

## 1. Introducción

Este sistema está diseñado para optimizar la gestión de un restaurante mediante la digitalización de procesos clave. Permite a los meseros registrar pedidos de manera rápida y confiable, enviándoles automáticamente a cocina y caja, con la opción de modificar o unir cuentas de mesas. Los clientes pueden consultar la disponibilidad de mesas, pedidos y parqueadero, así como realizar reservas para mejorar su experiencia. Por su parte, el cajero puede consultar cuentas, aplicar descuentos o fraccionar pagos, evitando errores y agilizando la atención. Además, la aplicación móvil ofrece a los usuarios la posibilidad de registrarse, acceder a funciones personalizadas y guardar historial de consumos, mientras que el administrador obtiene información detallada sobre ventas y ganancias del día.

## 2. Arquitectura del Sistema

El sistema se construye bajo una arquitectura cliente-servidor con tres capas principales: frontend, backend y base de datos.

### Frontend (Interfaz Web)

- Desarrollado a partir de una plantilla de Bootstrap, adaptada con formularios y páginas personalizadas.
- Los usuarios interactúan a través de una interfaz amigable donde pueden realizar pedidos, consultar disponibilidad, reservas o acceder a su cuenta.
- La comunicación con el backend se realiza mediante formularios integrados o llamadas a una API REST.

### Backend (Python)

- Implementado con un framework web como Flask o Django.
- Contiene la lógica de negocio del sistema: gestión de pedidos, control de cuentas, aplicación de descuentos, administración de reservas, control de parqueadero y manejo de usuarios.
- Expone servicios a través de endpoints que permiten la comunicación entre la interfaz web y la base de datos.

### Base de Datos (BD)

Se plantea el uso de MySQL.

- Contiene tablas principales: Usuarios, Mesas, Pedidos, Cuentas, Reservas y Parqueadero, que centralizan la información del restaurante y permiten reportes y control administrativo.

## 3. Diseño de Casos de Uso (Derivados de las Historias de Usuario)

Actores: Mesero, Sistema, Cocina, Caja.

### Flujo principal:

Mesero ingresa el número de mesa y observaciones.

Sistema registra pedido y lo envía a cocina y caja.

Mesero recibe confirmación.

**Flujos alternativos:**

Modificación de pedido, unión de mesas, separación al cerrar cuenta.

#### **4. Modelo de Datos (Diseño de la Base de Datos)**

Ejemplo de tablas mínimas:

Usuario(id, nombre, rol, contraseña).

Mesa(id, número, estado, unida\_con).

Pedido(id, mesa\_id, detalle, estado, fecha).

Cuenta(id, mesa\_id, total, descuento, estado).

Reserva(id, cliente\_id, mesa\_id, fecha, estado).

Parqueadero(id, cupo\_total, cupo\_disponible).

#### **5. Diseño de Interfaz (Wireframes y Mockups)**

Pantalla de mesero: registro/modificación de pedidos.

Pantalla de cliente: consulta disponibilidad, reserva.

Pantalla de cajero: consulta cuenta, aplicar descuentos.

Pantalla de administrador: reporte de ventas, control de ganancias.

#### **6. Flujo de Datos**

Un diagrama de cómo viaja la información:

Cliente → Web → Backend → BD.

Mesero → Web → Python → BD → Cocina/Caja.

#### **7. Tecnologías y Herramientas**

Backend: Python (Flask/Django).

Base de datos: MySQL(producción).

Frontend: Bootstrap(con formularios incrustados o APIs).

Otros: GitHub (control de versiones), Draw.io (diagramas).