

Levantamiento de requerimientos

1. Requerimientos Funcionales

Para Consumidores:

1. Acceso a menús digitales organizados por categorías (entradas, platos principales, bebidas, postres).
2. Visualización de precios actualizados y descripciones detalladas de los platillos.
3. Búsqueda de restaurantes por ubicación, tipo de cocina, y palabras clave.
4. Posibilidad de realizar pedidos:
 - Pedido en restaurante.
 - Pedido para recoger.
 - Pedido a domicilio.
5. Seguimiento del estado de los pedidos (por ejemplo: "Confirmado", "Preparándose", "En camino").
6. Valoración de restaurantes y platillos, incluyendo un sistema de estrellas y comentarios.
7. Registro y autenticación de usuarios mediante correo electrónico o redes sociales.
8. Historial de pedidos y favoritos para facilitar futuras elecciones.

Para Restaurantes:

1. Interfaz para gestionar menús:
 - Crear, editar y eliminar platillos.
 - Subir imágenes y actualizar precios.
2. Visualización en tiempo real de los pedidos realizados por los consumidores.
3. Gestión del estado de los pedidos (confirmar, preparar, entregar).
4. Estadísticas de ventas, incluyendo platillos más populares y horarios de mayor demanda.
5. Acceso a valoraciones y comentarios de los consumidores.
6. Administración de usuarios: asignación de roles y permisos a empleados (por ejemplo: administrador, cocinero, repartidor).
7. Gestión del perfil del restaurante (nombre, ubicación, horarios de atención, contacto).

2. Requerimientos No Funcionales

1. Base de Datos:

- Centralizada y fácil de manipular.
- Estructurada para almacenar usuarios, restaurantes, menús, pedidos y valoraciones.
- Compatibilidad con sistemas de gestión relacional (ej., MySQL o PostgreSQL).

2. Backend:

- Desarrollado con un framework robusto (por ejemplo, Node.js con Express o Django en Python).

- API RESTful para la comunicación entre frontend y backend.
- Manejo seguro de datos, incluyendo cifrado para contraseñas y conexiones HTTPS.

3. Frontend:

- **Consumidores:**
 - Diseño amigable y responsivo, adaptable a dispositivos móviles y navegadores.
 - Uso de tecnologías modernas como React.js o Angular.
- **Restaurantes:**
 - Interfaz intuitiva para la gestión operativa.
 - Tableros de control con gráficos interactivos para estadísticas.

4. Rendimiento:

- Tiempo de carga rápido (< 2 segundos para la mayoría de las operaciones).
- Escalabilidad para soportar un gran número de usuarios y transacciones simultáneas.

5. Seguridad:

- Autenticación y autorización robusta (por ejemplo, JWT o OAuth2).
- Protección contra ataques comunes como SQL Injection y XSS.

6. Soporte y Mantenimiento:

- Documentación clara del código y la API.
- Fácil de extender con nuevas funcionalidades en el futuro.