

## Express.js

### ¿Qué es?



Express.js es un framework minimalista y flexible para Node.js que permite construir aplicaciones web y APIs de manera rápida, estructurada y eficiente. Está escrito en JavaScript y corre del lado del servidor, lo que lo convierte en una herramienta ideal para desarrolladores que ya dominan este lenguaje en el frontend. Express se basa en el principio de mantener las cosas simples pero extensibles: ofrece una estructura ligera que puedes ampliar con módulos según tus necesidades.

Su diseño está inspirado en frameworks como Sinatra (Ruby), lo que significa que prioriza la claridad y la modularidad. Es parte fundamental del stack MEVN/MERN (MongoDB, Express, Vue/React, Node), y se ha convertido en el estándar de facto para el desarrollo backend con JavaScript.

### ¿Para qué se usa?

Simplifica muchas de las tareas que normalmente requerirían mucho código en Node.js puro. Por ejemplo, te permite definir rutas HTTP (GET, POST, PUT, DELETE) de forma clara, manejar solicitudes y respuestas, servir archivos estáticos, y usar middleware para interceptar y modificar el flujo de datos. También facilita la integración con motores de plantillas como EJS o Pug para generar vistas dinámicas, y con bases de datos como MongoDB o PostgreSQL para persistencia de datos.

Además, Express permite crear APIs RESTful con facilidad, lo que lo hace perfecto para aplicaciones móviles, sistemas distribuidos, y microservicios. Su arquitectura basada en middleware te permite agregar funcionalidades como autenticación, validación, logging o compresión sin complicar el código principal.

### ¿Dónde se usa?

- APIs RESTful que sirven datos a clientes web o móviles.
- Aplicaciones de una sola página (SPA) que se comunican con el backend mediante AJAX o fetch.
- Sistemas de autenticación con JWT, OAuth o sesiones.
- Dashboards en tiempo real usando WebSockets (por ejemplo, con Socket.IO).
- Microservicios que forman parte de arquitecturas distribuidas.
- Automatización de tareas y herramientas CLI que interactúan con servicios de red o hardware.

## Herramientas y Características

- **Sistema de enrutamiento:** Define rutas dinámicas y estáticas con facilidad.
- **Middleware:** Permite interceptar peticiones para validación, autenticación, logging, etc.
- **Gestión de solicitudes/respuestas HTTP:** Simplifica el manejo de headers, parámetros, cuerpos y estados.
- **Servir archivos estáticos:** Ideal para entregar HTML, CSS, JS o imágenes desde una carpeta pública.
- **Motores de plantillas:** Compatible con EJS, Pug, Handlebars para generar vistas dinámicas.
- **Integración con bases de datos:** Funciona bien con Mongoose (MongoDB), Sequelize (SQL), entre otros.
- **Seguridad y rendimiento:** Se puede complementar con Helmet, CORS, rate-limiters y compresión.
- **Facilidad de pruebas:** Compatible con herramientas como Mocha, Jest y Supertest para testing automatizado.
- **Modularidad:** Puedes dividir tu aplicación en routers y controladores para mantener el código limpio.