

## **Tecnologías**

- Nodejs
- Express
- MongoDB
- React

## **Casos de uso**

### **Usuarios y autenticación**

Crear un sistema de autenticación que permita a los usuarios registrarse como administradores o pacientes, iniciar sesión y recuperar su contraseña.

Se incluye desde la gestión de médicos y turnos médicos, hasta estadísticas y seguimiento del historial de los mismos.

### **Endpoints a generar**

#### **Usuarios**

- Registrar un usuario, puede ser admin o paciente
- Iniciar sesión (generar jwt)
- Recuperar contraseña (enviar correo electrónico con enlace de recupero)

#### Observaciones

1. La autenticación debe ser basada en JWT
2. Se deben implementar mecanismos seguros para el manejo de contraseñas
3. La recuperación de contraseña debe incluir un enlace único y seguro, enviado por correo electrónico

#### **Médicos**

- Listar todos los médicos registrados en el sistema
- Mostrar el detalle de un médico incluyendo sus turnos disponibles
- Agregar un nuevo médico al sistema (solo disponible como admin)
- Actualizar información de un médico en el sistema (solo disponible como admin)

#### **Especialidades**

- Listar todas las especialidades médicas disponibles

#### **Turnos**

- Crear un nuevo turno disponible en el sistema (solo disponible como admin)
- Actualizar la información de un turno disponible en el sistema (solo disponible como admin)
- Eliminar un turno disponible del sistema (solo disponible como admin)
- Listar turnos por médicos (solo disponible como admin)
- Listar turnos por pacientes
- Reservar turno (solo disponible para paciente)
- Cancelar turno (solo disponible para paciente)

- Historial de cancelaciones por paciente

**Observaciones finales**

1. Asegurarse de que los endpoints sean seguros y solo accesibles para usuarios autenticados
2. Implementar el manejo adecuado de errores y respuestas del servidor
3. Utilizar paginación para todos los listados
4. Considerar la lógica de negocio para gestionar la disponibilidad de turnos y reservas