

Examen módulo 2

El objetivo de este examen consiste en crear una aplicación web distribuida.

La aplicación nos permitirá registrarnos como usuarios y subir fichas con información de nuestras mascotas.

En el lado del servidor construireis una api-rest basada en nodejs y express junto con mongoDB para la capa de persistencia.

En el lado del frondEnd crearéis una aplicación con VUE capaz de consumir esta api.

Parte 1 - BACKEND

Ejercicio 1

Crea un servidor con nodejs express capaz de responder con un mensaje de ok en forma de json a los siguientes end-points. Las peticiones que reciben datos debe mostrar también los datos recibidos junto al mensaje de ok

- **Registro usuario: localhost:8000/api/registro**
 - Petición *post* donde se enviará los siguientes datos en body:
 - nombre
 - user (mail)
 - pass (contraseña de letras y números)
- **Login: localhost:8000/api/login**
 - Petición *post* donde se enviará los siguientes datos en body:
 - user (mail)
 - pass (contraseña de letras y números)
- **Mascotas: localhost:8000/api/mascotas**
 - Petición *post* donde se enviará los siguientes datos en body:
 - nombre (nombre de la mascota)
 - descripción (descripción de la mascota)
 - Petición *get* (devolverá una lista de mascotas)
- **Mascotas: localhost:8000/api/mascotas/:id**
 - Petición *get* (devolverá los datos de la mascota con ese id)
 - Petición *put*(para editar mascota) donde se enviará los siguientes datos en body:
 - nombre (nombre de la mascota)
 - descripción (descripción de la mascota)
 - Petición *delete* (borrará la mascota con ese id)
- **Usuarios: localhost:8000/api/usuarios**
 - Petición *get* (devolverá una lista de usuarios)

- **Mascotas: localhost:8000/api/usuarios/:id**
 - Petición *delete* (borrará el usuario con ese id)
- **Imágenes: localhost:8000/api/upload**
 - Petición *post* (permite subir un archivo)
- **Imágenes: localhost:8000/api/upload/:archivo**
 - Petición *delete* (borrará el archivo con nombre *archivo*)

**Verifica cada uno de los endpoints mediante POSTMAN*

Ejercicio 2

Crea una base de datos mediante mongoAtlas y conéctala a tu servidor.

Crea los controladores necesarios para que cada endpoint realice su cometido.
(la contraseña debe ser guardada en bd encriptada)

**Verifica cada uno de los endpoints mediante POSTMAN*

Ejercicio 3

Crea un sistema de autenticación basado en tokens de manera que existan dos roles, registrado y administrador. Los accesos estarán limitados con los siguientes criterios:

- Cualquier usuario puede ver la lista de mascotas
- Solo los usuarios **registrados** podrán añadir mascotas
- El usuario **administrador** podrá acceder al resto de las funcionalidades (crear, ver, editar y borrar mascotas y ver y borrar usuarios)