



Instituto Politécnico Nacional



**Unidad Profesional Interdisciplinaria de ingeniería
Campus Zacatecas**

Sistemas distribuidos

Erika Paloma Sánchez Femat

**Práctica 1: Creación de una API para gestionar datos de una
tienda en línea utilizando MongoDB**

Jesús Miguel Hernández García 2022670112

Raúl Cardoso Acevedo 2022670107

Alonso Domínguez López 2022670196

Guillermo Ordaz Rodríguez 2022670242

Salvador Trejo Martínez 2022670126

Zacatecas, Zac. 10 de septiembre de 2024

Introducción

Se desarrollará una API RESTful capaz de interactuar con una base de datos NoSQL utilizando MongoDB. La API se enfocará en manejar las operaciones básicas de una tienda en línea, tales como la creación, lectura, actualización y eliminación de productos, clientes, categorías y pedidos (operaciones CRUD: Create, Read, Update y Delete). MongoDB será la base de datos elegida debido a su flexibilidad para manejar colecciones de datos no estructurados, ideal para escenarios donde los datos pueden cambiar o crecer de manera dinámica, como en el caso de una tienda en línea.

Al finalizar la práctica, tendrás un servicio web funcional que permitirá gestionar de manera eficiente los datos de una tienda en línea, sentando las bases para un sistema escalable y flexible.

Desarrollo

Pruebas realizadas

Para el desarrollo de esta práctica a cada integrante se le asignó un apartado de la tienda en línea, cada uno se encargó de crear un endpoint para posteriormente, mediante un router conectar todo. En cada endpoint se agregaron 10 pruebas para verificar que las funciones CRUD funcionaran de manera correcta (ejemplo: en productos se agregaron 10 productos con sus respectivos precios y descripciones).

Resultados obtenidos

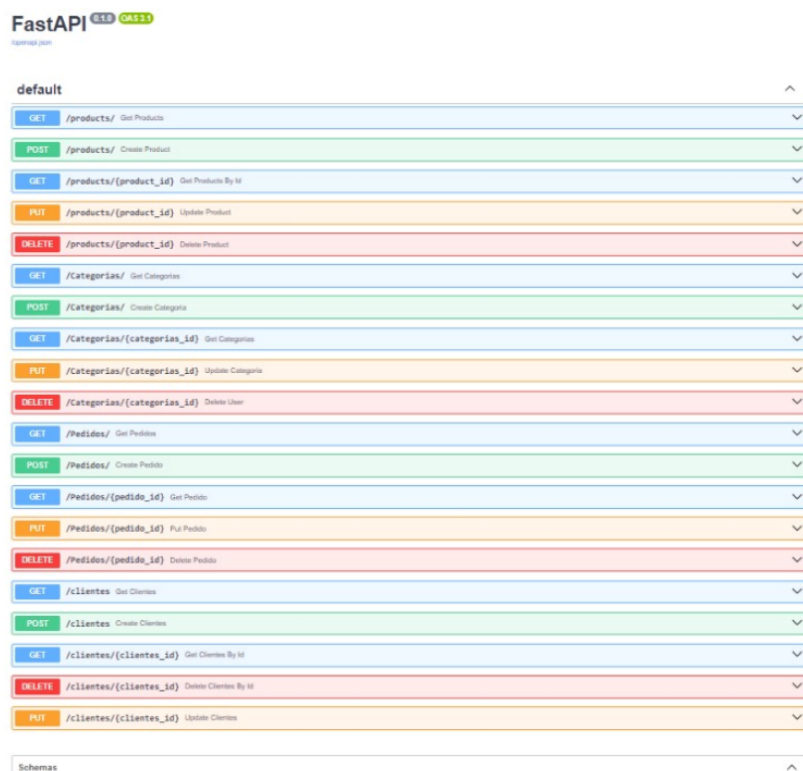


Imagen 1. Endpoints para interactuar con cada colección.

En la imagen 1 se observan todas las acciones que se pueden realizar basándonos en las 4 colecciones hechas. Se tiene para productos, categorías, pedidos y clientes.

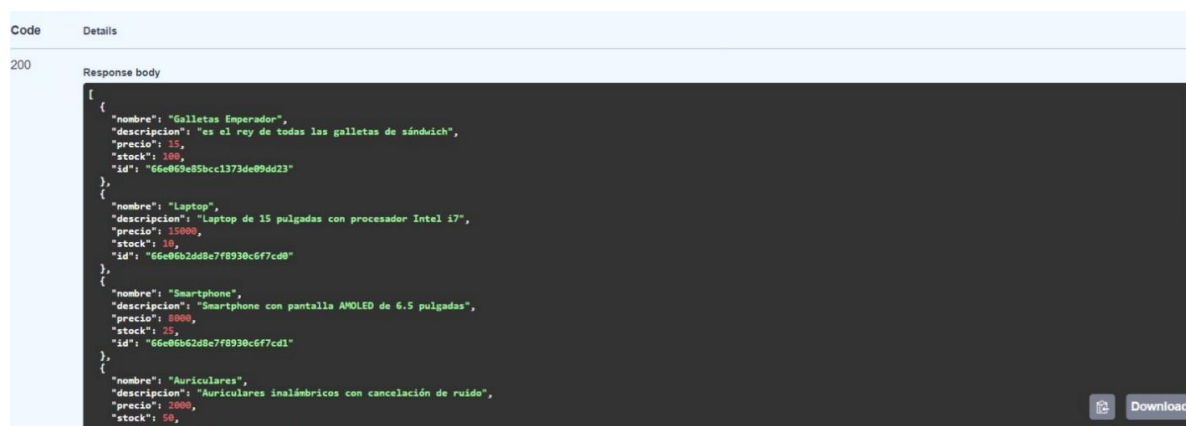


Imagen 2. GET de productos.

Obsérvese en la imagen 2 como se utiliza un GET para poder visualizar los productos registrados anteriormente.

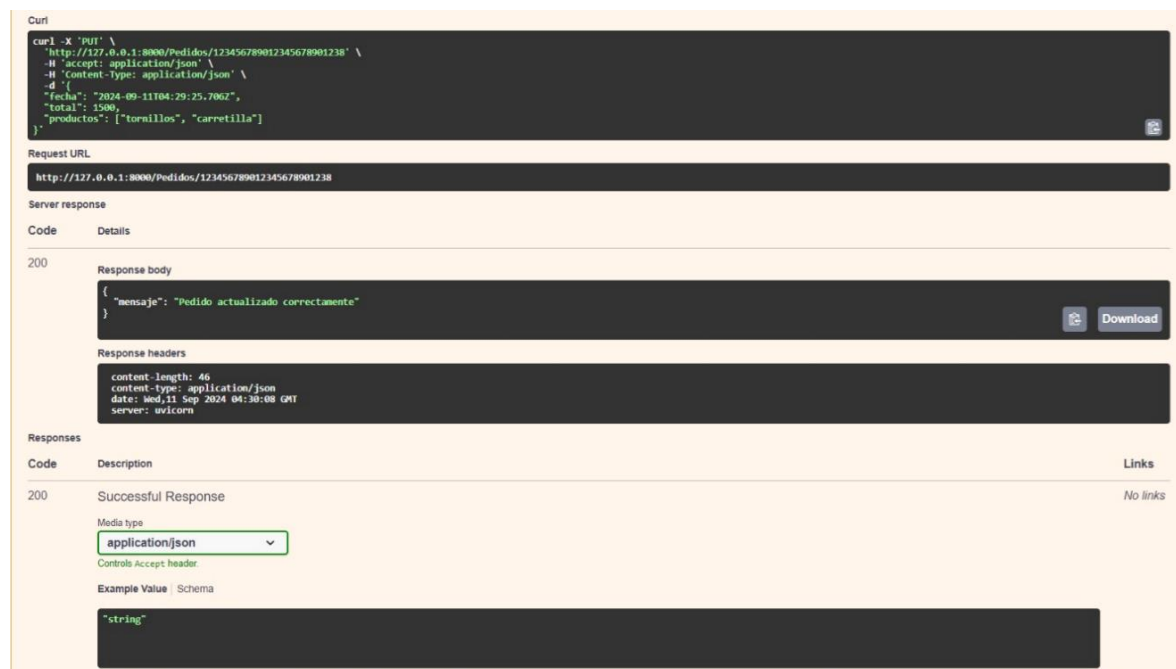


Imagen 3. PUT de pedidos.

```
_id: ObjectId('123456789012345678901238')
fecha : 2024-09-10T19:14:24.082+00:00
total : 777
▼ productos : Array (2)
  0: "plantillas"
  1: "botas"
```

Imagen 4. PUT de pedidos.

```
_id: ObjectId('123456789012345678901238')
fecha : 2024-09-11T04:29:25.706+00:00
total : 1500
▼ productos : Array (2)
  0: "tornillos"
  1: "carretilla"
```

Imagen 5. PUT de pedidos.

En la Imagen 3 se logra visualizar como se ha realizado un cambio en un pedido, en este caso, obsérvese en la imagen 4 como en primera instancia se tiene un pedido que tiene un total de 777 y los cuales son unas plantillas y unas botas. Posteriormente, obsérvese en la Imagen 5 como el pedido tiene exactamente la misma ID pero se modificó el total y los productos.

Conclusiones

Se ha logrado construir un sistema capaz de manejar eficientemente las colecciones de productos, clientes y pedidos, sentando las bases para un servicio web robusto y escalable.

Esta experiencia ha permitido comprender no solo los principios fundamentales de las APIs RESTful, sino también la integración con bases de datos NoSQL, como MongoDB, que ofrecen una gran flexibilidad y eficiencia para almacenar datos en aplicaciones web modernas. Además, se ha trabajado en la correcta estructuración de los endpoints y en el uso de los métodos HTTP correspondientes, garantizando que la API siga los estándares de buenas prácticas en el desarrollo de servicios web.