

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 app de bonificación para los alumnos.

Colección "Usuarios":

Campos: ID, Nombre, Apellido, Rol, Puntos, etc.

Esta colección almacenaría la información de todos los usuarios, incluyendo alumnos y profesores. Cada usuario tendría un identificador único, nombre, apellido, rol (alumno o profesor), puntos acumulados y otros detalles relevantes.

Colección "Grupos":

Campos: ID, Nombre, AlumnosIds, ProfesorId, etc.

En esta colección se registrarían los diferentes grupos. Cada grupo tendría un identificador único, un nombre descriptivo, una lista de identificadores de alumnos asignados a ese grupo y el identificador del profesor a cargo.

Colección "Dudas":

Campos: ID, Usuario_ID(alumno), Pregunta, Fecha, Resuelta, etc.

Esta colección registraría las dudas planteadas por los alumnos, relacionando cada duda con el alumno que la generó, la pregunta en sí, la fecha en que se planteó y un campo para indicar si la duda ha sido resuelta o no.

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 votación a una respuesta de una duda.
- 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

- Dudas
 - Endpoint para crear una duda (tiene que estar autenticado).
 - Endpoint para actualizar una duda (tiene que estar autenticado).
 - Endpoint para eliminar una duda (tiene que estar autenticado).
 - Endpoint para traer todas las dudas junto a los usuarios que hicieron esas dudas y junto a las respuestas de la duda.

- Endpoint para buscar duda por nombre.
- Endpoint para buscar duda por id.
- Implementa validación a la hora de crear una duda para que se rellene todos los campos y si no se hace que devuelva un mensaje.
- Paginación de 10 en 10.
- Marcar duda como resuelta (booleano):
 - Endpoint para marcar una duda resuelta.
 - Endpoint para quitar la duda resuelta.
- Respuestas
 - Endpoint para crear una respuesta en una determinada duda.
- 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.
 - Endpoint para dar puntos (rol profesor).
- Backend disponible en producción.
- Middleware para comprobar la autoría de la duda a la hora de editar/eliminar el mismo y marcar la duda como resuelta o no.

1.2. Extras

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

- Middleware para comprobar la autoría de la respuesta a la hora de editar/eliminar la misma.
- El Endpoint que nos trae la información del usuario conectado, además que nos traiga las dudas.
- Implementa el correo de confirmación para el registro.
- Implementa que solo si tienes un correo de **@edem.es** puedas registrarte.
- Validación en el login:
 - Si no has confirmado tu correo no puedes conectarte.
- El endpoint que traiga todas las dudas junto a los usuarios que hicieron esas preguntas y junto a las respuestas de las dudas, que también traiga los usuarios que hicieron las respuestas.
- Endpoint para buscar usuario por nombre.
- Endpoint para buscar usuario por id.
- Crea una documentación de los endpoints de tu proyecto.
- Respuestas
 - CRUD respuestas
 - Votación:
 - Dar un voto a una respuesta.
 - Quitar un voto a una respuesta.

1.3. Entregables

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