Ejercicio (API de parking)

Descripción del ejercicio

Deberás construir una API RESTful utilizando Node.js para la gestión de un parking.

Casos de uso

1. **Reservar una plaza de aparcamiento**: un *cliente* desea reservar una plaza de aparcamiento para un vehículo en particular. El cliente hace una solicitud POST a la API con los detalles del vehículo y la fecha y hora de la reserva. La API verifica que haya una plaza de aparcamiento disponible en la fecha y hora especificadas y reserva la plaza de aparcamiento para el vehículo del cliente. La API devuelve una respuesta con los detalles de la reserva.
2. **Consultar la ocupación del parking**: un *empleado* desea conocer la ocupación actual del parking. El empleado hace una solicitud GET a la API para obtener información sobre la ocupación actual del parking. La API consulta la base de datos para obtener información actualizada sobre las plazas de aparcamiento ocupadas y devuelve una respuesta al empleado con los detalles de la ocupación del parking.
3. **Actualizar los detalles de un usuario**: un administrador desea actualizar los detalles de un usuario en particular, como su nombre, dirección de correo electrónico o número de teléfono. El administrador hace una solicitud PUT a la API con los detalles actualizados del usuario. La API verifica que el usuario tenga los permisos adecuados para realizar la actualización y actualiza los detalles del usuario en la base de datos. La API devuelve una respuesta con los detalles actualizados del usuario.
4. **Acceder a los logs del parking:** un administrador desea acceder a los logs de actividad del parking para conocer el historial de reservas, cancelaciones, entradas y salidas de vehículos, etc. El administrador hace una solicitud GET a la API para obtener los registros de actividad del parking. La API verifica que el usuario tenga los permisos adecuados para acceder a los registros y devuelve una respuesta con los registros de actividad del parking.

La aplicación tendrá las siguientes características funcionales adicionales:

- La aplicación debe tener un sistema de autenticación basado en JWT.
- La aplicación debe tener un sistema de autorización basado en roles. Los roles serán "admin", "empleado" y "cliente".
- La aplicación debe permitir la creación, lectura, actualización y eliminación de entidades a través de la API.

Requisitos tecnicos del ejercicio

- La aplicación debe estar escrita en JavaScript o Typescript, utilizando Node.js, debe utilizar Express o Nestjs con Express.
- La aplicación debe utilizar MongoDB para persistencia de los logs y PostgreSQL o MySQL como base de datos principal para la persitencias de las entidades del negocio.
- La aplicación debe tener pruebas e2e automatizadas para los 3 casos de uso

Entregables

- Código fuente de la aplicación en github.
- Documentación técnica que explique cómo ejecutar la aplicación y cómo utilizar la API.
- Coleccion de postman para los endpoints de la aplicación

Plazo de entrega

El plazo de entrega es de un **maximo de 3 semanas**, la fecha limite se acuerda en el momento de la entrega del ejercicio al candidato. Cumplir con la fecha limite es un requisito **necesario**

Aspectos a Evaluar

Se evaluarán los siguientes aspectos:

- La calidad del código y la arquitectura de la aplicación.
- La capacidad del candidato para implementar un sistema de autenticación basado en tokens JWT.
- La capacidad del candidato para implementar un sistema de autorización basado en roles.
- La capacidad del candidato para trabajar con varias bases de datos en una misma aplicación.
- La capacidad del candidato de implementar un CRUD
- La capacidad del candidato de implementar logica de negocio
- La capacidad del candidato de llevar un registro de logs relevantes relacionado con los procesos criticos de la aplicacion
- La capacidad del candidato para implementar tests e2e.
- La capacidad del candidato para usar de forma efectiva herramientas complementarias como Postman (tambien puede usar cualquier herramienta analoga)
- La capacidad del candidato para documentar su trabajo.
- La capacidad del candidato para usar git