UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCIÓN A LA PROGRAMACIÓN 1

CATEDRÁTICO: ING. NETFALÍ CALDERÓN

TUTOR ACADÉMICO: RODRIGO PORÓN

MANUAL TÉCNICO

JORGE ANDRÉS MEJÍA SÚCHITE

CARNÉ: 202300376

SECCIÓN: E

GUATEMALA, 27 DE ABRIL DEL 2024

Índice

Descripción General de la Solución

Este manual técnico presenta el proyecto que se ha dividido en dos partes fundamentales: el frontend y el backend. El frontend se ha desarrollado utilizando Next.js, mientras que el backend se ha implementado con Nest.js, ejecutándose sobre Express. La aplicación hace uso de MongoDB como su base de datos principal, aprovechando diversas rutas diseñadas para ejecutar acciones específicas. Además, cuenta con almacenamiento de archivos en el lado del servidor para almacenar las imágenes utilizadas. Este manual proporciona una guía detallada para comprender y operar eficazmente todas las facetas del proyecto.

Diagrama de Clases

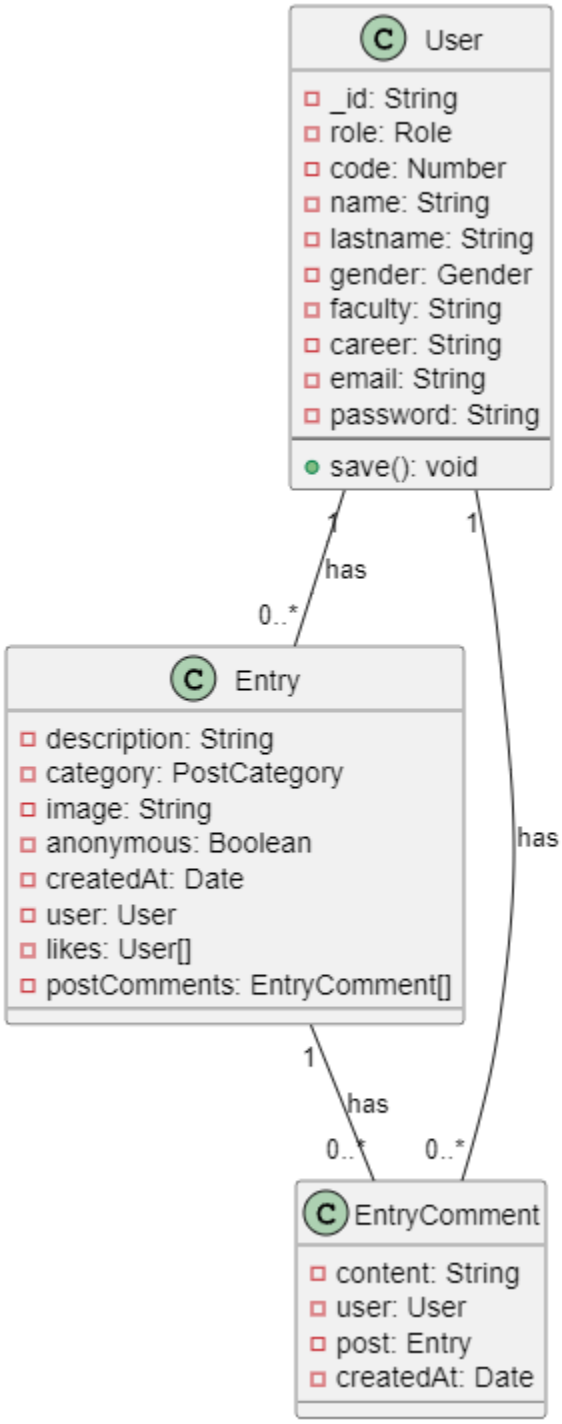

Tabla de Endpoints

Ruta	Método	Descripción
/auth/login	POST	Este método maneja las solicitudes de inicio de sesión. Toma un objeto SignInDto como entrada y llama al método signIn del servicio authService para autenticar al usuario. Devuelve el resultado de la autenticación.
/auth/me	GET	Este método maneja las solicitudes para obtener información del usuario actualmente autenticado. Obtiene la información del usuario desde la solicitud y devuelve un objeto con los datos del usuario y la expiración del token.
/auth/admin	GET	Este método maneja las solicitudes para una ruta específica destinada a administradores. Requiere que el usuario tenga el rol de administrador (Role.Admin). Devuelve un mensaje indicando que el usuario es administrador.

/entries/top	GET	Este método obtiene las cinco mejores entradas según los likes. Retorna las cinco mejores entradas.
/entries/categories	GET	Este método obtiene el número de publicaciones por categoría. Retorna el número de publicaciones por categoría.
/entries/users	GET	Este método obtiene los diez usuarios con más publicaciones. Retorna los diez usuarios con más publicaciones.
/entries/id	GET	Este método obtiene una entrada por su ID. Toma el ID de la entrada como parámetro de consulta. Retorna la entrada correspondiente al ID proporcionado.
/entries	POST	Este método crea una nueva entrada. Toma un objeto como cuerpo de la solicitud y una imagen como archivo adjunto. Retorna la entrada creada.
/entries	DELETE	Este método elimina una entrada. Toma el ID de la entrada como parámetro de consulta. Requiere que el usuario tenga el rol de

		administrador. Retorna el resultado de la eliminación.
/users	GET	Este método obtiene todos los usuarios. Requiere que el usuario tenga el rol de administrador (Role.Admin). Retorna todos los usuarios.
/users/code	GET	Este método encuentra un usuario por su código. Toma el código del usuario como parámetro de consulta. Retorna el usuario correspondiente al código proporcionado.
/users/id	GET	Este método encuentra un usuario por su ID. Toma el ID del usuario como parámetro de consulta. Retorna el usuario correspondiente al ID proporcionado.
/users/profile	GET	Este método obtiene el perfil del usuario autenticado. Toma el código del usuario del objeto de solicitud (req.user.code). Retorna el perfil del usuario correspondiente al código proporcionado.

/users	POST	Este método crea un nuevo usuario. Este método es público, es decir, no requiere autenticación. Toma un objeto CreateUserDto como cuerpo de la solicitud. Retorna el usuario creado.
/users/json	POST	Este método crea nuevos usuarios a partir de un archivo JSON. Requiere que el usuario tenga el rol de administrador (Role.Admin). Toma un archivo JSON como archivo adjunto. Retorna los usuarios creados a partir del archivo JSON.
/users/profile	POST	Este método actualiza el perfil del usuario autenticado. Toma un objeto UpdateUserDto como cuerpo de la solicitud y el código de usuario del objeto de solicitud (req.user.code). Retorna el usuario actualizado.
/users	DELETE	Este método elimina un usuario. Requiere que el usuario tenga el rol de administrador (Role.Admin). Toma el ID del usuario como parámetro de

		consulta. Retorna el resultado de la eliminación.