



Introducción a Node.js

EDWIN ANTONIO LÓPEZ ORDÓÑEZ

¿Qué es Node.js?

- Es un entorno de ejecución para JavaScript.
- Construido con V8, el motor de JavaScript de Google Chrome.
- Está orientado a eventos asíncronos.
- Utilizado generalmente para la construcción de servidores web, pero no se limita solo a eso.
- Es independiente del sistema operativo.

¿Qué es JavaScript?

- Lenguaje de programación ligero, interpretado o compilado “just-in-time”.
- Creado por Netscape.
- Es débilmente tipado.
- Recomendación para aprenderlo: <https://www.sololearn.com/learn/courses/javascript-introduction>



Obteniendo Node.js

- Lo descargamos de la página oficial: <https://nodejs.org/es/download>
- Hay 2 versiones recomendadas, la "LTS" (Long Term Support), la cual tendrá 18 meses de soporte y luego 12 meses de mantenimiento y la "Latest" que nos trae las características más nuevas, pero también incluirá bugs.
- Existen distintas formas de instalarlo, las cuales variarán de sistema operativo en sistema operativo.

Node Package Manager – npm

- Pieza muy importante del ecosistema de Node.js, tanto así que por lo regular se instala junto al Node.js.
- Es un repositorio online para la publicación de paquetes de software libre para su utilización en proyectos de Node.js.
- También es una herramienta para la terminal la cual nos permitirá la interacción con dicho repositorio para la instalación de paquetes, actualización de los mismos y manejo de dependencias.
- Su página web es: <https://www.npmjs.com/>



Primeros pasos con Node.js

- Después de instalar el mismo debemos abrir una terminal e ingresar el comando:
`npm init`
- Este comando nos pedirá mucha información respecto al paquete, para evitar tener que llenarla toda podemos ejecutarla con el parámetro `-y`.
- Esto nos generará un archivo `package.json`, el cual es la columna vertebral de cualquier proyecto de Node.js (entre otros).
- Este archivo contendrá el detalle de todos los paquetes utilizados en el proyecto, lo cual nos ayudará a evitar complicaciones a la hora de distribuir nuestra aplicación.

Solicitudes y respuestas en HTTP

- En el contexto de HTTP (Protocolo de Transferencia de Hipertexto), el "request" (solicitud) y el "response" (respuesta) son dos conceptos fundamentales que se utilizan para comunicarse entre un cliente y un servidor web.
- El HTTP request (solicitud HTTP) es una petición realizada por un cliente para obtener información o realizar una acción en un servidor, mientras que el HTTP response (respuesta HTTP) es la respuesta enviada por el servidor al cliente en respuesta a esa solicitud.



Solicitudes HTTP

- Es una petición realizada por un cliente (por lo general, un navegador web) para obtener información o realizar una acción en un servidor web.
- Generalmente contiene varios componentes clave, incluyendo:
 - La URL (Uniform Resource Locator) que identifica la ubicación y el recurso que se está solicitando.
 - El método HTTP, que especifica la acción que se desea realizar en el recurso (por ejemplo, GET para obtener datos, POST para enviar datos, etc.).
 - Cabeceras (headers) que contienen información adicional sobre la solicitud, como el tipo de navegador del cliente, las cookies, y más.
 - Opcionalmente, datos del cuerpo de la solicitud, que se utilizan en algunos métodos HTTP para enviar información adicional al servidor.




Tipos de solicitudes HTTP

- GET: Solicita recursos del servidor, como páginas web o imágenes.
- POST: Envía datos al servidor para su procesamiento, estos se envían en el cuerpo de la solicitud, garantizando su confidencialidad.
- PUT: Actualiza un recurso específico en el servidor.
- DELETE: Borra un recurso específico en el servidor.



Respuestas HTTP

- Es la respuesta que el servidor envía de vuelta al cliente como resultado de la solicitud realizada por este último.
- Una respuesta HTTP también contiene varios componentes clave, incluyendo:
 - El código de estado HTTP, que indica si la solicitud se completó correctamente o si hubo algún error (por ejemplo, el código 200 OK significa éxito, mientras que el código 404 Not Found indica que el recurso no se encontró).
 - Cabeceras (headers) que proporcionan información adicional sobre la respuesta, como el tipo de contenido, la fecha de la respuesta, etc.
 - El cuerpo de la respuesta, que contiene el contenido real que el cliente solicitó (por ejemplo, una página web en formato HTML, datos JSON, una imagen, etc.).



Tipos de código de estado HTTP

- Son números de tres dígitos que se envían en las respuestas HTTP del servidor para indicar el resultado de una solicitud realizada por el cliente.
- De éxito:
 - 200 (OK): La solicitud se completó con éxito.
 - 201 (Created): Se creó exitosamente un nuevo recurso en el servidor.
- De error del cliente:
 - 400 (Bad request): Solicitud incorrecta o mal formada y no pudo ser procesada.
 - 401 (Unauthorized): Se requiere autenticación para acceder al recurso.

¿Qué es JSON?

- Su nombre viene de "JavaScript Object Notation" (Notación de Objetos de JavaScript), es un formato de datos ligero y de fácil lectura que se utiliza para el intercambio de datos entre sistemas. JSON se ha vuelto extremadamente popular en el desarrollo de aplicaciones web y API debido a su simplicidad y facilidad de uso.
- Ejemplo:

```
{  
  "nombre": "Juan",  
  "edad": 30,  
  "ciudad": "Villa Nueva"  
}
```



¿Dudas o preguntas?

Gracias por la atención

