

# Proyecto Final Nest JS

## 1. Objetivo

El proyecto consiste en crear nuestro propio blog para esto tenemos definido una serie de endpoints RESTful

(<https://iamdoomling.notion.site/iamdoomling/Trabajo-pr-ctico-final-f366a1dab34245ae83726bb31fb59a25>) para gestionar usuarios, posts y autenticación usando Nest.js. Considerando la utilización de MongoDB y un middleware de autorización para usuarios administradores.

## 2. Autenticación

Para la autenticación usamos Password (jwt) y obtendremos un access\_token, el cual enviaremos en Authorization al consumir las apis.

## 3. Base de Datos

Para la base de datos usaremos MongoDB y podremos ver en `mongodb://localhost:27017`

## 4. Apis

### 4.1. Usuarios

Para el endpoint `DELETE /users/{id}` (solo administradores) se usó un middleware **AdminMiddleware** para validar si el usuario isAdmin.

	Endpoint	Descripción	Swagger
<code>POST /users</code>	METHOD: POST <code>http://localhost:3000/users</code>	Registrar nuevos usuarios. Cada usuario debe tener username, password y isAdmin	<a href="http://localhost:3000/swagger#/Users/AuthController_userRegister">http://localhost:3000/swagger#/Users/AuthController_userRegister</a>
<code>POST /users/login</code>	METHOD: POST <code>http://localhost:3000/users/login</code>	Inicio de sesión para usuarios.	<a href="http://localhost:3000/swagger#/Users/AuthController_login">http://localhost:3000/swagger#/Users/AuthController_login</a>

GET /users	METHOD: GET http://localhost:3000/users	Listado de usuarios	<a href="http://localhost:3000/swagger#/Users/AuthController_getUsers">http://localhost:3000/swagger#/Users/AuthController_getUsers</a>
GET /users/{id}	METHOD: GET http://localhost:3000/users/65b52ccb3c90c7a8ad033933	Obtener detalle de un usuario específico pasando la query el userId.	<a href="http://localhost:3000/swagger#/Users/AuthController_getUserById">http://localhost:3000/swagger#/Users/AuthController_getUserById</a>
PUT /users/{id}	METHOD: PUT http://localhost:3000/users/65b52f4cda03af7bbe5c5105	Actualizar un usuario específico (sólo su propio perfil o si es administrador).	<a href="http://localhost:3000/swagger#/Users/AuthController_updateUserById">http://localhost:3000/swagger#/Users/AuthController_updateUserById</a>
DELETE /users/{id}	METHOD: DELETE http://localhost:3000/users/delete/65b59ef8b1c3574110fd1109	Eliminar un usuario (solo administradores).	<a href="http://localhost:3000/swagger#/Users/AuthController_deleteUserById">http://localhost:3000/swagger#/Users/AuthController_deleteUserById</a>

## 4.2. Posts

	Endpoint	Descripción	Swagger
POST /posts	METHOD: POST http://localhost:3000/posts	Crear un nuevo post (solo usuarios registrados). Los post tendrán id, título, autor, contenido y un array de categorías	<a href="http://localhost:3000/swagger#/Posts/PostsController_create">http://localhost:3000/swagger#/Posts/PostsController_create</a>
GET /posts	METHOD: GET http://localhost:3000/posts?rows=10&page=0	Listado de todos los posts. Debe admitir parámetros para paginar	<a href="http://localhost:3000/swagger#/Posts/PostsController_getAllPosts">http://localhost:3000/swagger#/Posts/PostsController_getAllPosts</a>

		resultados (el default de resultados si no hay param será 10)	
GET /posts/{id}	METHOD: GET http://localhost:3000/posts/65b5e83cc4126a41e536f077	Ver detalles de un post específico.	<a href="http://localhost:3000/swagger#/Posts/PostsController_getPostById">http://localhost:3000/swagger#/Posts/PostsController_getPostById</a>
PUT /posts/{id}	METHOD: PUT http://localhost:3000/posts/65b5f57e118534d63841b542	Actualizar un post (solo el autor o administradores).	<a href="http://localhost:3000/swagger#/Posts/PostsController_updatePostById">http://localhost:3000/swagger#/Posts/PostsController_updatePostById</a>
DELETE /posts/{id}	METHOD: DELETE http://localhost:3000/posts/65b606c98594dad2a4dd53fc	Eliminar un post (solo el autor o administradores).	<a href="http://localhost:3000/swagger#/Posts/PostsController_deleteById">http://localhost:3000/swagger#/Posts/PostsController_deleteById</a>
GET /posts/user/{userId}	METHOD: GET http://localhost:3000/posts/user/65b59eb7b1c3574110fd1100	Ver todos los posts de un usuario específico.	<a href="http://localhost:3000/swagger#/Posts/PostsController_getPostByUser">http://localhost:3000/swagger#/Posts/PostsController_getPostByUser</a>

### 4.3. Búsqueda y Filtrado

	Endpoint	Descripción	Swagger
GET /posts/search	Rehusamos la api GET /posts y adicionamos query título y contenido  METHOD: GET http://localhost:3000/posts?rows=10&title=posts19&content=welcome&page=0	Buscar posts por título, contenido, etc. Debe admitir parámetros para paginar resultados (el default de resultados si no hay param será	<a href="http://localhost:3000/swagger#/Posts/PostsController_getAllPosts">http://localhost:3000/swagger#/Posts/PostsController_getAllPosts</a>

		10)	
GET /posts/filter	<p>Rehusamos la api GET /posts y adicionamos query categoría y autor</p> <p>METHOD: GET</p> <p>http://localhost:3000/posts?rows=10&amp;title=posts19&amp;content=welcome&amp;categories=travel,experience&amp;author=test1&amp;page=0</p>	Endpoints adicionales para filtrar posts por categoría o autor	<a href="http://localhost:3000/swagger#/Posts/PostsController_getAllPosts">http://localhost:3000/swagger#/Posts/PostsController_getAllPosts</a>

#### 4.4. Administración

Para los endpoints se usó un middleware adicional **AdminMiddleware** para validar si el usuario isAdmin.

	Endpoints	Descripción	Swagger
GET /admin/users	<p>METHOD: GET</p> <p>http://localhost:3000/posts/admin/users</p>	Obtener todos los usuarios (solo administradores).	<a href="http://localhost:3000/swagger#/Posts/PostsController_getAllUser">http://localhost:3000/swagger#/Posts/PostsController_getAllUser</a>
DELETE /admin/users/{id}	<p>METHOD: DELETE</p> <p>http://localhost:3000/posts/admin/users/65b99aa63685da61fdae0b7c</p>	Eliminar usuarios (solo administradores).	<a href="http://localhost:3000/swagger#/Posts/PostsController_deleteAdminById">http://localhost:3000/swagger#/Posts/PostsController_deleteAdminById</a>
GET /admin/posts	<p>METHOD: GET</p> <p>http://localhost:3000/posts/admin/posts</p>	Obtener todos los posts con opciones de moderación (borrar o editar) (solo administradores).	<a href="http://localhost:3000/swagger#/Posts/PostsController_getAllPostsAdmin">http://localhost:3000/swagger#/Posts/PostsController_getAllPostsAdmin</a>

#### 5. Despliegue

Para el despliegue usaremos Lightsail aws

Ruta:

Github: <https://github.com/Janetquispeu/nest-project> (rama main)

## 6. Documentación de Apis

Swagger: <http://localhost:3000/swagger#/>