El futuro digital es de todos

MinTIC

Introducción a NodeJS y npm

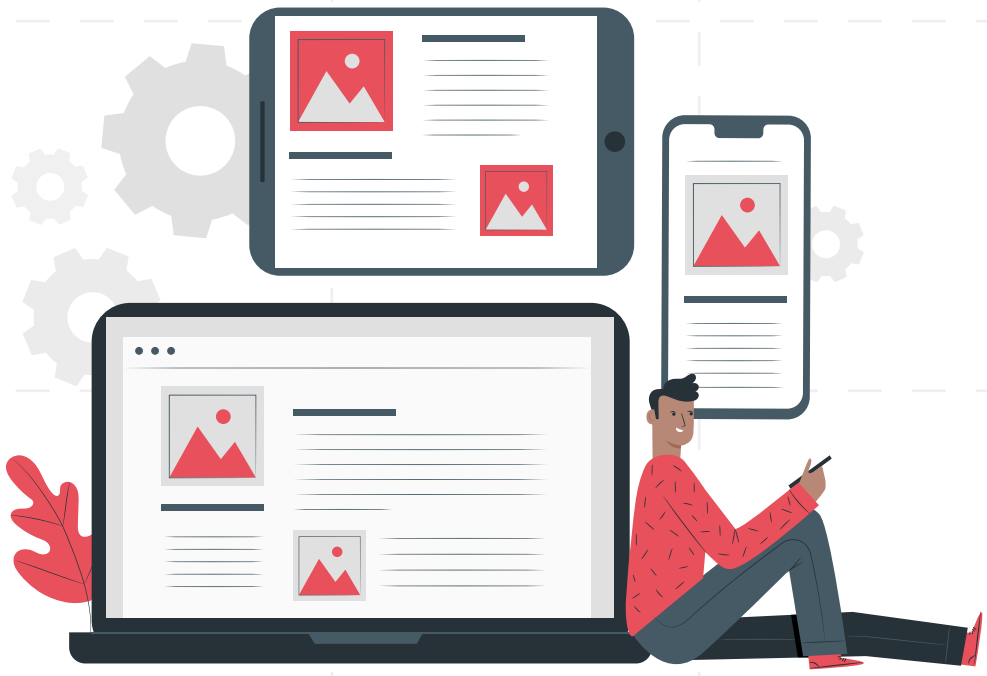
Universidad de Caldas

Hola

Git y MongoDB Atlas nos soportan el trabajo en equipo en la programación web. Aunque esto necesita profundización por parte de ustedes y para ello cuentan con el material complementario de la semana.

NodeJS es un framework escrito en lenguaje Javascript, un lenguaje que hasta hace un tiempo se encontraba en caída libre y tendía a desaparecer.

Debemos comprender dos conceptos clave para la programación web, el Cliente y el Servidor. El Cliente *(Frontend)*, en programación web, es aquel componente que nos permite como usuarios, interactuar con el sistema. En este cliente encontramos campos de texto, botones, listas desplegables, menús, etc. Todos estos elementos que encontramos en el cliente son desarrollados a través de lenguajes de etiquetado como el HTML, CSS, y el propio Javascript. Para ello, en el curso utilizaremos frameworks que permiten agilizar el trabajo y disminuir el tiempo de desarrollo.



Cliente

Por su parte, el Servidor es aquello que el usuario no visualiza, pero que sin ello, el Cliente *(Frontend)* no tendría funcionalidad alguna. En el Servidor *(Backend)* se encuentra toda la lógica del negocio, la interacción con otros sistemas, el acceso a las bases de datos, entre otras funcionalidades.

Se debe aclarar que actualmente existen estrategias y metodologías que son soportadas por lenguajes de programación para dar solución a problemáticas actuales. Entre ellos se encuentran los microservicios *(microfrontends)* y algunas arquitecturas basadas en capas y en componentes, como las arquitecturas limpias.

Servidor

Es fundamental conocer un poco más acerca de arquitecturas de *software*, sin embargo, es un tema que llevaría el tiempo de un módulo completo, por ello los invitamos a profundizar por cuenta propia en este tema tan importante.

NodeJS permitió que Javascript surgiera de nuevo con su ejecución en entorno de Servidor. Además, brinda a los desarrolladores soluciones bastante dinámicas, que manejan muy bien el tema de concurrencia de usuarios. La concurrencia de usuarios se refiere a la gran cantidad de conexiones que puede tener un mismo servidor en una pequeña ventana de tiempo, haciendo que un servicio o un servidor se vuelvan lentos y, en ocasiones, ya no responda a nuestras peticiones. Esto se logra a través de los llamados asíncronos que son soportados por NodeJS.

Node, como es conocido, ha permitido a diversas compañías a nivel mundial la creación de diversos *frameworks* para facilitar el desarrollo de *software* y estandarizar procesos. Todos estos *framework* utilizan lo mejor de Node para mejorar así la concurrencia de usuarios. NodeJS tiene la ventaja de ejecutarse en el Cliente y en el Servidor, de tal manera que los conocedores de este *framework* puedan desarrollar en *Frontend* o en *Backend* indistintamente. También, esto permite a los desarrolladores reutilizar código de Cliente en Servidor.

Una parte fundamental para Node, es la evolución del mismo lenguaje Javascript, permitiendo que se aproveche al máximo cada uno de los recursos de la máquina donde se ejecuta este lenguaje. Una de las estrategias trascendentales en las funciones asíncronas de Node son las promesas y los observables.

Node se basa en módulos, es decir, es un rompecabezas que podemos armar de una manera independiente de las piezas que necesitemos en un momento dado. De una manera más clara, podemos decir que Node es el núcleo de todo, y podemos agregar otros componentes desarrollados para ampliar las funcionalidades base.

Para gestionar todos estos módulos, que en realidad se llaman Paquetes, tenemos el NPM o administrador de Paquetes de Node. Los Paquetes por si solos son unidades pequeñas que brindan funcionalidades específicas. Una aplicación de Node puede contener cientos de Paquetes que pueden ser reutilizados en muchos proyectos diferentes. Cada Paquete tiene poco código y su mantenimiento se hace de manera sencilla. Como desventaja tenemos el control de versiones, ya que un sistema puede hacer uso de un Paquete específico en la versión uno (1), pero si ese Paquete cambia a una versión dos (2), puede modificar su funcionalidad y por ende, cambiará la funcionalidad del sistema que lo utiliza.

Para finalizar realizaremos la instalación de NodeJS y la prueba de que se ha instalado correctamente:

Pasos de la instalación

1. Descarga del instalador desde nodejs.org
2. Paso a paso de la instalación.
3. Prueba de la consola de NodeJS.

Explicación instalación video

Una vez instalado NodeJS en nuestro computador y que comprendemos la naturaleza de este *framework* es hora de practicar un poco y de profundizar. Desde consola de NodeJS practiquen algoritmos de módulos anteriores.

Gracias por acompañarnos.

Universidad de Caldas