

Aplicación: Sistema de Gestión de Reservas para Restaurante en Ionic

Contexto del Proyecto

Desarrolle una aplicación en **Ionic** (Angular, React o Vue) que permita gestionar **reservas en un restaurante**. Para ello, debe crear tres componentes reutilizables, una pantalla principal que integre estos componentes y un backend funcional que permita realizar operaciones sobre las reservas.

Componentes Reutilizables

Implemente los siguientes **tres componentes** como elementos independientes y reutilizables:

1. **Componente de selección de fecha y hora** de la reserva.
2. **Componente de ingreso de datos del cliente** (nombre, contacto, etc.).
3. **Componente para escoger la mesa** entre las disponibles.

Pantalla Principal

Cree una pantalla principal llamada **Gestión de Reservas**, que consuma los tres componentes anteriores y unifique todo el proceso de reserva en una sola vista interactiva.

Backend Funcional

- Desarrolle un backend que exponga una **API REST** con operaciones CRUD completas sobre las reservas.
- Utilice **Swagger** para documentar y probar la API.
- El backend debe almacenar y gestionar los datos de fecha/hora, cliente y mesa reservada.

Control de Versiones y Organización del Repositorio

- Cree un **repositorio nuevo** con el nombre: **parcial-corte2**.
- Utilice **Git** para gestionar el proyecto.
- Cree una **rama por cada Historia de Usuario (HU)**, nombradas de la siguiente manera:
 - **feat/HU-1**: Componente de fecha y hora.
 - **feat/HU-2**: Componente de datos del cliente.
 - **feat/HU-3**: Componente de selección de mesa.
 - **feat/HU-4**: Pantalla principal que integra todos los componentes.
 - **feat/HU-5**: Backend con CRUD completo y documentación Swagger.
- Todas las ramas **feat** deben ser **fusionadas a la rama develop**, la cual será la **única rama evaluada** y debe contener la versión consolidada del proyecto.

 Estructura requerida en la rama **develop**

La rama **develop** debe contener dos subcarpetas principales:

```
/develop
├── /app    → Contiene el frontend desarrollado en Ionic (Angular, React o Vue).
└── /back   → Contiene el backend con la API REST documentada en Swagger.
```

Video Explicativo

- Grabe un **video de máximo 3 minutos** que demuestre:
 - La funcionalidad de cada componente en la aplicación.
 - La integración en la pantalla principal.
 - El funcionamiento del backend a través de Swagger u otra herramienta de prueba de API.

Entregables en Moodle

1. **Archivo .zip** de la rama **develop** del proyecto.
2. **Documento Word** (manual de usuario) que contenga:
 - Capturas de cada componente y de la pantalla principal.
 - Explicación funcional de cada sección.
 - Evidencia de integración de los componentes.
3. **Enlace al repositorio GitHub** **parcial-corte2**.
4. **Enlace al video explicativo** del proyecto.

Este requerimiento tiene como objetivo demostrar el manejo de componentes reutilizables, integración de frontend con backend, control de versiones profesional y documentación funcional de un proyecto completo.