

Proyecto Semanal: Sistema de Reservas Universitarias

1. Descripción general

El estudiante debe diseñar y construir un prototipo funcional de una aplicación web para reservar espacios universitarios (aulas, canchas, laboratorios). El enfoque será **frontend** (no requiere backend real), pero debe incluir: Un flujo de reserva (formulario + confirmación visual). Validaciones básicas. Persistencia temporal (JSON o LocalStorage). Control de versiones en GitHub. Al menos una prueba automatizada.

2. Requerimientos mínimos

Requisitos Funcionales (RF): RF1: El usuario puede ver la lista de espacios disponibles. RF2: El usuario puede crear una reserva (nombre, fecha, hora, espacio). RF3: El sistema debe evitar reservas duplicadas (misma fecha y hora). RF4: El usuario puede cancelar su reserva. RF5: Se mostrará una imagen o plano del espacio seleccionado.

Requisitos No Funcionales (RNF): RNF1: El sistema debe responder en menos de 3 segundos. RNF2: La información se guarda localmente (LocalStorage o archivo JSON). RNF3: Los datos deben mostrarse en una interfaz clara y accesible. RNF4: Se debe aplicar una estructura de carpetas ordenada. RNF5: Código limpio, comentado y con commits claros.

3. Historias de Usuario

Cada estudiante debe redactar al menos tres historias de usuario siguiendo el formato **INVEST** y definir criterios de aceptación. **Ejemplos:** **HU1:** Como estudiante, quiero ver los espacios disponibles para elegir el adecuado. **HU2:** Como estudiante, quiero registrar una reserva para asegurar mi horario. **HU3:** Como estudiante, quiero ver una imagen del espacio para confirmar mi selección. Cada historia debe incluir criterios de aceptación claros y estar alineada con los requisitos funcionales definidos.

7. Recomendaciones

Usar nombres descriptivos en commits (por ejemplo: feat: agregar formulario de reserva). Aplicar flujo de ramas GitHub Flow (main + feature branches). Realizar al menos 5 commits significativos y bien descritos. Usar componentes reutilizables y CSS modular. Crear una carpeta /tests con al menos una prueba automatizada. El proyecto debe poder clonarse y ejecutarse con facilidad (npm install → npm run dev).

8. Ejercicio complementario (opcional para nota extra)

Implementar una funcionalidad adicional, como: Notificación visual o sonora al confirmar reserva. Filtro por tipo de espacio (aula, laboratorio, cancha). Validación de horario (no permitir fechas pasadas).

Puntaje adicional: hasta +5 puntos.