

# **Evidencia de Producto: GA7-220501096-AA5-EV04**

## **API del proyecto Trading Bot**

### **Portada**

**Nombre: Carlos Alvarez**

**Ficha: (2977356)**

**Proyecto: GA7-220501096-AA5-EV04**

**Actividad: Diseño, Desarrollo y Testing de Servicios Web**

**Fecha: 26 de noviembre de 2025**

### **1. Introducción**

Este documento describe el proceso de testing y verificación funcional de la API desarrollada para el proyecto de trading automatizado, usando la herramienta Postman. El objetivo es evidenciar el correcto funcionamiento de los servicios web (endpoints) desarrollados con Spring Boot y comprobar, mediante pruebas, que cumplen los requisitos propuestos.

### **2. Objetivos**

- Validar los principales endpoints implementados en el backend mediante pruebas en Postman.**
- Documentar los resultados obtenidos, incluyendo ejemplos de request/response y pantallazos.**
- Demostrar la correcta integración con la base de datos y la lógica de negocio expuesta por la API REST.**

### 3. Endpoints principales

A continuación, se listan los principales endpoints probados. Se documentan el método HTTP, la URL, su funcionalidad y ejemplos de los datos utilizados en las pruebas.

#### a) Crear indicador EMA

Método: POST

URL :<http://localhost:8080/api/indicators/ema>

Cuerpo (JSON):

```
{  
  "asset": "XAUUSD",  
  "periods": 30,  
  "timeframe": "1h"  
}
```

Respuesta esperada:

```
{  
  "id": 1,  
  "asset": "XAUUSD",  
  "ema30": 1910.78,  
  "timestamp": "2025-11-24T12:10:09"  
}
```

#### b) Crear operación de trading

- Método: POST
- URL :<http://localhost:8080/api/trades/create>
- Cuerpo (JSON):

```
{  
  "asset": "XAUUSD",  
  "type": "BUY",  
  "entryPrice": 1910.00,  
  "quantity": 2,  
  "indicators": "EMA,SAR,Fibonacci"  
}
```

Respuesta esperada:

```
{  
  "id": 1,  
  "asset": "XAUUSD",  
  "type": "BUY",  
  "entryPrice": 1910.00,  
  "exitPrice": null,  
  "quantity": 2,  
  "status": "OPEN",  
  "entryTime": "2025-11-24T12:12:30",  
  "indicators": "EMA,SAR,Fibonacci",  
  "profitLoss": null  
}
```

**c) Cerrar el comercio**

Método: PUT

URL :<http://localhost:8080/api/trades/{id}/close?exitPrice=1932.00>

Respuesta esperada:

```
{  
    "id": 1,  
    "asset": "XAUUSD",  
    "type": "BUY",  
    "entryPrice": 1910.00,  
    "exitPrice": 1932.00,  
    "quantity": 2,  
    "status": "CLOSED",  
    "entryTime": "2025-11-24T12:12:30",  
    "exitTime": "2025-11-24T12:15:43",  
    "indicators": "EMA,SAR,Fibonacci",  
    "profitLoss": 44.0  
}
```

**d) Consultar indicadores**

Método: GET

URL :<http://localhost:8080/api/indicators/XAUUSD>

Respuesta ejemplo:

```
[  
    {  
        "id": 1,  
        "asset": "XAUUSD",  
        "ema30": 1910.78,  
        "sar": null,  
        "fibonacciLevel": null,  
        "timestamp": "2025-11-24T12:10:09"  
    }  
]
```

## 4. Testing en Postman y evidencia visual

Capturas de pantalla:

Pruebas de cada endpoint en Postman (body, response).

Configuración de headers y parámetros.

Ejemplos de respuestas exitosas y respuestas de error (si aplica).

The image contains three separate screenshots of the Postman application interface, each showing a successful API request. The first screenshot shows a request to 'http://localhost:8080/api/indicators/ema' with a JSON body containing asset: 'eurusd', periods: 30, and timeframe: '1h'. The second screenshot shows a request to 'http://localhost:8080/api/indicators/sar' with a JSON body containing asset: 'string', initialAF: 0, maxAF: 0, and timeframe: 'string'. Both of these requests resulted in a 200 OK response with an empty JSON body. The third screenshot shows a request to 'http://localhost:8080/api/indicators/fibonacci' with a JSON body containing asset: 'string', highPrice: 0, lowPrice: 0, and timeframe: 'string'. This request also resulted in a 200 OK response with an empty JSON body. Each screenshot includes the Postman header bar, the request URL and method in the main input field, the JSON body in the 'Body' tab, and the response details in the bottom right corner.

## 5. Conclusiones

Las pruebas realizadas en Postman demuestran que la API responde correctamente a los diferentes escenarios planteados, procesando los datos enviados y entregando respuestas en el formato esperado. Gracias a estos resultados, se puede afirmar que la API cumple con los requisitos del proyecto GA7-220501096-AA5-EV04 y está lista para integración o despliegue.

## 6. Anexos

Video del proceso de testing (adjunto en la entrega ZIP).

Archivo endpoints.txt (con todos los detalles de cada endpoint).

Enlace al repositorio del proyecto (si aplica).