



Tema: Node

Alumno: Hernández Rodrigo

Mail: rodrigoenzohernandez@gmail.com

Fecha: 20/07/2020

1. ¿Por qué nació Node? ¿Para qué sirve?

Node nació para permitir utilizar el lenguaje JavaScript del lado del servidor, permitiendo tener tanto en el front end, como el back end, el mismo lenguaje de programación. Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente

Enlaces para leer:

- Enlace 1: <https://openwebinars.net/blog/que-es-nodejs/#:~:text=iniciar%20la%20programaci%C3%B3n.-,Node.,Gracias%20a%20Node.>

Este enlace permite una introducción al funcionamiento de Node JS, sus ventajas y su objetivo.

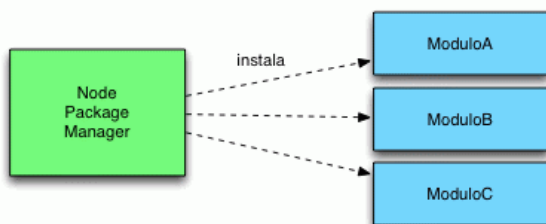
- Enlace 2: <https://blog.webtraining.zone/cuando-se-creo-nodejs/#:~:text=el%20%C3%A1mbito%20enterprise.-,Node.,originales%20usadas%20durante%20esa%20conferencia%20.&text=js%2C%20eso%20significa%20que%20realmente,%C3%A9l%20%C2%A1no%20lo%20dejes%20ir!>

Lo interesante de este enlace es que cuenta con el link a Youtube con la presentación de Node JS realizada por Ryan Dahl.

2. ¿Cuál es la diferencia entre Node y NPM?

La diferencia es que Node JS es un entorno de ejecución de ejecución para JavaScript, mientras que NPM (Node Package Manager) es un gestor de paquetes de node, que permite de forma sencilla:

- instalar cualquier librería disponible con solo una línea de código
- administrar módulos
- distribuir paquetes
- agregar dependencias



Enlaces para leer:

- Enlace 1: <https://devcode.la/blog/que-es-npm/>

Este enlace es interesante porque cuenta con una breve descripción de npm y lista los comandos más importantes.

- Enlace 2: <https://docs.npmjs.com/about-npm/>

Este enlace es interesante ya que cuenta con la documentación oficial de npm.

3. ¿Qué es un módulo en el ambiente de Node?

Un módulo en el ambiente de Node es una librería, que se instala con el Node Package Manager y luego podemos utilizar.



Enlaces para leer:

- Enlace 1: <https://www.genbeta.com/desarrollo/node-js-y-npm>

Interesante enlace que explica de forma sencilla la diferencia entre npm (node package manager) y npm (node package module), que suelen confundirse ya que tienen las mismas siglas.

- Enlace 2: <https://www.npmjs.com/>

Este enlace nos dirige al repositorio general de Node, el cual almacena todos los Node Package Modules que han sido publicados.

4. Si se tuviera que definir el tipo de dato de un módulo requerido, ¿qué dato sería?

Si se tuviera que definir el tipo de dato de un módulo requerido, diría que es un objeto literal, ya que la función nativa de node Require() que recibe como parámetro el nombre del módulo (un string), devuelve un objeto literal.

Enlaces para leer:

- Enlace 1: <https://www.escuelajavascript.com/como-funciona-el-sistema-de-modulos-commonjs-y-require/>

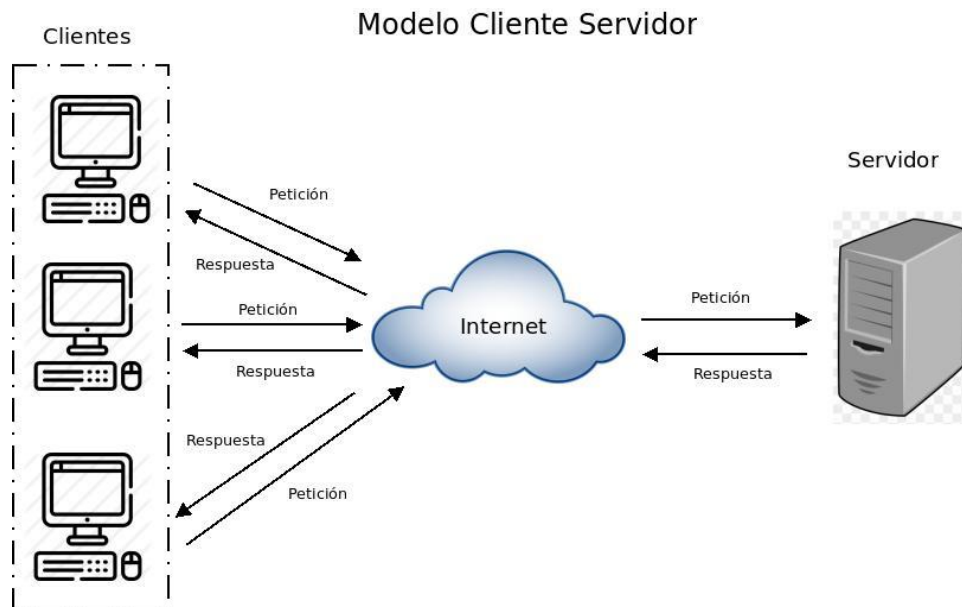
Este enlace cuenta con una explicación del sistema de módulos, algunos ejemplos, explicación de la función require, de los módulos load, compile y de las distintas versiones de npm.

- Enlace 2: <https://www.todojs.com/modulo-is-para-navegador-y-node/>

El enlace cuenta con una introducción a los módulos y un ejemplo completo de utilización de un módulo.

5. Dentro del entorno client / server ¿Qué diferencia a un cliente de un servidor?

En un entorno de cliente – servidor, un cliente se refiere al navegador web que envía solicitudes mientras que el servidor es un ordenador que cuenta con una aplicación en ejecución capaz de atender las peticiones de un cliente y devolverle una respuesta, un servidor puede ser tanto físico como virtual.



Enlaces para leer:

- **Enlace 1:** <https://blog.infranetworking.com/modelo-cliente-servidor/>

El enlace cuenta con una descripción detallada del modelo cliente servidor e información relacionada.

- **Enlace 2:** <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>

El enlace cuenta con las dos definiciones principales de servidor, en el ambito de IT. Una hace referencia al ordenador que pone recursos a disposición a través de una red, y el segundo se refiere al programa que funciona en dicho ordenador

6. ¿Para qué sirve el protocolo HTTP?

HTTP (Hypertext Transfer Protocol / Protocolo de Transferencia de Hipertexto). Es un protocolo de comunicación que permite transferir información a través de internet. Con el http se establecen criterios de sintaxis y semántica informática (forma y significado) para el establecimiento de la comunicación entre el cliente y el servidor.

Enlaces para leer:

- **Enlace 1:** <https://developer.mozilla.org/es/docs/Web/HTTP/Overview#:~:text=HTTP%2C%20de%20sus%20siglas%20en,como%20pueden%20ser%20documentos%20HTML.>

Este enlace cuenta con las generalidades del protocolo HTTP.

- Enlace 2: <https://concepto.de/http/#ixzz6Sli4Ax99>

El enlace cuenta con la definición clara y sencilla de HTTP.

7. ¿Cuales son lo métodos más usados dentro del protocolo HTTP?

Los métodos más usados dentro del protocolo HTTP son GET y POST.

- El método GET solicita una representación de un recurso específico, es decir, que información será enviada hacia el cliente. Las peticiones que usan el método GET sólo deben recuperar datos.
- El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor, en otras palabras el POST indica que hay datos que van a ser enviados hacia el servidor (creando o modificando un recurso, o generando un documento temporal para ser enviado).

Una de las grandes diferencias entre ambos es que el método GET puede visualizarse en la URL, mientras que el POST se ejecuta de forma “oculta” o en segundo plano.

Enlaces para leer:

- Enlace 1: <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

El enlace cuenta con todos los métodos de petición HTTP y una breve descripción.

- Enlace 2: <https://developer.mozilla.org/es/docs/Web/HTTP/Messages>

El enlace cuenta con ejemplos de peticiones HTTP.