

**Nombre de la aplicación:**

**Curso Escolar:** 2024-2025

**Autor:** Dominic Arturo Obil Osorio

**Tutor:** Antonio Gabriel González Casado

**Fecha de Inicio:** 07/03/2025

**Fecha de Finalización:** 10/06/2025



# Índice

1. Introducción
2. Motivación del Proyecto
3. Descripción del Proyecto
4. Stack Tecnológico
5. Requisitos Funcionales
6. Modelo de datos



# 1. Introducción

Es una aplicación web diseñada para ayudar a los restaurantes en la gestión de reservas, optimización de mesas y mejora de la experiencia del cliente. Ofrecerá una interfaz intuitiva para que los clientes reserven mesas y permitirá a los restaurantes gestionar su aforo de manera eficiente.

## 2. Motivación del Proyecto

La idea de desarrollar esta aplicación surge directamente de mi experiencia personal trabajando como camarero, donde tuve que utilizar una aplicación para gestionar pedidos y reservas que era muy mala, con constantes fallos y poca usabilidad. Esta experiencia me permitió identificar claramente los problemas y las necesidades reales en el sector, especialmente en restaurantes pequeños y medianos.

## 3. Descripción del Proyecto

Permitirá a los clientes reservar mesas de manera online, evitando llamadas telefónicas y errores en la gestión de aforo. Los administradores del restaurante podrán gestionar reservas en tiempo real, establecer horarios disponibles, configurar límites de aforo y administrar cancelaciones. Además, se implementará un sistema de notificaciones para confirmar, recordar y cancelar reservas según sea necesario.

### Características principales:

- Reserva online de mesas por parte de los clientes.
- Gestión de disponibilidad y turnos por parte del restaurante.
- Gestión de cancelaciones y sobreocupación.
- Opción de lista de espera para mesas ocupadas.

## 4. Stack Tecnológico

- **Frontend:** Angular o React (SPA)
- **Backend:** Java + Spring Boot (API REST)
- **Base de datos:** MySQL / MariaDB
- **Seguridad:** JWT (Autenticación basada en tokens)
- **ORM:** JPA (Hibernate)
- **Otras herramientas:** Git, Docker, Postman

## 5. Lista de Requisitos Funcionales



Para clientes:

- RF01 - El sistema deberá permitir el registro de clientes con autenticación segura.
- RF02 - El sistema deberá permitir el inicio de sesión de clientes.
- RF03 - El sistema deberá permitir el cierre de sesión de clientes.
- RF04 - El sistema deberá permitir la búsqueda de disponibilidad de mesas según fecha y hora.
- RF05 - El sistema deberá permitir realizar reservas de mesas según la disponibilidad.
- RF06 - El sistema deberá permitir la cancelación de reservas realizadas.
- RF07 - El sistema deberá permitir la modificación de reservas realizadas.
- RF08 - El sistema deberá enviar notificaciones de confirmación de reserva.
- RF09 - El sistema deberá enviar notificaciones de recordatorio antes de la reserva.
- RF10 - El sistema deberá permitir a los clientes acceder a su historial de reservas.

Para restaurantes:

- RF11 - El sistema deberá permitir el registro de administradores de restaurantes.
- RF12 - El sistema deberá permitir el inicio de sesión de administradores.
- RF13 - El sistema deberá permitir el cierre de sesión de administradores.
- RF14 - El sistema deberá proporcionar un panel de control para la gestión de reservas.
- RF15 - El sistema deberá permitir la configuración de horarios y disponibilidad de mesas.
- RF16 - El sistema deberá permitir el control de aforo y turnos de mesas.
- RF17 - El sistema deberá permitir la gestión de listas de espera.

## 6. Modelo de datos

Se utilizan tablas relacionales normalizadas:

- users, roles, user\_roles
- Restaurante, Horario, Turno
- Reserva, Lista\_Espera, Notificacion

Cada entidad cuenta con relaciones claras entre cliente, reservas y restaurantes. Se aplican claves foráneas, timestamps y restricciones para mantener la integridad.