

Alcance de la Solución.....	1
Sistema de Gestión de Salas de Reuniones.....	1
1. Módulo de Gestión de Salas.....	1
2. Sistema de Reservas.....	2
3. Notificaciones Automáticas.....	2
4. Sistema de Autenticación y Roles.....	3
5. Reporte de Uso de Salas.....	3
Escalabilidad y Seguridad.....	4

Alcance de la Solución

Sistema de Gestión de Salas de Reuniones

Objetivo: Desarrollar un sistema de gestión de salas de reuniones que permita la administración eficiente de salas y recursos asociados, facilite la reserva de espacios por parte de los empleados, gestione notificaciones automáticas, y genere reportes de uso detallados. El sistema debe ser seguro, escalable y de alto rendimiento para soportar el crecimiento y las necesidades operativas de la organización.

1. Módulo de Gestión de Salas

Descripción: Este módulo permite a los administradores crear, editar, y eliminar salas de reuniones, así como gestionar los recursos disponibles en cada una.

- **Funcionalidades clave:**
 - **Creación y administración de salas:** Las salas tendrán atributos como capacidad, ubicación y recursos disponibles (proyector, pizarra, etc.). Esto se realizará mediante una interfaz web que permita agregar, modificar o eliminar salas.
 - **Validaciones de entrada:** El sistema validará los datos introducidos para evitar errores, como crear una sala sin capacidad o con ubicaciones duplicadas.

Tecnología:

- **Backend:** Express.js (Node.js) para la gestión de API RESTful que maneja las operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
- **Base de Datos:** MongoDB (NoSQL) o MySQL, dependiendo del nivel de flexibilidad y escalabilidad requerida. Índices optimizados para búsqueda rápida.

Requisitos técnicos:

- Implementación de validaciones tanto en el frontend como en el backend para garantizar la consistencia de los datos.
- API protegida con autenticación basada en tokens JWT.

2. Sistema de Reservas

Descripción: Este módulo permitirá a los usuarios ver la disponibilidad de las salas en un calendario interactivo y realizar reservas para un periodo específico, asegurando que no haya conflictos de horario.

- **Funcionalidades clave:**
 - **Visualización de disponibilidad:** Los usuarios podrán ver en un calendario interactivo las horas disponibles y reservadas de cada sala.
 - **Reservas:** Los usuarios pueden seleccionar una sala y un rango de horas, y el sistema validará si está disponible para evitar colisiones.
 - **Modificación y cancelación:** Los usuarios pueden modificar o cancelar sus reservas, siempre que estén dentro de los plazos permitidos.

Tecnología:

- **Frontend:** React.js o Angular para la creación de una interfaz intuitiva y responsive.
- **Backend:** API RESTful en Express.js que maneje la creación, modificación, y cancelación de reservas.
- **Base de Datos:** MongoDB con optimizaciones de índices para gestionar eficientemente la disponibilidad de las salas y evitar conflictos de horarios.

Requisitos técnicos:

- **Control de concurrencia:** El sistema debe implementar mecanismos de bloqueo de registros o transacciones para evitar reservas simultáneas conflictivas.
- **Validación de datos:** Validar las fechas y horas seleccionadas por los usuarios para asegurar que son válidas y disponibles.
- **Almacenamiento de reservas:** Optimización de las consultas de reservas utilizando particionamiento de datos por fecha o sala, si es necesario.

3. Notificaciones Automáticas

Descripción: Este módulo permitirá el envío de notificaciones a los usuarios cuando se realicen reservas, modificaciones o cancelaciones. También notificará a los usuarios cuando una sala se libere debido a una cancelación.

- **Funcionalidades clave:**
 - **Notificaciones por correo:** Envío automático de correos electrónicos cuando una reserva es creada, modificada o cancelada.
 - **Notificación de cancelaciones:** Los usuarios recibirán un aviso si una sala que estaba ocupada se libera debido a la cancelación de una reserva.

Tecnología:

- **Servicio de correo:** Nodemailer (en el backend) para el envío de correos electrónicos a los usuarios. Se pueden usar servicios de terceros como SendGrid o Amazon SES.
- **Backend:** Express.js para gestionar la lógica de notificaciones.

Requisitos técnicos:

- Implementación de colas de mensajes (con Redis o RabbitMQ) para asegurar la entrega de notificaciones sin retrasos.
- Configuración de plantillas de correo dinámicas que reflejen los detalles de la reserva o modificación.

4. Sistema de Autenticación y Roles

Descripción: El sistema contará con un esquema de autenticación basado en JWT que asegurará que solo los usuarios registrados puedan acceder y realizar acciones. Además, los usuarios tendrán diferentes roles, como Administrador o Empleado, con permisos específicos.

- **Funcionalidades clave:**
 - **Autenticación basada en JWT:** Los usuarios deben iniciar sesión para acceder al sistema. Las solicitudes a la API estarán protegidas por tokens JWT.
 - **Roles y permisos:** Los administradores podrán gestionar las salas y visualizar todos los reportes, mientras que los empleados solo podrán realizar y gestionar sus propias reservas.

Tecnología:

- **Autenticación:** Implementación de JSON Web Tokens (JWT) para asegurar las rutas de la API.
- **Gestión de roles:** Middleware en Express.js que verifique los permisos del usuario antes de permitir la ejecución de ciertas acciones.

Requisitos técnicos:

- **Encriptación de contraseñas:** Uso de bcrypt para el almacenamiento seguro de contraseñas.
- **Control de acceso basado en roles (RBAC):** Definición de rutas y permisos específicos para cada rol en el sistema.

5. Reporte de Uso de Salas

Descripción: Generación de reportes detallados sobre el uso de las salas, como frecuencia de uso, horas reservadas, etc. Los administradores podrán consultar estos datos para optimizar el uso de recursos.

- **Funcionalidades clave:**
 - **Generación de reportes:** Reportes que muestren las horas reservadas por sala, la frecuencia de uso, y los usuarios que más utilizan las salas.
 - **Exportación de datos:** Posibilidad de exportar los reportes en formatos como CSV o PDF.

Tecnología:

- **Frontend:** Generación de tablas dinámicas en React.js con filtros para facilitar la visualización de datos.
- **Backend:** Generación de reportes utilizando Express.js y MongoDB (o MySQL) para el cálculo de estadísticas de uso.

Requisitos técnicos:

- **Consultas agregadas:** Uso de consultas agregadas en MongoDB o MySQL para obtener métricas de uso de las salas.
- **Programación de tareas:** Posibilidad de programar la generación de reportes en intervalos específicos utilizando herramientas como cron jobs.

Escalabilidad y Seguridad

Escalabilidad: El sistema está diseñado para ser escalable, utilizando arquitecturas basadas en microservicios y tecnologías en la nube (AWS, Azure) para escalar horizontalmente según sea necesario.

- **Uso de contenedores:** Docker y Kubernetes para la implementación y administración de microservicios de manera distribuida.
- **Balanceo de carga:** Uso de balanceadores de carga para distribuir el tráfico entre varias instancias del servidor.

Seguridad:

- **Cifrado de datos:** Los datos sensibles estarán cifrados tanto en tránsito (TLS/SSL) como en reposo (en bases de datos).
- **Autenticación robusta:** Se implementará un sistema de autenticación robusto con medidas de protección contra ataques como fuerza bruta, cross-site scripting (XSS), y SQL injection.