**app.js → Ejercicios 1 al 5 (servidor, rutas, parámetros, middleware)**

**// app.js**

const express = require('express');

const app = express();

const PORT = 8080;

**// Middleware logger (5.1)**

app.use((req, res, next) => {

const now = new Date().toLocaleString();

console.log(`[${now}] ${req.method} ${req.url}`);

next();

});

**// Ruta raíz (1.1)**

app.get('/', (req, res) => {

res.send('¡Mi primer servidor con Express!');

});

**// 1.2 - Welcome (correr por separado en welcome.js)**

**/// Omitido en este archivo**

**// welcome.js**

const express = require('express');

const app = express();

app.get('/', (req, res) => {

res.send('<h1>Bienvenido a mi sitio web</h1>');

});

app.listen(4000, () => {

console.log('Servidor de bienvenida corriendo en http://localhost:4000');

});

**// 2.1 - Múltiples páginas**

app.get('/nosotros', (req, res) => {

res.send('Acerca de Nosotros');

});

app.get('/contacto', (req, res) => {

res.send('Página de Contacto');

});

**// 2.2 - Tienda online**

app.get('/tienda', (req, res) => res.send('Bienvenido a MiTienda'));

app.get('/tienda/productos', (req, res) => res.send('Listado de todos nuestros productos'));

app.get('/tienda/categorias', (req, res) => res.send('Categorías disponibles: Electrónica, Ropa, Hogar'));

app.get('/tienda/ofertas', (req, res) => res.send('¡No te pierdas nuestras ofertas especiales!'));

**// 2.3 - Ruta 404**

app.use((req, res) => {

res.status(404).send('Error 404: La página que buscas no existe.');

});

**// 3.1 - Saludo personalizado**

app.get('/saludo/:nombre', (req, res) => {

res.send(`¡Hola, ${req.params.nombre}! Bienvenido.`);

});

**// 3.2 - Detalle producto**

app.get('/productos/:idProducto', (req, res) => {

res.send(`Información del producto con ID: ${req.params.idProducto}`);

});

**// 3.3 - Suma**

app.get('/calcular/sumar/:num1/:num2', (req, res) => {

const n1 = parseFloat(req.params.num1);

const n2 = parseFloat(req.params.num2);

const resultado = n1 + n2;

res.send(`Resultado de la suma: ${resultado}`);

});

**// 3.4 - Ruta con dos parámetros**

app.get('/productos/:categoria/:subcategoria', (req, res) => {

const { categoria, subcategoria } = req.params;

res.send(`Mostrando productos de la categoría ${categoria}, subcategoría ${subcategoria}`);

});

**// 4.1 - Query simple**

app.get('/buscar', (req, res) => {

const q = req.query.q;

if (q) {

res.send(`Resultados de búsqueda para: ${q}`);

} else {

res.send('Por favor, ingresa un término de búsqueda');

}

});

**// 4.2 - Filtrado avanzado**

app.get('/productos', (req, res) => {

const { categoria, precio\_max } = req.query;

if (categoria && precio\_max) {

res.send(`Filtrando por categoría ${categoria} y precio máximo ${precio\_max}`);

} else if (categoria) {

res.send(`Filtrando por categoría: ${categoria}`);

} else if (precio\_max) {

res.send(`Filtrando por precio máximo: ${precio\_max}`);

} else {

res.send('Mostrando todos los productos');

}

});

**// 5.2 - Middleware de autenticación falso**

const autenticacion = (req, res, next) => {

const apiKey = req.query.apiKey;

if (apiKey === '12345') {

next();

} else {

res.status(401).send('Acceso denegado. Se requiere una API Key válida.');

}

};

app.get('/admin', autenticacion, (req, res) => {

res.send('Área administrativa segura.');

});

**// Iniciar servidor**

app.listen(PORT, () => {

console.log(`Servidor corriendo en http://localhost:${PORT}`);

});

**api-productos.js → Sección 6: JSON / API de productos**

**// api-productos.js**

const express = require('express');

const app = express();

app.use(express.json());

const productos = [

{ id: 1, nombre: 'Laptop' },

{ id: 2, nombre: 'Celular' },

];

**// 6.1 - Lista completa**

app.get('/api/productos', (req, res) => {

res.json(productos);

});

**// 6.2 - Buscar producto por ID**

app.get('/api/productos/:id', (req, res) => {

const id = parseInt(req.params.id);

const producto = productos.find(p => p.id === id);

if (producto) {

res.json(producto);

} else {

res.status(404).json({ error: 'Producto no encontrado' });

}

});

**// 6.3 - Crear producto**

let siguienteId = 3;

app.post('/api/productos', (req, res) => {

const nuevoProducto = {

id: siguienteId++,

nombre: req.body.nombre

};

productos.push(nuevoProducto);

res.status(201).json(nuevoProducto);

});

app.listen(3001, () => {

console.log('API de productos corriendo en http://localhost:3001');

});

**api.js → Sección 7: CRUD para Libros**

**// api.js**

const express = require('express');

const app = express();

app.use(express.json());

let libros = [

{ id: 1, titulo: 'El Principito', autor: 'Antoine de Saint-Exupéry' },

{ id: 2, titulo: 'Cien años de soledad', autor: 'Gabriel García Márquez' }

];

let nextId = 3;

**// GET todos**

app.get('/libros', (req, res) => res.json(libros));

**// GET por ID**

app.get('/libros/:id', (req, res) => {

const libro = libros.find(l => l.id == req.params.id);

libro ? res.json(libro) : res.status(404).json({ error: 'Libro no encontrado' });

});

**// POST nuevo libro**

app.post('/libros', (req, res) => {

const nuevo = { id: nextId++, ...req.body };

libros.push(nuevo);

res.status(201).json(nuevo);

});

**// PUT actualizar**

app.put('/libros/:id', (req, res) => {

const index = libros.findIndex(l => l.id == req.params.id);

if (index !== -1) {

libros[index] = { ...libros[index], ...req.body };

res.json(libros[index]);

} else {

res.status(404).json({ error: 'Libro no encontrado' });

}

});

**// DELETE libro**

app.delete('/libros/:id', (req, res) => {

const index = libros.findIndex(l => l.id == req.params.id);

if (index !== -1) {

const eliminado = libros.splice(index, 1)[0];

res.json(eliminado);

} else {

res.status(404).json({ error: 'Libro no encontrado' });

}

});

app.listen(3002, () => {

console.log('API de libros corriendo en http://localhost:3002');

});