

Sistema de Gestión de Reservas de Habitaciones de un Hotel

Este proyecto consiste en un sistema para gestionar reservas de habitaciones en un hotel.

Incluye un **backend** desarrollado con Spring Boot, una base de datos MySQL, y un **frontend** en React.

Aquí se describen los pasos para su desarrollo y uso.

Descripción General del Proyecto

El sistema permite realizar las siguientes operaciones:

1. **Registrar una reserva**: Proporcionando datos como el nombre del cliente, fechas de inicio y fin,
y el tipo de habitación (Sencilla, Doble, Suite).
2. **Listar reservas existentes**: Mostrando el cliente, fechas y tipo de habitación.
3. **Actualizar fechas de una reserva**: Modificando una reserva existente.
4. **Cancelar una reserva**: Eliminándola de la base de datos.

Cómo se hizo

Backend

- **Tecnología utilizada**:

El backend se desarrolló usando **Java 17** y el **Spring Framework**.

- Se utilizó Spring Boot para configurar rápidamente un entorno de desarrollo eficiente.
- Spring Data JPA se usó para interactuar con la base de datos.
- El controlador de MySQL permitió la conexión con la base de datos.

- ****Estructura****:

- Se creó un ****API REST**** para manejar las solicitudes HTTP.
- Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) se implementaron para gestionar las reservas.

- ****Pruebas del backend****:

Usando herramientas como Postman o cURL, se probaron los endpoints de la API para asegurarse de que las operaciones CRUD funcionan correctamente.

Base de Datos

- ****Tecnología utilizada****:

Se configuró una base de datos MySQL llamada `hotel_reservas`.

- ****Estructura de la tabla****:

La tabla `reservas` contiene las siguientes columnas:

- `id`: Identificador único de la reserva (auto-incremental).
- `nombre_cliente`: Nombre del cliente.
- `fecha_inicio`: Fecha de inicio de la reserva.
- `fecha_fin`: Fecha de fin de la reserva.
- `tipo_habitacion`: Tipo de habitación reservada (Sencilla, Doble, Suite).

- ****Configuración****:

La conexión con la base de datos se realizó mediante un archivo de configuración en el backend, donde se definió el nombre de la base de datos, el usuario y la contraseña.

Frontend

- ****Tecnología utilizada****:

El frontend se desarrolló con **React**.

- **Estructura de la interfaz**:

- **Formulario de reservas**: Permite ingresar los datos necesarios para registrar una reserva.
- **Tabla de reservas**: Lista todas las reservas existentes.
- **Botones de acción**: Incluyen opciones para actualizar las fechas de una reserva o cancelarla.

- **Conexión con el backend**:

Se utilizó la biblioteca `axios` para enviar solicitudes HTTP al API REST.

- **Pruebas del frontend**:

Se verificó que todas las interacciones de la interfaz correspondan con las operaciones del backend.

Cómo usar el sistema

Requisitos previos

1. Tener instalados los siguientes programas en el sistema:

- **Java 17** o superior.
- **MySQL**.
- **Node.js** y **npm** (para ejecutar React).

2. Clonar el repositorio desde GitHub.

Instrucciones de uso

1. **Configuración de la base de datos**:

- Crear una base de datos llamada `hotel_reservas` en MySQL.
- Asegurarse de que la tabla `reservas` esté configurada correctamente.

2. **Ejecutar el backend**:

- Navegar a la carpeta del backend en la terminal.
- Ejecutar el comando para iniciar la aplicación Spring Boot.
- El backend estará disponible en `http://localhost:8080`.

3. **Ejecutar el frontend**:

- Navegar a la carpeta del frontend en la terminal.
- Instalar las dependencias usando `npm install`.
- Iniciar la aplicación con `npm start`.
- El frontend estará disponible en `http://localhost:3000`.

4. **Interacción con el sistema**:

- Registrar una nueva reserva desde el formulario.
- Ver todas las reservas en la tabla.
- Usar los botones de acción para actualizar o eliminar reservas.

Consideraciones

- Asegúrate de que el backend esté ejecutándose antes de iniciar el frontend.
- Verifica que las configuraciones de la base de datos en el backend coincidan con tu entorno local.