

Node.JS

Nacido de la resolución a la limitación para los desarrolladores el no poder trabajar en interacciones web desde frontend, que los llevaba a tener que aprender y mantener más lenguajes diferentes, bloqueos en la ejecución de entrada y salida que trabajan con modelos multi-hilo (memoria saturada, bajo rendimiento) o la falta de un ecosistema moderno y flexible.



Node.JS, proveniente de JavaScript, es un entorno de ejecución orientado a objetos, no bloqueante y basado en el motor V8 de Google Chrome, con el objetivo de permitir la ejecución de código JavaScript en el lado del servidor, ampliando las capacidades del lenguaje más allá del navegador.

Esta tecnología usa un modelo asíncrono y dirigido por eventos, convirtiéndose en la herramienta ideal para aplicaciones escalables y de alto rendimiento. Su funcionamiento en general parte del Motor V8, que ejecuta el código JavaScript de forma muy rápida, ya que lo convierte en código máquina. Permite que el servidor maneje múltiples solicitudes simultáneamente si se bloquea gracias al mecanismo Event Loop. Incluye una gran cantidad de módulos para manejar archivos, redes, procesos, etc. Además de un sistema de gestión de paquetes importante, NPM (Node Package Manager), que facilita la instalación de librerías y frameworks.

Para nuestro proyecto es ideal para manejar multiples conexiones simultáneamente y un alto rendimiento gracias a su arquitectura no bloqueante, perfecto para aplicaciones web que deben atender varios usuarios. Nos permitirá usar JavaScript tanto en frontend como en backend que simplificará el desarrollo. Su amplia comunidad y el ecosistema de miles de librerías disponibles para acelerar el trabajo y el uso de APIs eficientes que nos facilitara con la creación de servicios backend para ser consumidos desde aplicaciones web o móviles.

¿Cómo empezar a usar Node.js?

1. **Descarga Node.js desde su pagina oficial:** <https://nodejs.org/es> (incluye npm)
2. **Inicialización del proyecto:** en la carpeta del proyecto ➡ `npm init -y`, esto crea un archivo `package.json` donde se guardará dependencias y configuraciones.
3. **Instalación de dependencias:** instalar librerias necesarias, por ej: `npm install express`
4. **Creacion de servidor:** escribir un archivo (ej: `server.js`) donde se programa el servidor con Node.JS o con Frameworks como Express
5. **Ejecucion del servidor:** correr proyecto con `node server.js`, el servidor escucha solicitudes en un puerto (ej: `http://localhost:3000`)

6. **Uso de herramientas adicionales:** se pueden instalar utilidades como *nodemon* para reiniciar el servidor automáticamente durante el desarrollo. (ej: *nodemon server.js*)

Ejemplo de uso mínimo

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.end('Hola, este es un servidor con Node.js');
});

server.listen(3000, () => {
  console.log('Servidor escuchando en http://localhost:3000');
});
```