

Proyecto: Sistema de Gestión de Reservas para Espacios Compartidos

Descripción:

Desarrollar una API backend que gestione las reservas de espacios compartidos como salas de reuniones o zonas de trabajo colaborativo en una oficina. El sistema permitirá que los usuarios realicen y gestionen reservas de manera eficiente, manejando la concurrencia de forma controlada, asegurando la integridad de los datos y proporcionando seguridad en el acceso a la API.

Requerimientos del Proyecto:

1. Gestión de Usuarios y Roles:

- Los usuarios podrán registrarse e iniciar sesión.
- Se definirán dos roles principales: **Administrador** (gestiona los espacios y reservas) y **Usuario** (realiza las reservas).
- La autenticación y autorización se manejarán usando **Spring Security** con **JWT** (JSON Web Token).

2. Gestión de Espacios:

- Los administradores podrán crear, editar y eliminar espacios disponibles para la reserva (salas de reuniones, escritorios, etc.).
- Los usuarios podrán visualizar los espacios disponibles, filtrarlos por características (capacidad, ubicación, etc.) y consultar la disponibilidad.

3. Sistema de Reservas:

- Los usuarios podrán realizar y cancelar reservas.

- El sistema deberá garantizar que no haya reservas dobles para un mismo espacio en un intervalo de tiempo específico, manejando colisiones de reservas mediante transacciones seguras con **Spring Transaction Management**.
- Las reservas confirmadas se mantendrán bloqueadas para otros usuarios durante el período reservado.

4. **Tareas Programadas:**

- Implementar tareas automáticas con **@Scheduled** para liberar espacios que no hayan sido confirmados en un tiempo determinado o para eliminar reservas antiguas del sistema.

5. **Almacenamiento de Datos:**

- Utilizar **Spring Data JPA** para gestionar la persistencia de los datos en una base de datos relacional (MySQL o PostgreSQL).
- Almacenar información sobre usuarios, espacios y reservas.
- Asegurar un acceso eficiente y concurrente a la base de datos utilizando estrategias adecuadas para el manejo de transacciones.

6. **Seguridad y Autorización:**

- El sistema deberá proteger los endpoints de la API mediante **Spring Security**.
- Los usuarios deberán estar autenticados para realizar cualquier acción relacionada con las reservas o gestión de espacios.

- Los administradores tendrán permisos especiales para la gestión de espacios y supervisión de reservas.

7. Manejo de Errores y Excepciones:

- Implementar un manejo robusto de errores que controle posibles conflictos de reservas concurrentes.
- Mostrar mensajes de error claros cuando haya intentos de reservar espacios que ya hayan sido ocupados.

Tecnologías a Utilizar:

- **Spring Boot:** Para configurar rápidamente la aplicación.
- **Spring MVC:** Para crear los endpoints de la API REST.
- **Spring Data JPA:** Para la persistencia de datos en la base de datos.
- **Spring Security con JWT:** Para la autenticación y autorización de los usuarios.
- **@Scheduled:** Para ejecutar tareas recurrentes como la liberación de espacios no confirmados.
- **MySQL/PostgreSQL:** Como base de datos relacional.

Funcionalidades adicionales (opcional):

- Panel de administración donde los administradores puedan ver estadísticas de uso de los espacios.
- Permitir a los usuarios calificar o comentar los espacios una vez que los hayan utilizado.

- Posibilidad de integración con un servicio de pago si los espacios requieren alquiler.