

Proyecto: Sistema Inteligente de Gestión de Pedidos y Ruteo Logístico

Arquitectura Fullstack

- **Backend:**
 - *Lenguaje:* Node.js + TypeScript
 - *API:* REST con Swagger/OpenAPI para integración transparente.
 - *Infraestructura:* Supabase/GCP (escalable, con auto-backups).
 - *Base de datos:* PostgreSQL con PostGIS para geodatos.
- **Frontend:**
 - *Web:* React + TypeScript, Zustand para estado, Tailwind CSS.
 - *Móvil:* React Native (iOS/Android) para repartidores.
 - *Mapas:* Google Maps SDK + Mapbox para visualización.

Módulos Detallados (Receta Paso a Paso)

Módulo 0: Núcleo del Sistema (*Nuevo*)

Objetivo: Cimientos para integraciones y seguridad.

Implementación:

1. Configurar autenticación JWT + OAuth2.0 (Supabase Auth).
2. Diseñar base de datos centralizada:
 - Tablas: pedidos, repartidores, almacenes, rutas, inventario.
3. Crear API Gateway (manejo centralizado de solicitudes).

Entregable: Backend base operativo con roles de usuario (admin, repartidor, gerente).

Módulo 1: Gestión Inteligente de Pedidos (*Corregido*)

Objetivo: Recepción automática y procesamiento de pedidos.

Implementación:

1. Desarrollar webhook para ERP (ej: SAP, Oracle) usando Node.js.

2. Validar datos:
 - Schema JSON con dirección, productos, prioridad.
3. Crear cola de procesamiento (RabbitMQ) para evitar pérdidas.
Entregable: Endpoint /api/orders con logs de auditoría.

Módulo 2: Inventario en Tiempo Real (Completado)

Objetivo: Consulta unificada de stock en múltiples almacenes.

Implementación:

1. Conectar APIs de inventario existentes (GraphQL/REST).
2. Implementar caché Redis (reducir latencia en consultas recurrentes).
3. Alertas WebSocket para stock crítico.
Entregable: Dashboard de inventario con sincronización ≤ 3 seg.

Módulo 3: Geocodificación Avanzada (Optimizado)

Objetivo: Convertir direcciones a coordenadas con precisión.

Implementación:

1. Integrar Google Geocoding API + OpenStreetMap (redundancia).
2. Usar Elasticsearch para normalizar direcciones ("Av." vs "Avenida").
3. Limitar costos: Cachear resultados 24h en PostgreSQL.
Entregable: Servicio /api/geocode con 99.9% de precisión.

Módulo 4: Motor de Ruteo Inteligente (Reestructurado)

Objetivo: Rutas óptimas recolección+entrega.

Implementación:

1. Algoritmo híbrido:
 - *Google Directions API* para tráfico en tiempo real.
 - *Algoritmo genético* para optimizar múltiples paradas.
2. Inputs:

- Ubicación almacenes (Módulo 2).
- Geolocalización repartidores (Módulo 6).

3. Variables:

- capacidad_vehículo, ventana_tiempo, restricciones_tráfico.

Módulo 5: Asignación Dinámica de Repartidores (*Nuevo*)

Objetivo: Asignar repartidor óptimo en segundos.

Implementación:

1. Modelo de prioridad:

typescript

Copy

Download

$prioridad = (0.4 * proximidad) + (0.3 * disponibilidad) + (0.3 * capacidad)$

2. Consumir ubicaciones GPS en tiempo real (WebSockets).
3. Integrar con Módulo 4 para asignar ruta automáticamente.

Entregable: API /api/assign-driver con respuesta ≤ 5 seg.

Módulo 6: App Móvil para Repartidores (*Completado*)

Objetivo: Gestión en tiempo real de entregas.

Funcionalidades clave:

- Navegación integrada (Google Maps/Apple Maps).
- Captura digital:
 - Fotos con geotag.
 - Firmas electrónicas (librería react-native-signature-capture).
- Modo offline (sincronización automática al recuperar conexión).

Entregable: App instalable en iOS/Android.

Módulo 7: Dashboards Gerenciales (*Ampliado*)

Objetivo: Visualización de KPIs estratégicos.

Implementación:

1. Librerías:
 - Gráficos: Recharts + D3.js.
 - Mapas en tiempo real: Mapbox GL JS.
 2. KPIs obligatorios:
 - Entregas exitosas/hora.
 - % rutas optimizadas.
 - Tiempo promedio recolección→entrega.
 - Calor de repartidores en mapa.
 3. Permisos: Personalizar vistas por rol (gerente vs operador).
- Entregable:** URL /dashboard.

Módulo 8: Cumplimiento Legal (*Nuevo*)

Objetivo: Garantizar tratamiento ético de datos.

Implementación:

1. Encriptar datos sensibles (AES-256).
2. Auditoría GDPR:
 - Borrar datos de clientes tras 6 meses.
 - Consentimiento explícito para geolocalización.
3. Integrar firma digital (eIDAS) para validación legal.

Entregable: Reporte de cumplimiento RGPD/LOPD.

Flujo de Implementación (Orden Crónico)

1. **Preparación:**
 - Configurar entornos (Dev, QA, Prod) en GCP.
 - Diseñar ERD de base de datos.

2. **Fase 1 (Backend):** Módulos 0 → 1 → 2 → 3 → 5 → 8.
3. **Fase 2 (Frontend):** Módulos 4 → 6 → 7.
4. **Pruebas:**
 - Test de carga (Locust): 1000 pedidos/minuto.
 - Validación mobile en dispositivos reales (BrowserStack).
5. **Despliegue:**
 - CI/CD con GitHub Actions.
 - Rollout progresivo (10% → 100% de usuarios).

Checklist Final

- Integración con ERP funcionando.
- App móvil captura firmas/fotos con GPS.
- Dashboard muestra rutas activas en tiempo real.
- Motor de ruteo reduce $\geq 15\%$ los costes logísticos.
- Auditoría legal aprobada.

Nota: Todos los módulos deben documentarse en Swagger y generar logs estructurados (JSON) para monitorización. Usar Prometheus + Grafana para métricas de rendimiento.