Programa: Desarrollador Junior

Prueba Programada: Desarrollo con Python, MongoDB, API y WebSockets

Leonel Antonio Gonzalez Garcia

Instrucciones:

Resuelve los siguientes ejercicios y responde las preguntas teóricas relacionadas.

Ejercicio 1: Integración de Python con MongoDB

Descripción:

Desarrolla un programa en Python que conecte a una base de datos en MongoDB, inserte datos de una colección llamada clientes, y luego realice una consulta que filtre clientes por ciudad.

Requisitos:

Crea una conexión a MongoDB local o en la nube (MongoDB Atlas).

Inserta al menos 5 documentos en la colección clientes. Cada documento debe contener las siguientes propiedades:

- nombre: string
- edad: int
- ciudad: string

Realiza una consulta que devuelva todos los clientes que vivan en una ciudad específica ingresada por el usuario.

Puntos extra:

Implemente la eliminación de un cliente utilizando su nombre.

Ejercicio 2: API REST con Node.js y Postman

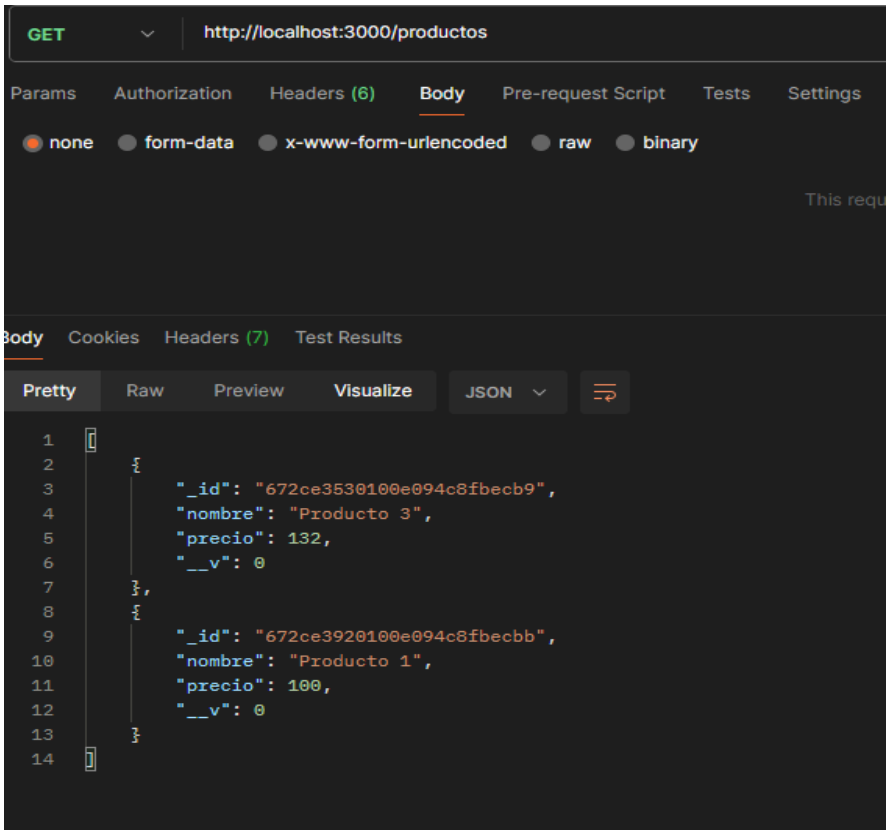
Descripción:

Crea un API REST utilizando Node.js que permita realizar operaciones CRUD sobre una colección productos en una base de datos MongoDB.

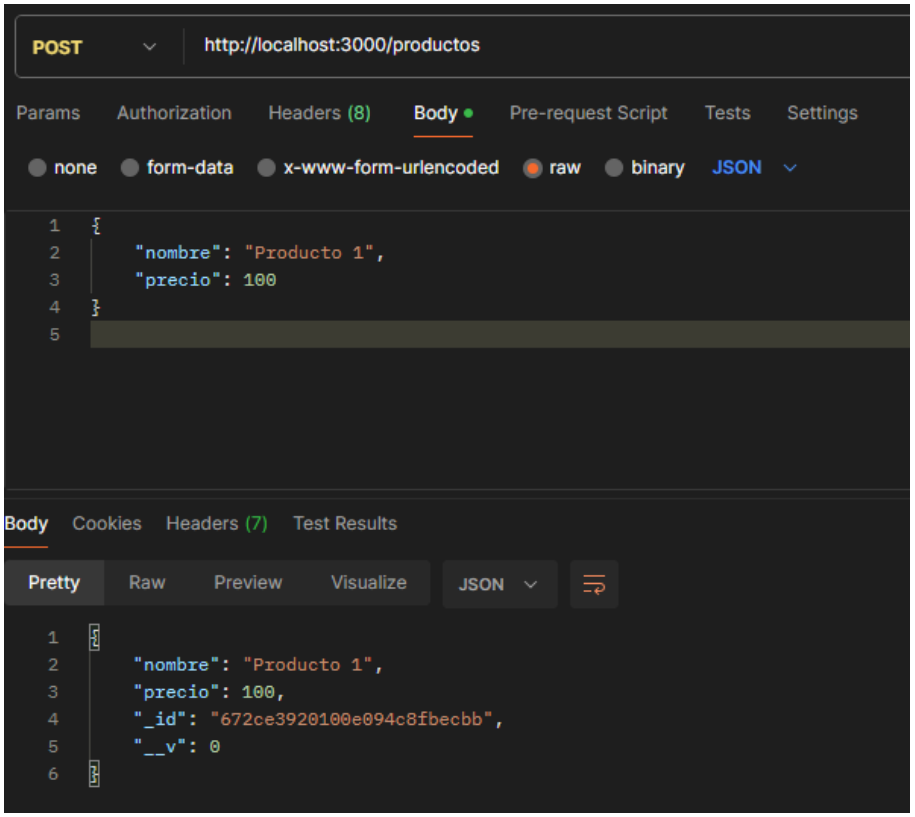
Requisitos:

El API debe tener los siguientes endpoints:

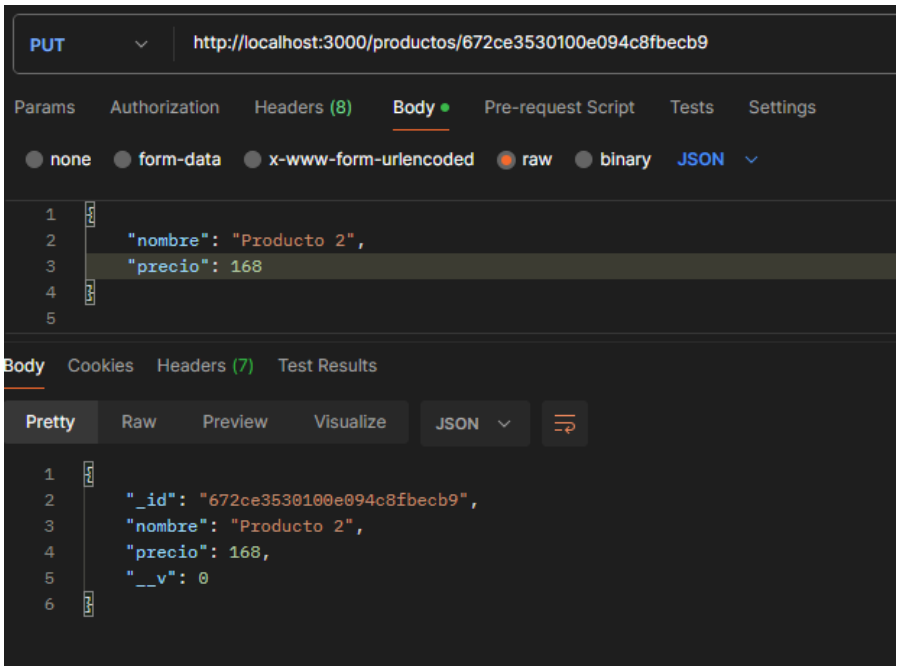
- GET /productos: Devuelve todos los productos.



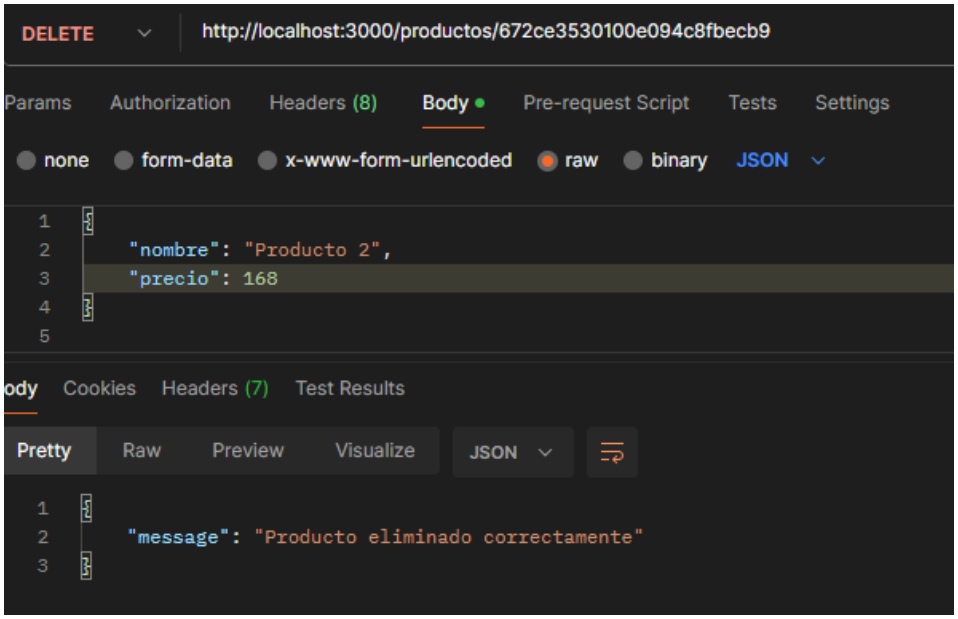
- POST /productos: Agrega un nuevo producto.



- PUT /productos/:id: Actualiza un producto por su ID.



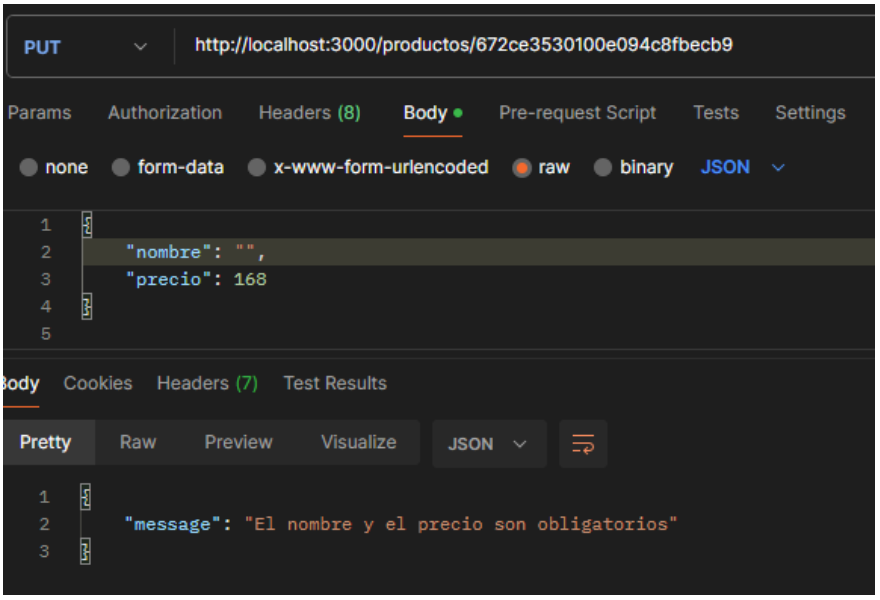
- DELETE /productos/:id: Elimina un producto por su ID.



Prueba los endpoints usando Postman y muestra capturas de las solicitudes y respuestas.

Puntos extra:

Incluye validaciones para que los campos nombre y precio no estén vacíos.



Ejercicio 3: WebSocket que consume un API externo

Programa: Desarrollador Junior

Descripción:

Desarrolla una aplicación con Node.js que utilice WebSockets para recibir datos en tiempo real de un API externo.

Requisitos:

- Conéctate a un WebSocket público, como wss://api-pub.bitfinex.com/ws/2, para recibir datos de precios de criptomonedas.
- Procesa los datos recibidos y muestra en consola los precios de una criptomoneda específica (por ejemplo, BTC/USD).
- Envía al cliente conectado un mensaje con los datos procesados cada 5 segundos.

Puntos extra:

Permite al cliente suscribirse a la criptomoneda que desea recibir.

Preguntas Teóricas

Responde las siguientes preguntas:

- ¿Cuál es la ventaja de utilizar MongoDB en lugar de una base de datos relacional como MySQL?
  Entre las ventajas que nos proporciona MongoDB están su facilidad para adaptarse a cambios en los datos sin la necesidad de mantener un esquema fijo, también esta su velocidad para el manejo de datos en múltiples servidores que sirve bastante para el manejo de grandes volúmenes de datos en aplicaciones que tienen un crecimiento constante en sus datos lo que también nos lleva a que sea mucho más rápido en la lectura y escritura de los datos también gracias a que se base en documentos de tipo JSON.

- Describe cómo Postman puede facilitar el desarrollo y prueba de un API.
  Postman facilita la prueba de APIs ya que permite enviar solicitudes HTTP de manera rápida y sencilla sin la necesidad de escribir código adicional, también ayuda con la simulación de datos y la depuración mediante registros.
- **Explica el ciclo de vida de una conexión WebSocket y cómo se diferencia de una solicitud HTTP estándar.**
- ¿Cómo manejarías errores en un API REST para que los usuarios reciban respuestas adecuadas?
  El manejar de forma correcta los errores en una API Rest sirve para brindarle al usuario una retroalimentación clara del porqué fallaron sus solicitudes y la forma más rápida de resolverlos. Una de las claves en el manejo de errores es utilizar códigos HTTP ya que brindan información de los errores de forma estandarizada y segura evitando que se brinde más información de la necesaria.
- ¿Qué beneficios tiene usar WebSockets frente a otras tecnologías para recibir datos en tiempo real?

El uso de Websockets para trabajar con aplicaciones en las que se requiere una comunicacion continua y en ambas direcciones entre un cliente y el servidor brinda varios beneficios como lo pueden ser la comunicacion continua que permite enviar y recibir informacion sin la necesidad de reiniciar la conexion cada vez, tambien brinda una menor latencia en el manejo de datos.

Puntaje Total: 100 puntos

- Ejercicio 1: 30 puntos
- Ejercicio 2: 30 puntos
- Ejercicio 3: 30 puntos
- Preguntas teóricas: 10 puntos (2 puntos por cada pregunta)