

Semana 2

Carlos López Lorenzo



Índice

Semana 2	1
1.Conectar la base de datos	3
2.Probar el endpoint con datos reales	4





1.Conectar la base de datos

-Sustituir los datos mockeados por datos en la base de datos:

Para esto comenzaremos creando un schéma de mongo en nuestro código con la estructura que deseemos que tengan los datos que vamos a almacenar

Este es un ejemplo que he creado en mi backend en un archivo .js que incluiré en mi código principal, con el schema creado la base de datos ya detectara o creará una tabla para este schema, luego para introducir los datos mockeados, ya que son pocos podemos introducirlos desde la base de datos manualmente, creando un método post o con el uso de plugins en caso de que sean muchos datos podemos pasarlos con el uso de código.

En mi caso he creado un método post para introducir los datos manualmente en la base de datos, con el siguiente código:

```
//ENpoint para introducir datos en la base de datos para hacer pruebas
app.post("/Personajes", async (req, res) => {
   try {
      const nuevoPersonaje = new Personaje(req.body);
      await nuevoPersonaje.save();
      res.status(201).json(nuevoPersonaje);
   } catch (error) {
      res.status(500).send("Error al agregar el Personaje");
   }
});
```





-Implementar consultas básicas como listar productos:

Para hacer la consulta usaremos el método get creado anteriormente que nos listara todos los personajes en formato json

```
app.get("/Personajes", async (req, res) => {
    try {
        const personajes = await Personaje.find();
        res.json(personajes);
    } catch (error) {
        console.error("Error al obtener los personajes:", error);
        res.status(500).json({ error: "Error al obtener los personajes" });
    }
});
```

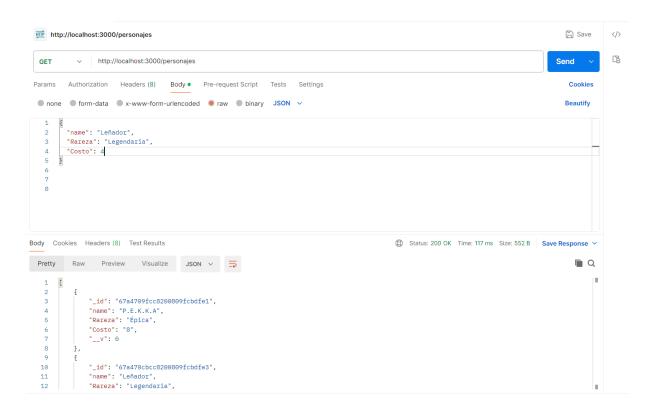
Con este podremos hacer una consulta en postman para obtener todos los personajes

2.Probar el endpoint con datos reales

Ahora añadiremos algunos datos reales con el método post y los listamos con el método get, tal que así:

```
POST http://localhost:3000/pe + ***
 http://localhost:3000/personajes
                                                                                              🖺 Save
                                                                                                         </>>
                                                                                                         http://localhost:3000/personajes
                                                                                            Send
Params Authorization Headers (8)
                               Body Pre-request Script Tests Settings
                                                                                               Cookies
 Beautify
        "name": "Leñador",
   2
        "Rareza": "Épica",
       "Costo": 8
   4
   5
Body Cookies Headers (8) Test Results
                                                                 (201 Created 90 ms 363 B Save Response >
  Pretty
                                                                                                ■ Q
   2
          "name": "P.E.K.K.A",
          "Rareza": "Épica",
   3
   4
         "Costo": "8",
          "_id": "67a4709fcc8200809fcbdfe1",
   5
```





Con estos métodos en postman podemos añadir y listar nuestros personajes.

