

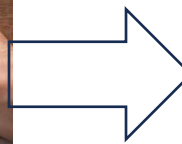
Desarrollo_de_aplicaciones_web_y_moviles

Septimo Semestre.

Software Punto de Venta (POS)

Muchas pequeñas tiendas de abarrotes gestionan ventas, control de stock y cuentas con **libretas y apuntes manuales**. Esto conlleva errores humanos, pérdidas por faltantes, demora en detectar productos con baja rotación, y dificultad para generar reportes o tomar decisiones rápidas (reordenar, ofertas, control de proveedores). Los dueños necesitan una solución económica, simple y móvil para:

- Registrar ventas rápidas con la cámara (escaneo de códigos).
- Actualizar inventario automáticamente.
- Generar reportes diarios/semanales.
- Llevar control de proveedores y precios.
- Operar offline y sincronizar cuando haya red.
- Operar con productos que se venden a agranel o de forma individual
- Alertas de productos próximos a vencer para reducir pérdidas.
- Programación de ofertas y descuentos por temporada o festividad.



2) Objetivo general

Construir una app móvil multiplataforma (Android/iOS) con React Native que funcione como Punto de Venta (POS) para abarrotes, integrada con una API y base de datos en la nube, que permita registrar ventas mediante escáner de códigos, controlar inventarios, autenticación de usuarios, reportes y administración de productos/proveedores.



3) Objetivos específicos

- Implementar CRUD de productos e inventario.
- Registrar ventas y descontar stock automáticamente.
- Integrar lector de códigos de barras con la cámara del celular.
- Crear API REST con Node.js/Express y base de datos en la nube (Postgres/Supabase).
- Implementar autenticación (JWT o Firebase) y buenas prácticas de seguridad.
- Soportar persistencia local y sincronización básica (offline-first opcional).
- Entregar app instalable y documentación técnica / usuario.

4) Personal Interesado

- Dueño/empleador de la tienda (usuario final): realiza ventas, revisa inventario, genera reportes.
- Administrador (owner): configura productos, proveedores y usuarios.

5) Historias de usuario (ejemplos)

1. Como cajero, quiero escanear el código con la cámara para agregar el producto a la venta, así ahorro tiempo y evito errores.
2. Como dueño, quiero ver el inventario en tiempo real para saber qué reordenar.
3. Como administrador, quiero crear/editar productos y proveedores desde la app.
4. Como encargado, quiero registrar una venta con efectivo o tarjeta (marcar forma de pago) y guardar el comprobante.
5. Como gerente, quiero generar un reporte de ventas del día para ver el total y las ventas por producto.
6. Como usuario legítimo, quiero iniciar sesión para que los datos estén protegidos.

6) Requerimientos funcionales (resumen)

- Autenticación (login/logout, roles básicos).
- CRUD productos: nombre, código (EAN/UPC), precio compra/venta, stock, proveedor, categoría, imagen (opcional).
- CRUD proveedores.
- Listado inventario con búsqueda/filtrado.
- Flujo de venta: crear venta, agregar productos (por escaneo o búsqueda), indicar cantidad, calcular subtotal/IVA/total, registrar pago, generar comprobante.
- Descontar stock automáticamente.
- Reportes: ventas por día, top productos, stock bajo.
- Escáner de códigos con cámara (expo-barcode-scanner / MLKit).
- Persistencia local (AsyncStorage/SQLite) para datos temporales y sincronización.
- Manejo de errores y mensajes al usuario.

7) Requerimientos no funcionales

- Multiplataforma (Android/iOS) con React Native
- Backend RESTful con Node.js + Express.
- Base de datos PostgreSQL en la nube (Supabase/Railway).
- Seguridad: contraseñas hasheadas, tokens JWT, validación de entrada.
- Rendimiento aceptable en dispositivos de gama baja.
- Documentación y README claros para reproducir instalación.

8) Arquitectura (cliente-servidor)

- **Cliente (React Native):** UI, navegación, cámara, validaciones, consumo de API, persistencia local.
- **Servidor (Node.js/Express):** API REST, lógica de negocio, validaciones, autenticación, conexión a BD.
- **Base de datos (Postgres):** tablas:
- **Deploy:** backend en Railway/Heroku, BD en Supabase; frontend entregado como APK/AAB e instalado mediante Expo/EAS.