

Documentación App Android

Proyecto: Alfa Ediciones • Plataforma: Android (Kotlin, Jetpack Compose, Room) • Backend: Firebase

1. Resumen

Aplicación Android para gestión de clientes y ventas con sincronización en Firebase. Incluye cálculo de CUIL, consulta de estado de venta, carga de datos, generación de PDF y panel de administración.

2. Funcionalidades principales

- Menú: CUIL, Estado de venta, Cargar datos, PDF, accesos BCRA/ANSES, Administrador.
- Cargar datos: guarda ventas en Firestore en colecciones `Debito`, `tarjeta`, `mercadopago`.
- Estado de venta: búsqueda por DNI en tiempo real; muestra apartados, devolución y "No apto venta" si corresponde.
- Administrador: marca APTO, define apartados múltiples, escribe devolución y lista/gestiona ventas.
- PDF: genera un ejemplo local con iText (extensible para reportes).

3. Arquitectura (alto nivel)

- UI: Jetpack Compose (pantallas en `MainActivity.kt`).
- Datos locales: Room/SQLite (módulo de incobrables). Importación desde Excel (Apache POI).
- Nube: Firebase (Firestore para ventas y config; Storage para Excel incobrables).

4. Modelo de datos (Firestore)

- Colecciones: `Debito`, `tarjeta`, `mercadopago`
- Campos comunes: `dni`, `nombreApellido`, `numeroEquipo`, `cargo`, `provincia`, `celular`, `vendedorNombre`, `vendedorCodigo`, `importeTotal`, `valorCuota`, `fechaCreacion`
- Por método: CBU → `cbu` • Tarjeta → `numeroTarjeta`, `fechaVencimiento`, `codigoTarjeta` • MP → `fechaVenta`
- Admin: `apto` (bool), `apartados` (array), `devolucion` (string)
- Config: `config/admin` (password), `config/app_state` (locked)

5. Flujos

5.1 Cargar datos (ventas)

1. Usuario completa formulario (método de pago y datos).
2. La app crea documento en colección según método.
3. Admin gestiona APTO/apartados/devolución.

5.2 Estado de venta

1. Busca por DNI en las 3 colecciones.
2. Muestra apartados seleccionados y devolución; si `apto=false`, indica "No apto venta".

5.3 Administrador

- Lista ventas combinadas (Debito/tarjeta/mercadopago).
- Setea `apto`, `apartados` y `devolucion` (merge en documento).
- Acciones: resetear datos de clientes (solo colecciones de ventas); actualizar incobrables locales desde Storage.

6. Dependencias clave

Android

- Compose BOM, Material3, Activity Compose
- Room (runtime, ktx, compiler)
- Firebase (Firestore, Auth, Storage, Analytics)
- Apache POI (Excel), iText (PDF), Coroutines

Versión mínima

- minSdk 26 • targetSdk 34
- JDK 17 (Gradle JDK)
- Compose Compiler ext. 1.5.x

7. Seguridad (sugerida)

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{col}/{id} {
      allow read: if col in ['Debito','tarjeta','mercadopago'];
      allow write: if request.auth != null && request.auth.token.admin == true;
    }
    match /config/{doc} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.token.admin == true;
    }
  }
}
```

8. Build y despliegue

- Android Studio: Build > Make Project • Run en dispositivo.

- Incobrables: actualizar desde Admin (descarga Excel desde Storage y repuebla SQLite).

9. Solución de problemas

- JAVA_HOME: usar JDK 17 (Embedded JDK).
- Daemons Kotlin/Gradle: limpiar `.gradle` , `.kotlin` , `build/` y recompilar.
- Permisos Android: INTERNET y ubicación (para widgets o funciones que la requieran).

© Alfa Ediciones