**Implementación de FastAPI con MongoDB**

**Objetivo del Reporte:**

Documentar las pruebas realizadas sobre la implementación de la API con **FastAPI** conectada a una base de datos **MongoDB**, incluyendo la definición de la conexión a la base de datos y las operaciones CRUD para manejar colecciones.

**1. Prueba de Conexión a MongoDB**

**Descripción:** Se probó la conexión a la base de datos MongoDB utilizando el método mongodb.connect\_to\_mongo() implementado en el archivo db.py. Este método utiliza la librería motor para manejar las operaciones de base de datos de manera asíncrona.

**Prueba:**

* Se ejecutó el servidor FastAPI y se verificó si la conexión con MongoDB fue exitosa.
* La base de datos utilizada fue mydatabase, y se utilizó la URI mongodb://localhost:27017.

**Resultados Obtenidos:**

* La conexión con la base de datos se realizó correctamente, y no se observaron errores en la consola.
* El cliente MongoDB (AsyncIOMotorClient) se conectó de manera estable, permitiendo que las operaciones en la base de datos se realicen sin interrupciones.

**Estado:** ✅ Conexión Exitosa

**2. Pruebas CRUD - Productos**

**a) Crear un Producto (POST /products/)**

**Descripción:** Se probó la creación de un nuevo documento en la colección products, que almacena información sobre los productos, incluyendo nombre, descripción y precio.

**Resultados Obtenidos:**

* El producto fue creado exitosamente, y se devolvió un id generado automáticamente.
* El documento fue añadido a la colección products en MongoDB sin errores.

**Estado:** ✅ Producto Creado Exitosamente

**b) Obtener Todos los Productos (GET /products/)**

**Descripción:** Se probó la obtención de todos los productos almacenados en la colección products.

**Prueba:**

* Se envió un request GET para recuperar la lista completa de productos.

**Resultados Obtenidos:**

* Se obtuvo una lista de todos los productos con sus respectivos atributos, incluido el ObjectId convertido a string.
* Los productos previamente creados se devolvieron correctamente.

**Estado:** ✅ Productos Obtenidos Correctamente

**c) Obtener Producto por ID (GET /products/{product\_id})**

**Descripción:** Se probó la obtención de un producto específico utilizando su \_id.

**Prueba:**

* Se envió un request GET con un product\_id válido.

**Resultados Obtenidos:**

* El producto con el ID especificado se devolvió correctamente.
* Si se envió un product\_id inválido, la API devolvió un error 404, como se esperaba.

**Estado:** ✅ Producto Recuperado Correctamente

**d) Actualizar un Producto (PUT /products/{product\_id})**

**Descripción:** Se probó la actualización de un producto existente mediante su \_id.

**Resultados Obtenidos:**

* El producto fue actualizado correctamente en la base de datos.
* La API devolvió un mensaje de confirmación indicando que la actualización fue exitosa.

**Estado:** ✅ Producto Actualizado Correctamente

**e) Eliminar un Producto (DELETE /products/{product\_id})**

**Descripción:** Se probó la eliminación de un producto utilizando su \_id.

**Prueba:**

* Se envió un request DELETE con un product\_id válido.

**Resultados Obtenidos:**

* El producto fue eliminado de la colección products correctamente.
* Si el product\_id no existía, la API devolvió un error 404.

**Estado:** ✅ Producto Eliminado Correctamente

**3. Pruebas CRUD - Clientes, Categorías y Pedidos**

Para las colecciones clients, categories, y pedidos, se replicaron pruebas similares a las de products, siguiendo la misma estructura para verificar el funcionamiento correcto de las rutas y las operaciones CRUD. A continuación, un resumen de los resultados:

**a) Crear, Obtener, Actualizar, y Eliminar Clientes**

* **Pruebas:** Se crearon, leyeron, actualizaron y eliminaron clientes con éxito.
* **Resultados:** Todas las operaciones CRUD funcionaron correctamente en la colección clients.

**Estado:** ✅ Operaciones CRUD con Clientes Correctas

**b) Crear, Obtener, Actualizar, y Eliminar Categorías**

* **Pruebas:** Se verificó la correcta manipulación de las categorías de productos.
* **Resultados:** Las operaciones CRUD en la colección categories fueron satisfactorias.

**Estado:** ✅ Operaciones CRUD con Categorías Correctas

**c) Crear, Obtener, Actualizar, y Eliminar Pedidos**

* **Pruebas:** Se crearon pedidos asociados a clientes, se recuperaron y actualizaron pedidos existentes, y se eliminaron pedidos de prueba.
* **Resultados:** Todas las pruebas CRUD funcionaron correctamente para la colección pedidos.

**Estado:** ✅ Operaciones CRUD con Pedidos Correctas

**Conclusiones del Reporte**

1. **Conexión a MongoDB:** Se estableció correctamente, y las operaciones de la base de datos se ejecutaron de forma estable y eficiente.
2. **Operaciones CRUD:** Las rutas implementadas para manejar las colecciones (products, clients, categories, pedidos) funcionaron correctamente en todas las pruebas realizadas.
3. **Rendimiento:** Las operaciones fueron asíncronas y no se observaron problemas de rendimiento ni latencia significativa en las pruebas de inserción, actualización y lectura de datos.

**Recomendaciones:**

* Continuar con la validación y pruebas automatizadas para manejar escenarios de error, como documentos inexistentes o entradas inválidas.
* Optimizar la validación de datos entrantes en las rutas de creación y actualización.