# Prueba de Concepto (PoC):

# Testing de APIs con Docker, Postman, Newman y WordPress

## ¿Qué es una API?

Una API (Application Programming Interface) es una interfaz que permite la comunicación entre dos sistemas. Las APIs REST utilizan HTTP para exponer recursos que pueden ser consumidos mediante métodos como GET, POST, PUT y DELETE. Las APIs son ampliamente utilizadas para permitir la interoperabilidad entre servicios distribuidos y son claves en la arquitectura moderna de software.

## Objetivo del Proyecto

Construir una prueba de concepto automatizada y contenerizada que permita probar múltiples APIs REST de forma visual (WordPress) y automatizada (Newman), usando Docker para orquestar todo el entorno.

## Herramientas Utilizadas

- Docker  
- Docker Compose  
- Postman  
- Newman  
- WordPress  
- Visual Studio Code  
- PowerShell

## Estructura del Proyecto

api-testing-poc/  
├-- docker-compose.yml  
└── newman/  
 ├-- Dockerfile  
 ├-- coleccion\_catfacts\_poc.json  
 ├--.dockerignore

## Paso a Paso Detallado para la implementación

### 1. Se debe crear la estructura de la s carpetas

mkdir api-testing-poc  
cd api-testing-poc  
mkdir newman

### 2. Se debe crear el archivo docker-compose.yml

version: '3.9'  
services:  
 wordpress:  
 image: wordpress:latest  
 ports:  
 - "8080:80"  
 environment:  
 WORDPRESS\_DB\_HOST: db:3306  
 WORDPRESS\_DB\_USER: root  
 WORDPRESS\_DB\_PASSWORD: root  
 WORDPRESS\_DB\_NAME: wp\_db  
 volumes:  
 - wp\_data:/var/www/html  
 depends\_on:  
 - db  
  
 db:  
 image: mysql:5.7  
 environment:  
 MYSQL\_DATABASE: wp\_db  
 MYSQL\_ROOT\_PASSWORD: root  
 volumes:  
 - db\_data:/var/lib/mysql  
  
 newman:  
 build: ./newman  
 depends\_on:  
 - wordpress  
 command: ["newman", "run", "coleccion\_catfacts\_poc.json"]  
  
volumes:  
 wp\_data:  
 db\_data:

### 3. Se debe crear el Dockerfile para Newman

FROM node:18  
RUN npm install -g newman  
WORKDIR /app  
COPY coleccion\_catfacts\_poc.json .  
CMD ["newman", "run", "coleccion\_catfacts\_poc.json"]

### 4. Creamos el Archivo .dockerignore

node\_modules  
npm-debug.log

### 5. Crear la colección de Postman

Diseñar pruebas en Postman y exportarlas como JSON con un nombre como: coleccion\_catfacts\_poc.json. El mismo se debe guardar en /newman.

### 6. Crear página en WordPress para probar APIs

1. Ir a http://localhost:8080 una vez que el entorno esté levantado.

2. Crear el código en html para visualizar de manera grafica las pruebas.  
3. Crear una nueva página desde el panel de WordPress.  
4. Insertar un bloque de HTML personalizado.  
5. Pegar el código HTML con fetch() para consumir las APIs y mostrar resultados dinámicamente.  
6. Usar JavaScript dentro de <script> para hacer llamadas HTTP desde la interfaz.

### 7. Levantar todo el entorno con el siguiente comando dentro de la terminal

docker-compose up --build

Este comando construye la imagen personalizada de Newman y levanta los servicios de WordPress y MySQL. Newman ejecuta automáticamente la colección al iniciar.

### 8. Validar manualmente en consola (PowerShell)

Invoke-RestMethod -Uri "https://jsonplaceholder.typicode.com/posts/1" -Method GET  
Invoke-RestMethod -Uri "https://catfact.ninja/fact" -Method GET

### 9. Comando para detener los servicios

docker-compose down

## Comunicación entre servicios

Todos los servicios definidos en docker-compose están conectados en una red Docker interna. WordPress puede comunicarse con MySQL usando el nombre del servicio 'db'. Newman no necesita interactuar directamente con WordPress, ya que solo ejecuta pruebas sobre APIs externas. WordPress muestra resultados directamente al usuario mediante JavaScript. PowerShell permite hacer validaciones adicionales fuera del entorno Docker.

## Conclusión

Esta PoC integra herramientas modernas para pruebas de APIs con un enfoque educativo y técnico. Permite a los usuarios validar, automatizar y visualizar pruebas de servicios REST, todo contenido en un entorno Docker fácilmente replicable.