

Manual Técnico:

Descripción general de la solución:

Este código (backend) fue desarrollado usando el framework Express en Node.js. El proyecto tiene varias funciones, como la autenticación de usuarios, gestión de publicaciones y comentarios, y el manejo de usuarios. La API puede realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para usuarios, publicaciones y comentarios.

Diagrama de clases:

Tabla de endpoints:

Métodos POST:

Método	Endpoint	Entrada	Salida	Descripción
POST	/api/login	password, carnet (JSON)	{message: "Usuario encontrado", usuario: user}	Autentica un usuario
POST	/api/register	email, password, carnet, nombre, apellido, genero, facultad, carrera (JSON)	{message: "Usuario creado"}	Registra a un nuevo usuario
POST	/api/createPost	anonimo, carnet, descripcion, imagen, nombre, apellido, carrera, facultad, categoria (JSON)	{message: "Post creado exitosamente", post}	Crea un nuevo post
POST	/api/addComment	postId, userId, text (JSON)	{mensaje: "Comentario agregado exitosamente", comentario}	Agrega un nuevo comentario a un post

POST	'/api/Like'	postId, userId (JSON)	{mensaje: "Like agregado exitosamente"}	Agrega o quita un like a un post
-------------	-------------	--------------------------	---	----------------------------------

Métodos GET:

Método	Endpoint	Salida	Descripción
GET	'/api/getPosts'	{publicaciones: list_posts}	Obtengo todas las publicaciones.
GET	'/api/getAllUsers'	{usuarios: list_users}	Obtengo todos los usuarios.

Métodos PUT:

Método	Endpoint	Entrada	Salida	Descripción
PUT	'/api/update'	email, password, nombre, apellido, genero, facultad, carrera (JSON)	{message: "Usuario actualizado", usuarioActualizado}	Actualiza los datos de un usuario.

Métodos DELETE:

Método	Endpoint	Entrada	Salida	Descripción
DELETE	/api/deleteUser	carnet (JSON)	{message: "Usuario eliminado"}	Elimina un usuario por su carnet.
DELETE	/api/deletePost	id (JSON)	{message: "Post eliminado"}	Elimina un post por su medio de su id.