

Documentación Técnica: Sistema de Gestión de Entidades y Empleados

Índice

1. Descripción del Proyecto
2. Tecnologías Utilizadas
3. Estructura del Proyecto
4. Endpoints del Backend
5. Interfaz de Usuario (Frontend)
6. Flujo de Autenticación JWT
7. Paginación y Manejo de Empleados
8. Swagger y CORS
9. Pruebas

1. Descripción del Proyecto

Este proyecto implementa un sistema de gestión de entidades y empleados con las siguientes funcionalidades:

CRUD de Empleados: Crear, leer, actualizar y eliminar empleados.

CRUD de Entidades: Crear, leer, actualizar y eliminar entidades.

Autenticación: Sistema de login con JWT para proteger ciertas rutas.

Frontend interactivo: React se utiliza para presentar una interfaz interactiva con funcionalidades como paginación y edición en tiempo real.

2. Tecnologías Utilizadas

Backend:

- .NET 6
- MongoDB
- Autenticación con JWT
- Swagger

Frontend:

- React
- Bootstrap
- CSS personalizado

Base de Datos:

- MongoDB

3. Estructura del proyecto

Backend:

```
bash
Copiar código
EntitiesManagementAPI/
├── Controllers/      # Controladores de la API
├── Models/           # Modelos para MongoDB
├── Services/         # Lógica de negocio y acceso a MongoDB
└── Program.cs        # Configuración principal del backend
```

Frontend:

```
bash
Copiar código
src/
|
```

— components/	# Componentes de React (EmployeeList, EntityList)
— assets/	# Recursos como imágenes y estilos CSS
— services/	# Archivo API para realizar peticiones al backend
— App.js	# Configuración de rutas y componentes principales

4. Enpoints del backend

Método	Endpoint	Descripción	Autenticación
POST	/api/auth/login	Inicia sesión y devuelve un token JWT	No
GET	/api/employees	Obtiene todos los empleados	No
GET	/api/employees/{id}	Obtiene un empleado por su ID	No
POST	/api/employees	Crea un nuevo empleado	Sí
PUT	/api/employees/{id}	Actualiza un empleado existente	Sí
DELETE	/api/employees/{id}	Elimina un empleado por su ID	Sí
GET	/api/entities	Obtiene todas las entidades	No
GET	/api/entities/{id}	Obtiene una entidad por su ID	No
POST	/api/entities	Crea una nueva entidad	Sí
PUT	/api/entities/{id}	Actualiza una entidad existente	Sí
DELETE	/api/entities/{id}	Elimina una entidad por su ID	Sí

5. Interfaz de Usuario (Frontend)

Componentes Principales:

Home: Página inicial que muestra opciones como empleados, entidades, documentación y manual.

EmployeeList: Manejo de empleados con funciones de:

- Crear, editar y eliminar empleados.
- Paginación dinámica.

EntityList: Manejo de entidades con funcionalidades similares a EmployeeList.

6. Flujo de Autenticación JWT

1.Inicio de Sesión:

El usuario envía su nombre y contraseña a /api/auth/login.

Si las credenciales son válidas, el servidor devuelve un token JWT.

2.Acceso Protegido:

Las rutas como creación, edición y eliminación requieren que el usuario envíe el token en el encabezado Authorization.

7. Paginación y Manejo de Empleados

Frontend:

Los empleados se cargan por partes (4 empleados por página).

La paginación es controlada por el estado currentPage.

Backend:

Los datos se obtienen de MongoDB sin paginación nativa, pero podrían adaptarse.

8. Swagger y CORS

Swagger:

Disponible en el entorno de desarrollo en /swagger.

Documentación interactiva con soporte para tokens JWT.

CORS:

Configurado para permitir solicitudes desde http://localhost:3000.

9. Pruebas

Pruebas Manuales:

Verificar el flujo de autenticación.

Probar las operaciones CRUD en empleados y entidades.

Pruebas de UI:

Probar la interfaz de React para garantizar la usabilidad.