

ACTA CONSTITUTIVA / PROPUESTA DE PROYECTO

Nombre del Proyecto: Sistema Integral de Gestión para Restaurante con Módulo de Apoyo a la Toma de Decisiones (POS + SSD)

Responsable: Alejandro Lopez Cañedo

Duración estimada: 1 semestre (16 semanas)

Objetivo General:

Desarrollar un sistema web integral para la administración de un restaurante que incluya punto de venta (POS), control de mesas, gestión de menú y recetas, control de inventario, módulo de cocina/bar (KDS), pagos, reportes y un subsistema de apoyo a la toma de decisiones (SSD) con visualización de datos, análisis y predicciones.

Alcance:

- Operar las funciones diarias de un restaurante (órdenes, pagos, inventario).
- Controlar y visualizar el estado de mesas y órdenes en tiempo real.
- Gestionar inventario y compras con afectación automática por ventas.
- Producir reportes operacionales y analíticos.
- Apoyar la toma de decisiones con dashboards, minería de datos y modelos de predicción.

Tecnologías a utilizar:

- Frontend: Angular CLI (v17+), Angular Material, RxJS, Routing, Reactive Forms, HttpClient, Guards, Interceptors, Directivas y Pipes personalizados, Web Components con @angular/elements.
- Backend: NestJS, Prisma ORM, PostgreSQL, Socket.IO (POS↔KDS y estado de mesas).
- SSD: PostgreSQL (esquema estrella para DW), ETL con Node/NestJS, minería de datos básica (reglas de asociación, rotación ABC), pronóstico simple de demanda.
- Infraestructura: Docker Compose (Postgres/PGAdmin), despliegue de frontend en AWS S3/CloudFront y backend en VPS/Render.
- Calidad: ESLint, Prettier, Jest, Supertest.

Requisitos Funcionales:

- Autenticación y roles con JWT y control de acceso.
- Gestión de mesas y zonas con estado en tiempo real.
- CRUD de menú, categorías y recetas.
- POS para tomar órdenes con modificadores y envío a KDS.
- KDS con control de cola y estados.
- Gestión de pagos y división de cuentas.
- Inventario con movimientos y órdenes de compra.
- Reportes operacionales.
- Módulo SSD con DW, dashboards, minería de datos y pronóstico.

Requisitos No Funcionales:

- Usabilidad: interfaz responsiva y amigable.
- Rendimiento: respuesta de la API < 500 ms.
- Escalabilidad: arquitectura modular.
- Seguridad: JWT, validaciones y protección contra inyección SQL.
- Disponibilidad: despliegue en la nube.
- Mantenibilidad: ESLint/Prettier y tests.
- Compatibilidad con navegadores modernos y móviles.

Entregables:

- Código fuente en repositorio Git.
- Base de datos documentada (diagrama ER y esquema estrella).
- Manual de usuario.
- Manual técnico.
- Demo funcional del sistema.
- Reporte final con evidencias del SSD implementado.