

# Electiva: Programación Web - Frontend

## Trabajo Práctico: Primer final.

Prof.: Ing. Gustavo Sosa Cataldo

Este trabajo involucra la implementación del frontend de un sistema informático utilizando tecnología Angular.

Entrega del TP:

- Lunes 17/11 (día del primer final).

### Grupos de alumnos

Los grupos serán de 5 alumnos como máximo, uno de los cuales estará identificado como el Líder del Proyecto (Project Leader), quien tendrá la obligación de tratar todos los temas concernientes a este trabajo con el cliente (profesor).

### Observaciones

- Para la implementación puede utilizar el IDE de su elección (Eclipse, Netbeans, IntelliJ IDEA, VSCode, etc).
- El frontend debe ser Angular
- Para el día de la entrega y defensa del TP deben estar presentes TODOS los integrantes del grupo, y el que no esté presente se llevará ausente.
  - a. Peso del trabajo: 50%
  - b. Peso del final escrito: 50%
- No se utilizará ninguna API (de manera a alivianar el desarrollo sin agregar dificultad al trabajo), es decir todo en memoria nada más (storage por ej o si quieren utilizar alguna base de datos tipo firebase no hay inconvenientes, también pueden usar json server que es para emular apis)
- El código fuente debe compartirse con el profesor mediante GITHUB o GITLAB.
  - a. Además de todos los fuentes, debe adjuntarse un manual de implementación que detalle paso a paso cómo realizar el despliegue de la aplicación.
  - b. Mi usuario en ambos repositorio es sosacataldo@gmail.com

### Requisitos técnicos mínimos

- El proyecto debe desarrollarse en Angular.
- La persistencia debe realizarse en memoria local (listas).
- La interfaz debe ser clara, responsive y coherente con la temática del sistema.

# Enunciado — Sistema de Reservas de Mesas en Restaurantes

Se solicita desarrollar un frontend en Angular que gestione restaurantes, sus ambientes internos y la reserva de mesas. Todos los datos deben almacenarse en `LocalStorage`.

---

## 1) Administración de Restaurantes

El sistema debe permitir mantener un listado de restaurantes.  
Campos mínimos:

- `id`
- `nombre`

Debe poder agregarse, editarse y eliminarse.

---

## 2) Administración de Zonas

Cada restaurante puede tener múltiples zonas o ambientes, por ejemplo: terraza, salón, patio, barra, etc.  
Campos mínimos:

- `id`
- `nombre`
- `restaurante al que pertenece`

La interfaz debe permitir ver únicamente las zonas del restaurante seleccionado.

---

## 3) Administración de Mesas

Cada zona dispone de varias mesas.  
Campos mínimos:

- `id`
- `número de mesa`
- `capacidad`
- `zona a la que pertenece`

La creación de una mesa debe permitir elegir un restaurante y luego una zona de ese restaurante.

---

## 4) Configuración de Horarios de Reserva por Zona

Cada zona debe tener configurados sus propios horarios posibles de reserva, pudiendo variar de un restaurante a otro.

Ejemplo (solo ilustrativo):

- `Zona Patio` — 11:00, 12:00, 13:00, 20:00, 21:00

El sistema debe permitir asignar o modificar estos horarios por zona.

---

## 5) Registro de Reservas

El sistema debe permitir registrar reservas de mesas. El flujo mínimo sugerido es:

1. Seleccionar restaurante
2. Seleccionar zona
3. Seleccionar fecha
4. Seleccionar uno de los horarios disponibles para esa zona
5. Ingresar cantidad de personas
6. Ingresar datos del cliente (nombre, apellido, teléfono)
7. Confirmar reserva

Al confirmar, el sistema debe:

- Buscar disponibilidad real de mesas para esa zona, fecha y hora.
- Considerar capacidad mínima necesaria según la cantidad de personas.
- Asegurarse de no asignar mesas ya reservadas para el mismo momento.
- Elegir automáticamente la mesa más adecuada entre las disponibles.
- Registrar la reserva solo si existe una mesa válida.

Estructura mínima sugerida para las reservas:

- id
  - fecha
  - hora
  - cantidadPersonas
  - idMesaAsignada
  - nombre
  - apellido
  - telefono
- 

## 6) Listado de Reservas

Debe haber una pantalla donde se puedan consultar las reservas existentes con filtros por:

- restaurante
  - zona
  - fecha
-