

NODE

¿Que es Node?

Se trata de un entorno de código abierto (open source) multiplataforma que ejecuta el código JavaScript fuera de un navegador. Este entorno de ejecución de JavaScript se orienta a eventos asíncronos (los eventos no dependen de que otros se hayan ejecutado previamente) y permite construir aplicaciones en red escalables, es decir, tiene la capacidad de realizar muchas conexiones de manera simultánea sin que tenga que leer el código línea a línea, ni abrir múltiples procesos.



¿Qué puedo hacer con Node.js?

Node.js fue creado por los desarrolladores originales de JavaScript con la idea de poder ejecutar este lenguaje fuera del entorno del navegador. Para ello utilizaron el motor V8 de Chrome. Este motor se ocupa de convertir el código JavaScript a código máquina en tiempo real (a lo que se llama JIT, o just-in-time), como corresponde a un lenguaje interpretado como JS (a diferencia de lo que ocurre con los lenguajes compilados que, como bien dice su nombre, deben ser compilados antes de poder ser ejecutados).

Node.js no solo permite crear servidores web o Aplicaciones Front End, sino que, los hace más ágiles y capaces de trabajar con otros lenguajes de secuencia como Python. Por esta razón, los desarrolladores lo utilizan sobre todo en aplicaciones de red que buscan ser rápidas o en proyectos de gran envergadura donde se necesita que los procesos sean ágiles.

Con Node.js podemos desarrollar:

- Juegos real-time
- Salas de chat
- Aplicaciones de recolección de datos
- Streaming
- Aplicaciones de alto tráfico y concurrencia (ej. Twitter)
- Back-End

NPM

NPM se puede considerar como las siglas de Node Package Manager, es decir, gestor de paquetes de NodeJS, un entorno de ejecución multiplataforma para ejecutar Javascript no sólo en un navegador web (como se concibió originalmente) sino fuera de él, y poder utilizarlo en sistemas de escritorio o servidores web.

Este gestor de paquetes (muy similar al concepto de apt-get en GNU/Linux), nos permitirá instalar de forma muy sencilla y automática paquetes Javascript (tanto de Node como Javascript para el navegador) para poder utilizarlos y mantenerlos en los proyectos o sistemas que utilicemos.



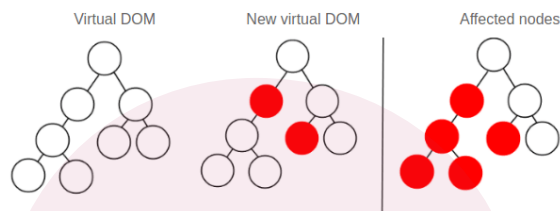
Virtual DOM (VDOM):

El Virtual DOM es un concepto utilizado en React.js para optimizar el rendimiento de las aplicaciones web.

Es una representación virtual de la estructura del DOM (Document Object Model) de una página web.

React utiliza el Virtual DOM para realizar comparaciones eficientes entre la representación actual del Virtual DOM y la representación anterior, para determinar los cambios que deben aplicarse en el DOM real.

El Virtual DOM permite a React minimizar las operaciones de manipulación del DOM real, lo que mejora el rendimiento de las aplicaciones al reducir la cantidad de actualizaciones necesarias.



ReactDOM:

ReactDOM es una parte específica de la biblioteca React.js que se encarga de manipular el DOM real en el navegador web.

ReactDOM es responsable de tomar la representación del Virtual DOM y aplicar los cambios necesarios en el DOM real para que la interfaz de usuario refleje el estado actual de la aplicación.

ReactDOM.render() es un método comúnmente utilizado para renderizar componentes de React en el DOM real.

En resumen, el Virtual DOM es un concepto fundamental en React que ayuda a optimizar las actualizaciones del DOM, mientras que ReactDOM es una parte de la biblioteca React que se encarga de gestionar las interacciones con el DOM real. Ambos trabajan juntos para lograr un rendimiento eficiente en aplicaciones React, pero no son lo mismo.

