

Guia Sistema Multisucursal Offline/Online

1. Objetivo

Desarrollar un sistema de gestión de inventario, ventas y facturación multisucursal que funcione offline y se sincronice en la nube, con soporte para varios dueños.

2. Tecnologías Recomendadas

- Frontend: React + Next.js (PWA), Material UI, Service Workers para offline.
- Backend: Java Spring Boot, Spring Security + JWT, API REST/GraphQL.
- Base de datos central: PostgreSQL + CouchDB para sincronización.
- Base local: IndexedDB (navegador) o SQLite (PC/móvil).
- Sincronización: PouchDB <-> CouchDB para replicación offline/online.
- Infraestructura: AWS/DigitalOcean, SSL, backups automáticos.

3. Arquitectura

- Cliente (React PWA) -> Backend Spring Boot -> BD Central (PostgreSQL + CouchDB).
- Cada sucursal trabaja localmente si no hay internet, luego sincroniza datos cuando se reconecta.

4. Módulos Iniciales

1. Usuarios y Roles
2. Productos e Inventario por sucursal
3. Ventas y Facturación
4. Compras y Proveedores
5. Traslados entre sucursales
6. Sincronización offline/online
7. Reportes

5. Flujo Offline -> Online

1. Usuario vende sin internet -> datos guardados localmente.
2. Reconexión detectada -> sincroniza datos con la nube.
3. Backend actualiza inventario global y resuelve conflictos.

4. Dueño ve dashboard consolidado en tiempo real.

6. Multitenancy (multi-dueño)

- Cada dueño es un tenant en la BD central.
- Cada tenant tiene varias sucursales.
- Backend filtra datos según usuario y dueño autenticado.

7. Roadmap de Desarrollo

Fase 1 (MVP):

- Spring Boot Backend, React PWA, Inventario + Ventas offline.

Fase 2:

- Sincronización PouchDB <-> CouchDB, multi-sucursal.

Fase 3:

- Facturación SAT, Compras, Proveedores.

Fase 4:

- Multi-dueño (SaaS), App móvil nativa, reportes avanzados.

8. Ventajas de Spring Boot

- Robusto y escalable para lógica compleja.
- Mejor manejo de transacciones e integridad.
- Seguridad avanzada con Spring Security.

9. Próximos pasos

1. Configurar Spring Boot + PostgreSQL.
2. Crear API de productos, ventas e inventario.
3. Implementar JWT Auth para roles/usuarios.
4. Crear React PWA que consuma la API.
5. Añadir IndexedDB y PouchDB para offline.
6. Configurar CouchDB para replicación.