

Ejercicio Práctico - Servidor Web en Node (JSON) + Cliente REST: Gestión de Productos

Actividad- Servidor Web en Node (JSON) + Cliente REST: Gestión de Productos

Introducción / Objetivo

Actividad breve para implementar un **servidor web en Node.js** que responda en **JSON** a solicitudes **GET/POST** y rechace otros métodos con el **código adecuado**, leyendo datos de forma **no bloqueante** desde el sistema de archivos. Además, se desarrollará un **cliente web** que consuma la API, liste y ordene productos, y permita agregar nuevos.

Descripción de la actividad (tal cual el enunciado)

1. Servidor web en Node

Crear un servidor web que responda a peticiones en formato **JSON**. Solo responderá correctamente a **GET** y **POST**; cualquier otro método será **rechazado** (con un **código de estado adecuado**).

a. GET

La respuesta a un GET debe ser obtenida de un **archivo local** desde el **file system** de forma **non-blocking**. Cada línea del archivo representa un producto con el formato: nombre, precio

Ejemplo de archivo con 2 productos:

Zapatillas, 25990

Jeans, 12990

b. POST

El método **POST** insertará un **nuevo producto**, el cual debe venir en el **body** del request en **formato JSON**.

2. Cliente web (API REST del ejercicio anterior)

Crear un cliente web que utilice la API REST desarrollada:

- Botón **"LISTA DE ARTICULOS"** que muestre **ordenadamente** el resultado en el **DOM**.
- La lista puede ser **reordenada** por **nombre** o por **precio**.
- Formulario con **2 campos** (nombre y precio) y botón **"AGREGAR NUEVO PRODUCTO"** que realice el **POST**.
- Incluir una **imagen** ilustrativa del producto.

Sugerencia mínima (opcional)

- Responde con cabeceras adecuadas: Content-Type: application/json.
- Usa lectura **no bloqueante** del archivo (p. ej., API de **fs.promises**).
- Códigos recomendados: **200** (OK), **201** (creado en POST), **400** (JSON inválido), **405** (método no permitido).

Forma de entrega

Sube un **.zip** con:

- **Backend** (código del servidor Node).
- **Frontend** (HTML/CSS/JS).
- Archivo de datos (ej.: **productos.txt**) y **una imagen** de ejemplo.
- **Evidencias**: pantallazos de la lista, del ordenamiento y de un POST exitoso.