

## Práctica No.11

Proyecto: **Sistema de Reservación de Salas de Conferencia**

### Descripción

Un sistema que permita gestionar la reserva de salas de conferencia en una empresa. Los usuarios pueden visualizar salas disponibles, realizar reservas, actualizar o cancelar reservas, y el administrador puede agregar, modificar o eliminar salas.

### Características del Proyecto

#### Usuarios

Usuarios generales:

- Pueden ver las salas disponibles.
- Pueden hacer reservas de salas.
- Pueden editar o cancelar sus reservas.

Administrador:

- Puede agregar, modificar o eliminar salas.
- Puede gestionar las reservas (ver todas, editar o eliminar).

#### Entidades en la Base de Datos

Tabla users:

- id: int (Primary Key, auto-increment).
- name: varchar(100).
- email: varchar(100).
- password: varchar(100).
- role: enum('user', 'admin').

#### Tabla rooms:

- id: int (Primary Key, auto-increment).
- name: varchar(100).
- capacity: int.
- location: varchar(200).
- availability: boolean (true: disponible, false: no disponible).

#### Tabla reservations:

- id: int (Primary Key, auto-increment).
- user\_id: int (Foreign Key to users).
- room\_id: int (Foreign Key to rooms).
- start\_time: datetime.
- end\_time: datetime.
- status: enum('active', 'cancelled').

### Arquitectura del Proyecto

Frontend: React

#### Rutas principales:

- Página de inicio (listar salas disponibles).
- Panel de administración (solo accesible para admin).
- Gestión de reservas (crear, editar, eliminar).
- Gestión de salas (admin).

Backend: Node.js (Express.js)

#### Rutas principales:

- /api/auth: Login y registro.
- /api/rooms: CRUD de salas.
- /api/reservations: CRUD de reservas.
- Middleware para autenticar roles (user y admin).

Base de Datos: MySQL

- ORM: Sequelize o Knex.js.

## Desarrollo Paso a Paso

### Frontend (React)

- Login/Registro: Formularios conectados a la API /api/auth.
- Gestión de Salas:
  - Página para listar todas las salas disponibles.
  - Formularios para agregar, editar o eliminar salas (admin).
- Gestión de Reservas:
  - Página para listar las reservas realizadas por el usuario.
  - Botón para editar o cancelar reservas.
- Panel de Admin:
  - Página que muestre las reservas y permita gestionarla.
- Manejo de rutas privadas (usuarios y administradores).

### Backend (Node.js)

- Configurar rutas RESTful para usuarios, salas y reservas.
- Middleware de autenticación (JWT) y autorización (roles).
- Controladores para cada operación CRUD.
- Validación de entradas utilizando librerías como express-validator.

### Base de Datos (MySQL)

- Crear y poblar las tablas.
- Configurar las relaciones entre tablas (Foreign Keys).
- Probar las operaciones CRUD directamente en MySQL Workbench o una herramienta similar.

### Funcionalidades Adicionales

- Notificaciones por correo al realizar una reserva (usando nodemailer).
- Validación de horarios para evitar solapamientos entre reservas.
- Paginación en las listas de reservas y salas.