

¿Qué es NodeJS?

NodeJS: Javascript en el servidor

¿Qué es NodeJS? Primeros pasos con NodeJS

En 2009, aparece un **entorno multiplataforma** llamado **NodeJS**. La idea era sencilla: Dada la popularidad en ascenso de Javascript, que en ese momento sólo era posible utilizarlo dentro de un navegador, se ideó un sistema que «sacara» Javascript del navegador, haciendo posible ejecutarlo en cualquier lugar fuera de él, por ejemplo, en un servidor web, convirtiéndolo en un lenguaje de servidor y no sólo un lenguaje de cliente (navegador) como hasta el momento.

Requisitos previos

NodeJS es un entorno multiplataforma, por lo que da igual si se trata de un sistema operativo Windows, Mac o GNU/Linux, NodeJS es capaz de ejecutarse en cualquiera de ellos. Sin embargo, y de forma totalmente subjetiva por quién escribe estas líneas, si estás utilizando un sistema operativo **Windows**, te recomiendo tomarte un tiempo en instalar y configurar **WSL**.

WSL (Windows Subsystem for Linux) es un sistema que te permite disponer de una terminal de Linux como si fuese nativa, lo que simplifica muchísimo el desarrollo web, ya que en la mayoría de los casos, terminamos creando desarrollos para subirlos a máquinas Linux.

¿Qué es Express.js?

Express.js es el framework backend más popular para [Node.js](#), y es una parte extensa del [ecosistema JavaScript](#).

Está diseñado para construir aplicaciones web de una sola página, multipágina e híbridas, también se ha convertido en el estándar para desarrollar aplicaciones backend con Node.js, y es la parte backend de algo conocido como la pila MEVN.

El MEVN es un Stack de software [JavaScript](#) libre y de código abierto para construir sitios y aplicaciones web dinámicas que tiene los siguientes componentes

1. **MongoDB:** [MongoDB](#) es la base de datos NoSQL estándar
2. **Express.js:** El framework de aplicaciones web por defecto para construir aplicaciones web
3. **Vue.js:** El framework progresivo de JavaScript utilizado para construir [aplicaciones web front-end](#)

4. **Node.js:** El motor de JavaScript utilizado para aplicaciones escalables del lado del servidor y de la red.

Esta guía explorará las características clave del framework Express.js y cómo construir tu primera aplicación.

¿Qué es una base de datos?

Base de datos definida

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un [sistema de gestión de bases de datos \(DBMS\)](#). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

[Obtén más información sobre Oracle Database](#)

¿Qué es una API REST?

Una API REST (también llamada API RESTful o API web RESTful) es una [interfaz de programación de aplicaciones \(API\)](#) que se ajusta a los principios de diseño del estilo arquitectónico de *transferencia de estado representacional* (REST). Las API REST proporcionan una forma flexible y ligera de integrar aplicaciones y conectar componentes en arquitecturas de [microservicios](#).



What is a REST API? (9:11)

Definida por primera vez en 2000 por el científico informático Dr. Roy Fielding en su disertación doctoral, REST proporciona un nivel relativamente alto de flexibilidad, escalabilidad y eficiencia para los desarrolladores. Por estas razones, las API REST han surgido como un método común para conectar componentes y aplicaciones en una arquitectura de [microservicios](#).

¿Qué es el desarrollo back-end?

El desarrollo front-end implica lo que un usuario ve en la pantalla cuando abre una URL específica de tu propiedad. Incluso en un entorno completamente estático (solo con HTML/CSS), cuando alguien abre un sitio web, algún servidor en el planeta necesita responderle con esos archivos HTML y CSS.

Ese servidor es solo un ordenador, como el que usas para navegar por Internet. Pero se ha ajustado para el rendimiento y no tiene componentes innecesarios como un mouse o un teclado adjuntos. Y se encuentra con toneladas de otros ordenadores, probablemente en un almacén de datos.

La programación de esos ordenadores de alguna manera especial se llama desarrollo de back-end .

Puedes pensar que el desarrollo de back-end se llama así porque se ejecuta a espaldas del usuario. Un visitante de tu sitio web nunca "accede" completamente al back-end. Simplemente se comunican con tu servidor, ya sea directamente a través de puertos para un acceso muy limitado (como la

transferencia de archivos HTML/CSS) o ni siquiera eso, enterrados profundamente bajo CDN o firewalls (como Cloudflare).

Ahora que tenemos una comprensión básica de lo que significa el desarrollo de back-end, analicemos algunas preguntas reales.

¿Qué es el desarrollo front-end?

El desarrollo front-end, también conocido como desarrollo del lado del cliente, es un componente crucial del desarrollo web. Es el proceso de creación de la interfaz de usuario de un sitio o aplicación web utilizando lenguajes como HTML, CSS y JavaScript. El front-end es lo que los usuarios ven y con lo que interactúan en un sitio web, y es responsable de la maquetación, el diseño y la funcionalidad de la interfaz de usuario.

Cuando se visita un sitio web, el código del front-end envía peticiones al servidor y recibe respuestas. El navegador traduce estas respuestas en lo que el usuario ve en su pantalla. Los desarrolladores de front-end utilizan varias herramientas y tecnologías para crear diseños adaptables que funcionen a la perfección en todos los dispositivos y navegadores.

Hay tres lenguajes principales utilizados en el desarrollo front-end: HTML, CSS y JavaScript. HTML se utiliza para crear la estructura de un sitio web, mientras que CSS se utiliza para darle estilo. JavaScript se utiliza para añadir interactividad y funcionalidad al sitio web.

Para que un proyecto de desarrollo front-end tenga éxito, los desarrolladores deben conocer a fondo el funcionamiento conjunto de cada uno de estos lenguajes. También deben tener buen ojo para el diseño, ya que el front-end es responsable de crear el atractivo visual de un sitio web.

¿Desarrollador full Stack?

Un desarrollador full Stack es un especialista en el desarrollo de sitios web que integra el diseño de la experiencia del cliente en una página (front end), así como la programación y mantenimiento de la arquitectura interna del sitio (back end).

El perfil de un desarrollador full Stack debe, por tanto, ser híbrido y complementar habilidades creativas y de diseño con un pensamiento estratégico y una capacidad analítica, así como tener conocimientos técnicos en programación y desarrollo web.

CRUD.

En **informática**, CRUD es el **acrónimo** de "Crear, Leer, Actualizar y Borrar" (del original en **inglés**: *Create, Read, Update and Delete*), que se usa para referirse a las funciones básicas en **bases de datos** o la capa de persistencia en un **software**.

En algunos lugares, se utilizan las siglas ABM para lo mismo ("Alta, Baja y Modificación"), obviando la operación de "obtener"; el acrónimo ABC para "Altas, Bajas y Cambios"; ABML siendo la última letra (L) de "listar, listado o lectura"; ABMC siendo la 'C' de "Consulta"; o bien CLAB que sería la traducción literal del acrónimo ("Crear, Leer, Actualizar y Borrar") también se llega a usar el acrónimo ABCC ("Altas, Bajas, Cambios y Consultas").

También es usado el ABCDEF: "Agregar, Buscar, Cambiar, Desplegar (listar), Eliminar, Fichar (Ficha, cédula o Reporte de un registro)".

En programación, crear, leer, actualizar y borrar (con el acrónimo CRUD) son las cuatro funciones básicas de la persistencia de **bases de datos**. Términos alternativos son usados a veces cuando se definen las cuatro funciones básicas de CRUD, como "recuperar" en vez de "leer", "modificar" en vez de "actualizar" o "destruir" en vez de "borrar". CRUD se usa también a veces para describir convenciones de interfaz de usuario que facilita la vista, búsqueda y modificación de la información; a menudo se usa en programación de formularios (forms) e informes (reports). El término fue popularizado por primera vez por **James Martin** en su libro del año 1980 *Managing the Data-base Environment*. El acrónimo puede extenderse a CRUDL para cubrir el listado de gran cantidad de datos que conllevan una complejidad tal como paginación cuando los registros de datos son demasiado grandes para alojarse fácilmente en memoria.