollo  
La filosofía de desarrollo se basa en resolver problemas reales con soluciones prácticas, priorizando la fiabilidad y la usabilidad sobre la complejidad innecesaria. Se ha diseñado con una visión global para evitar el problema de 'arreglar una cosa y estropear otra'.  
  
1.3 Objetivos Principales  
• Proporcionar un sistema de control de caja completo y fiable para mercadillos.  
• Garantizar un funcionamiento óptimo tanto online como offline.  
• Ofrecer un modelo escalable con versión gratuita y versión premium.  
• Adaptar la app al flujo real de trabajo en mercadillos.

## 2. MODELO DE NEGOCIO Y ESTRATEGIA COMERCIAL

2.1 Versiones  
• Versión FREE: Sistema monopuesto (un único dispositivo autorizado a la vez), funciones básicas completas, sincronización básica, control de caja, gestión de artículos y categorías.  
• Versión PREMIUM: Multipuesto, control de stock avanzado, control de costes, reportes, backup/exportación, fotos de artículos, códigos de barras y gestión multiusuario.  
  
2.2 Autenticación  
• Autenticación con Google y correo/contraseña usando Firebase Authentication.  
• Control monopuesto en la versión FREE similar al funcionamiento de WhatsApp.

## 3. ARQUITECTURA TÉCNICA Y GESTIÓN DE DATOS

• Arquitectura offline-first con Room como base de datos local y Firebase Firestore como almacenamiento en la nube.  
• Patrón MVVM, Repository Pattern y Clean Architecture.  
• Sincronización automática cuando hay conexión, con datos siempre disponibles offline.  
• Tecnologías: Kotlin, Jetpack Compose, Room, Firebase Firestore, Firebase Auth, Coroutines, Flow, Material Design 3.

## 4. ESTADO ACTUAL DEL DESARROLLO

4.1 Funcionalidades Completas  
• Autenticación y configuración inicial.  
• Gestión de categorías y artículos con sincronización híbrida.  
• Gestión de mercadillos y calendario con estados.  
• Tema, tipografía y colores personalizados.  
  
4.2 En Desarrollo  
• Flujo de ventas con pantalla intermedia de pago.  
• Envío de recibos por WhatsApp/SMS/Email.  
• Pestaña de productos Premium.  
• Sincronización de ventas con cola de pendientes.  
  
4.3 Pendiente  
• Listados y reportes.  
• Funciones Premium avanzadas.  
• Exportación/backup.  
• Integración con TPV físico (SumUp).

## 5. FLUJO DE VENTAS DEFINIDO

• Botón 'Realizar cargo' → Pantalla de selección de método de pago (Efectivo, Bizum, Tarjeta).  
• Efectivo: Pantalla de cálculo de cambio → Pantalla de envío de recibo.  
• Bizum/Tarjeta: Directo a pantalla de envío de recibo.  
• Recibo en texto plano con líneas de venta, total y método de pago.  
• Abonos: Desde listado de líneas, con confirmación, generando línea negativa.  
• Pestaña Productos (Premium): Grid con favoritos y resto alfabético, con filtro por categoría.  
• IVA incluido en precios introducidos por el usuario.

## 6. ROADMAP

Prioridad inmediata:  
1. Implementar flujo completo de pago + envío de recibo.  
2. Integrar abonos en UI.  
3. Activar pestaña de productos Premium.  
4. Mejorar sincronización de ventas.  
  
Futuras versiones:  
• Reportes y listados.  
• Control de costes y estadísticas avanzadas.  
• Modo multidispositivo Premium.  
• Exportación/backup.  
• Integración con TPV físico.