

Introducción

En el proyecto de backend se combinará los conocimientos adquiridos en las tecnologías node + express, además de MongoDB/mongoose.

El proyecto consistirá en una red social.

Descripción

Una vez analizadas las necesidades del proyecto, se espera que el alumno desarrolle una API REST que sea capaz de lo siguiente:

- Registro de usuarios usando Bcrypt.
- Login de usuarios + token + middleware.
- Que sea capaz de crear un CRUD.
- Dar/quitar Like a post.
- Backend disponible en producción.

Requisitos imprescindibles del proyecto:

- Uso de ramas con git, cuando se termine el proyecto deberán quedar dos ramas la master o main y la develop.
- Presentación de **README excelente**.

Tecnologías

Para el desarrollo de la API utilizaremos MongoDB con Mongoose y express.

El proyecto se subirá a un repositorio público de GitHub y se valorará la existencia de ramas, así como diversos commits de calidad legible alta para analizar la evolución del proyecto.

Será requisito indispensable que el repositorio disponga de un **Readme completo**.

1.1.Endpoints

- Posts
 - Endpoint para crear un post(tiene que estar autenticado)
 - Endpoint para actualizar un post (tiene que estar autenticado)
 - Endpoint para eliminar un post(tiene que estar autenticado)
 - Endpoint para traer todos los posts junto a los usuarios que hicieron ese post y junto a los comentarios del post
 - Endpoint para buscar post por nombre
 - Endpoint para buscar post por id
 - Implementa validación a la hora de crear un post para que se rellene todos los campos(salvo la imagen, que no sea requerida) y si no se hace que devuelva un mensaje
 - Paginación de 10 en 10
 - Likes:
 - Endpoint para dar un like a un post
 - Endpoint para quitar like a un post
- Comments
 - Endpoint para crear un comentario en un determinado post
- Usuarios
 - Endpoint para registrar un usuario utilizando bcrypt
 - Endpoint para login(utilizando bcrypt +JWT)
 - Endpoint que nos traiga la información del usuario conectado
 - Endpoint para el logout
 - Implementa validación a la hora de crear un usuario para que se rellene todos los campos y si no se hace que devuelva un mensaje
- Backend disponible en producción.
- Middleware para comprobar la autoría del post a la hora de editar/eliminar el mismo.

1.2. Extras

***Importante, si te dispones a hacer los extras de los comentarios se recomienda (si aún no lo has hecho) que crees una colección para los comentarios.**

- Middleware para comprobar la autoría del comentario a la hora de editar/eliminar el mismo.
- Implementa el middleware [multer](#) para poder adjuntar imágenes al crear o actualizar posts.
- Implementa el middleware [multer](#) para poder adjuntar imágenes al crear o actualizar comentarios.
- Implementa el middleware [multer](#) para poder adjuntar imágenes al crear o actualizar un usuario.
- Implementación de followers:
 - Que puedas seguir a otros usuarios
 - Que puedas dejar de seguir a otros usuarios
- El Endpoint que nos trae la información del usuario conectado, además que nos traiga los posts y el número de seguidores que tiene
- Implementa el correo de confirmación para el registro
- Validación en el login:
 - Si no has confirmado tu correo no puedes conectarte
- Endpoint que nos trae la información del usuario conectado junto a sus post y número de followers, también que nos muestre el nombre de los followers que siguen al usuario conectado
- El endpoint que trae todos los posts junto a los usuarios que hicieron ese post y junto a los comentarios del post que también traiga los usuarios que hicieron los comentarios
- Endpoint para buscar usuario por nombre
- Endpoint para buscar usuario por id
- Crea una documentación de los endpoints de tu proyecto
- Comments
 - CRUD comments
 - Likes:
 - Dar un like a un comentario
 - Quitar like a un comentario

1.3. Entregables

- Subir el proyecto a un repositorio en github
- Dejar el enlace del repositorio adjuntado en Classroom