

# 01 NodeJS: ¿Qué es Node.js?

Hasta ahora en el curso, hemos utilizado JavaScript principalmente desde la perspectiva del navegador web. Hemos manipulado el DOM para crear páginas interactivas, responder a eventos del usuario y a usar la consola del navegador como una herramienta para depurar el resultado del código.

Ahora, vamos a utilizar JavaScript fuera del navegador. Aquí es donde entra en juego **Node.js**.

**Node.js es un entorno de ejecución de JavaScript del lado del servidor.**

Desglosemos esto:

1. **Entorno de Ejecución (Runtime Environment):** Dentro del navegador (como Chrome, Firefox, etc.), hay un motor (como V8 en Chrome) que entiende y ejecuta vuestro código JavaScript. Node.js es, en esencia, un programa que instalamos en nuestro ordenador (servidor, portátil, etc.) que también contiene ese motor V8 de Google. Su propósito es el mismo, ejecutar código JavaScript. Hasta ahora, vuestro JS vivía y moría dentro de una pestaña del navegador. Node.js nos permite tomar ese mismo lenguaje JavaScript y usarlo para tareas completamente diferentes, directamente en el sistema operativo.
2. **Del Lado del Servidor:** Aunque su uso más popular es para construir el *backend* de aplicaciones web (servidores que gestionan bases de datos, lógica de negocio, APIs), Node.js es mucho más versátil. Se usa para:
  - **Crear servidores web y APIs RESTful:** La columna vertebral de muchas aplicaciones web modernas.
  - **Herramientas de línea de comandos (CLI):** Scripts para automatizar tareas, herramientas de desarrollo, etc.
  - **Procesos de build y bundling:** Muchas herramientas que usamos en el desarrollo frontend moderno (como Webpack, Vite, Parcel) están construidas con Node.js.
  - **Aplicaciones de escritorio (con frameworks como Electron):** Es posible hacer apps de escritorio con JS.
  - **Internet de las Cosas (IoT):** Controlar hardware y dispositivos.

## ¿Por qué es importante para nosotros?

La principal ventaja es poder **seguir usando el lenguaje que usa el navegador en el lado del cliente (JavaScript)** para construir una gama mucho más amplia de aplicaciones. La filosofía "JavaScript Everywhere" se hace realidad con Node.js.

## La Gran Diferencia con el Navegador (Un Adelanto):

Como Node.js no se ejecuta en un navegador, **no tiene acceso a elementos específicos del navegador**.

Sin embargo, Node.js nos da acceso a otras capacidades que el navegador, por seguridad, no permite:

- Acceso completo al **sistema de archivos** de nuestro ordenador (leer, escribir archivos).
- Interactuar directamente con **bases de datos**.
- Comunicación a bajo nivel por **red**.

- Acceso a información del **sistema operativo** y del proceso en ejecución.
-