

“NODE.JS PARA EL DESARROLLO DE APLICACIONES WEB”

Ingeniería en Tecnologías de Información y Comunicaciones – Ingeniera Vázquez Galicia
Mitzi Ailyn.

RESUMEN

Esta presente investigación trata de dar a conocer la definición, origen, características, ventajas, desventajas, aplicaciones, implementación, y el desarrollo de aplicaciones web mediante Node.js, con el propósito de dar a conocer la importancia que este entorno tiene en los sitios web, además de pretender ser una guía en instalación de esta herramienta para la creación de futuras prácticas o proyectos que desarrolle cualquier persona en algún futuro.

ABSTRACT

This present investigation tries to make known the definition, origin, characteristics, advantages, disadvantages, implementation, and development of web applications through Node.js, with the purpose of making known the importance that this environment has on websites, in addition of pretending to be a guide in the installation of this tool for the creation of future practices or projects that anyone can develop in the future.

I. INTRODUCCIÓN

Node.js es un entorno de tiempo de ejecución de JavaScript, de ahí su terminación “.js”. Este entorno de tiempo es open source, es decir, de código abierto, multiplataforma y que se ejecuta del lado del servidor.

Sirve para crear sitios web dinámicos muy eficientes, escritos con el lenguaje de

programación JavaScript. Normalmente, los desarrolladores se decantan por este entorno de ejecución cuando buscan que los procesos se ejecuten de forma ágil y sin ningún tipo de bloqueo cuando las conexiones se multiplican.

II. DEFINICIÓN

Node.js es un entorno en tiempo de ejecución multiplataforma para la capa del servidor, es decir, del lado del servidor, este entorno está basado en JavaScript.

Es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos.

El diseño de Node.js está inspirado en sistemas como el Event Machine de Ruby o el Twisted de Python. Sin embargo, Node.js presenta un bucle de eventos como una construcción en tiempo de ejecución en lugar de una biblioteca. Este bucle de eventos es invisible para el usuario.

Node.js está diseñado para simplificar la comunicación. No tiene subprocesos, pero te permite aprovechar múltiples núcleos en su entorno y compartir sockets entre procesos.

III. ORIGEN

Este entorno fue creado por los desarrolladores de JavaScript con el

objetivo de ir un paso más allá con este lenguaje de programación.

La creación de Node.js es en el año 2009, debido a que el lenguaje de programación JavaScript únicamente podía ejecutarse del lado del navegador o cliente, pero... ¿Qué ocurría si queríamos ejecutar este lenguaje fuera del navegador o del lado del servidor?

Como JavaScript únicamente se podía utilizar dentro del marco de las etiquetas `<script> </script>`, los desarrolladores tenían que implementar diferentes lenguajes y herramientas tanto en el frontend como en el backend.

Por esa razón, se creó Node.js, ya que, posee todo lo necesario para ejecutar código JavaScript del lado del servidor. Algo que facilita mucho el trabajo de los desarrolladores y el motivo por el que actualmente sea una de las herramientas de trabajo más usadas en desarrollo web.

IV. CARACTERÍSTICAS

Resumiendo, Node.js se ha hecho popular en los últimos años gracias a las siguientes características:

- » Velocidad. Node.js está construido sobre el motor de JavaScript V8 de Google Chrome, por eso su biblioteca es muy rápida en la ejecución de código.
- » Sin búfer. Las aplicaciones de Node.js generan los datos en trozos (chunks), nunca los almacenan en búfer.
- » Asíncrono y controlado por eventos. Las APIs de la biblioteca de Node.js son asíncronas, sin bloqueo. Un servidor basado en Node.js no espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarla, y un mecanismo de notificación de eventos ayuda al servidor a obtener una respuesta de la llamada a la API anterior.

- » Un subproceso escalable. Node.js utiliza un modelo de un solo subproceso con bucle de eventos. Gracias al mecanismo de eventos, el servidor responde sin bloqueos, como hemos dicho. Esto hace que el servidor sea altamente escalable comparando con los servidores tradicionales como el Servidor HTTP de Apache.

V. VENTAJAS

La arquitectura basada en eventos asíncronos de Node.js y la utilización del motor V8 de Google convierte a Node.js en uno de los entornos de ejecución con más crecimiento y el preferido para el desarrollo de aplicaciones web o de escritorio. Y es que las ventajas no son pocas:

- » Escalable: al poder procesar tantas conexiones de forma simultánea, Node.js es un entorno perfecto para construir aplicaciones en red escalables con un alto nivel de rendimiento.
- » Rendimiento y eficacia: la ejecución de procesos sin bloqueo y la utilización de menos recursos convierte a Node.js en un entorno rápido y muy eficaz.
- » Facilidad: Node.js es el entorno que utiliza JavaScript, un lenguaje muy bien estructurado y relativamente fácil de aprender.
- » Open source: es un software de código libre, esto quiere decir que el código es abierto y no es necesario poseer ninguna licencia para su utilización.
- » Comunidad y soporte: el uso y crecimiento de Node.js, así como la implementación de plataformas como GitHub ha hecho que la comunidad de Node.js crezca y se mantenga activa con el objetivo de mantener y mejorar este entorno, así como la disponibilidad de documentación relativa a este entorno de ejecución.

VI. DESVENTAJAS

Este entorno de ejecución también puede presentar algunos inconvenientes como los que se mencionan a continuación.

- » Curva de aprendizaje: Node.js necesita mucho de otros sistemas y requiere de más líneas de codificación, algo que puede resultar mucho más tedioso si se está acostumbrado a trabajar con otros sistemas o lenguajes de programación como PHP.
- » Compatibilidad: no todos los planes de alojamiento web son compatibles con Node.js, por lo que para utilizarlo se necesita un Hosting especializado en Node.js.
- » Módulos en desarrollo: aunque Node.js posee un sistema llamado NPM de módulos o dependencias, pueden resultar escasos si los comparas con otros sistemas.
- » Documentación: Es cierto que es un entorno que no para de crecer, por tal motivo, mucha documentación todavía se encuentra sin traducir.

VII. APLICACIÓN

Se puede utilizar Node.js para diferentes tipos de aplicaciones. Los siguientes son algunos de los ejemplos:

- » Aplicaciones de transmisión de datos (streaming)
- » Aplicaciones intensivas de datos en tiempo real
- » Aplicaciones vinculadas a E/S
- » Aplicaciones basadas en JSON:API
- » Aplicaciones de página única

VIII. IMPLEMENTACIÓN

Casi todas las marcas más importantes del mercado utilizan Node.js. A continuación, se muestran algunos ejemplos:

- » GoDaddy
- » Microsoft

- » eBay
- » General Electric
- » PayPal
- » Uber
- » NASA
- » Netflix
- » LinkedIn
- » Medium

IX. DESARROLLO EN APLICACIONES WEB.

Colocamos el siguiente link en nuestro navegador <https://nodejs.org/es/>



Ilustración 1. Página de descarga de Node.js

Descargamos el instalador de Windows, lo ejecutamos y aparecerá la siguiente pantalla.

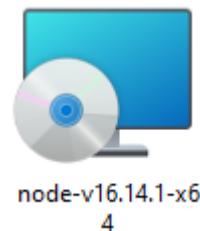


Ilustración 2. Instalador de Node.js.

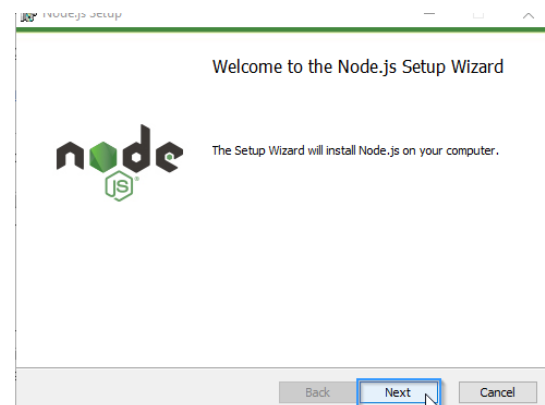
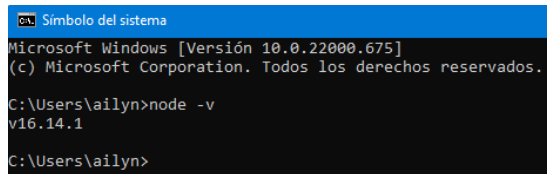


Ilustración 3. Instalación de Node.js.

Una vez finalizada la instalación, se podrá comprobar de una forma sencilla si está instalado correctamente. Para ello, abrimos el terminal del ordenador pulsando la tecla Windows y escribiendo “cmd”. Selecciona “Símbolo de sistema”, en esa terminal colocamos el siguiente comando `node -v` y pulsa Enter. Si Node.js está correctamente instalado te aparecerá la versión que tienes instalada.



```

Simbolo del sistema
Microsoft Windows [Versión 10.0.22000.675]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\ailyn>node -v
v16.14.1
C:\Users\ailyn>

```

Ilustración 4. Versión de Node.js.

De esta manera, se puede implementar node.js en el desarrollo de aplicaciones web. Con Node.js se pueden construir aplicaciones web tradicionales, como sitios web. De hecho, existen diversos proyectos de CMS basados en Node. Sin embargo, el uso más común de Node.js es el desarrollo de servicios web que devuelven datos en formato JSON, lo que llamamos habitualmente API REST. También con Node.js podemos construir aplicaciones de escritorio multiplataforma, compatibles con Windows Linux y Mac, así como programas de consola. Muchas herramientas del día a día de los desarrolladores están realizadas usando Node.js, desde editores como Atom o VSCode, hasta automatizadores de tareas como Gulp o empaquetadores de assets como Webpack o Rollup.

X. EJEMPLO DE DESARROLLO

Creamos una carpeta llamada “node-mysql”.

Abrimos Visual Code, dentro de la carpeta que acabamos de crear y desde su terminal tecleamos lo siguiente: `npm init`

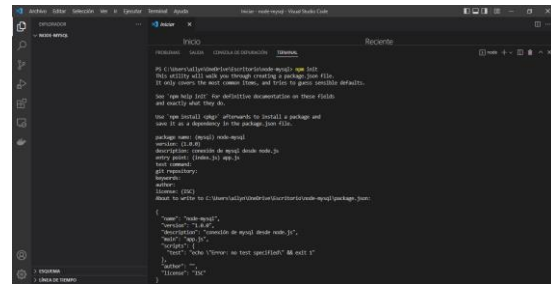


Ilustración 5. Instalación requerimientos del proyecto.

yes para crear un package json, y a partir de ahí empezar a crear las dependencias o bibliotecas del servidor.

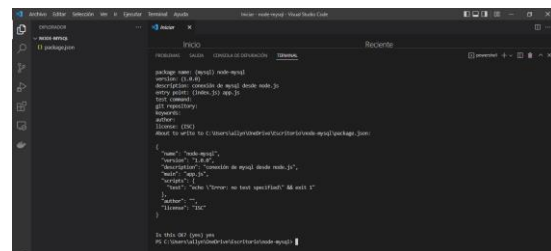
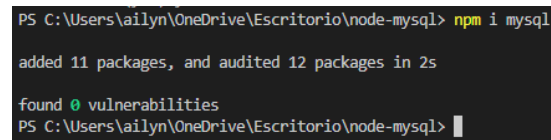


Ilustración 6. Instalación de package.json.

Vamos a instalar descargamos mysql para almacenar la información del sitio web con el comando `npm i mysql`



```

PS C:\Users\ailyn\OneDrive\Escritorio\node-mysql> npm i mysql

added 11 packages, and audited 12 packages in 2s

found 0 vulnerabilities
PS C:\Users\ailyn\OneDrive\Escritorio\node-mysql>

```

Ilustración 7. Instalación de mysql

Creamos un archivo llamado “app.js” que es en donde se colocara el código para la conexión de una base de datos con node.js

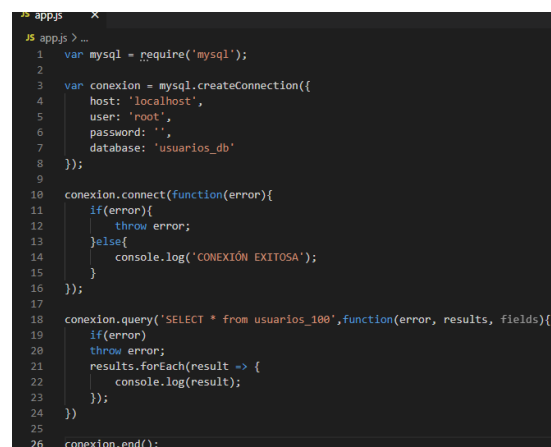
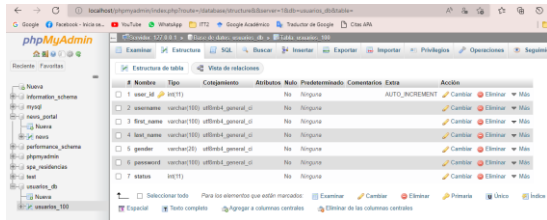


Ilustración 8. Conexión de mysql desde Node.js

Creamos nuestra base de datos llamada “usuarios_db” con la tabla “usuarios_100” y con los atributos que se muestran de la siguiente manera:



#	Nombre	Tipo	Conjuntos	Atributos	Notas	Procedimiento	Comentarios	Extra	Acción
1	user_id	INT(11)	No	Unsigned		AUTO_INCREMENT			Editar
2	username	varchar(100)	offseted_general_0	No	Unsigned				Editar
3	first_name	varchar(100)	offseted_general_0	No	Unsigned				Editar
4	last_name	varchar(100)	offseted_general_0	No	Unsigned				Editar
5	gender	varchar(2)	offseted_general_0	No	Unsigned				Editar
6	password	varchar(100)	offseted_general_0	No	Unsigned				Editar
7	status	INT(11)	No	Unsigned					Editar

Ilustración 9. Creación y estructura de la Base de Datos.

Insertamos datos dentro de la tabla “usuarios_100” para que por medio de una consulta en la terminal de Visual Studio Code, se visualice el contenido.

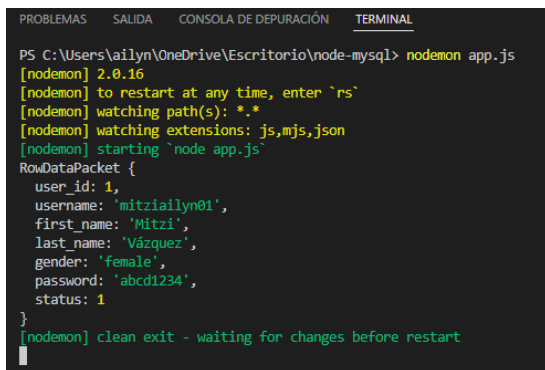


Ejecutar la(s) consulta(s) SQL en la tabla usuarios_db.usuarios_100:

```
INSERT INTO usuarios_100 VALUES('','mitzi11n01','Mitzi','Vázquez','female','abcd1234',1);
```

Ilustración 10. Inserción de datos en la Base de Datos.

Con el comando `nodemon` y el nombre del archivo que tiene la conexión, se visualiza en la terminal la información que contiene la base de datos.



```
PS C:\Users\ailyn\OneDrive\Escritorio\node-mysql> nodemon app.js
[nodemon] 2.0.16
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node app.js'
RowDataPacket {
  user_id: 1,
  username: 'mitziailyn01',
  first_name: 'Mitzi',
  last_name: 'Vázquez',
  gender: 'female',
  password: 'abcd1234',
  status: 1
}
[nodemon] clean exit - waiting for changes before restart
```

Ilustración 11. Conexión exitosa de MySQL con Node.js.

CONCLUSIONES

En conclusión, las aplicaciones de Node.js son escritas en JavaScript, conocido como el lenguaje de la moda, y pueden ejecutarse en OS X, Microsoft Windows y Linux.

Tiene la característica de acometer muchas tareas con poco consumo de recursos, lo que lo hace especialmente

interesante para el desarrollo de servicios de alta concurrencia.

Desde mi punto de vista, Node.js ha significado un gran cambio en las tecnologías digitales de nuestro tiempo.

Por decirlo de algún modo sencillo, Node.js es el lenguaje Javascript sacado del contexto del navegador. Sus creadores supieron valorar la potencia del lenguaje Javascript y usaron el motor "V8" (el motor Javascript open source del navegador Chrome) para crear una plataforma de ejecución capaz de aprovechar el popular lenguaje para acometer una gran variedad de proyectos.

Instalar Node.js es tan sencillo como dirigirse a la página de Node y descargar el ejecutable que permite instalar el motor en nuestro sistema. Para Windows y Linux el proceso es tan sencillo como instalar cualquier otro programa.

REFERENCIA BIBLIOGRÁFICAS

- [1] Kiessling, M. (2016). El Libro para Principiantes en Node.js. 01 Jun 2022, de Ayllu Solar Sitio web: https://ayllusolar.cl/wp-content/uploads/2016/08/node_js_Guia_Principiantes.pdf
- [2] Daniel García Olivares. (2022). Clase lista de quehaceres. Versión de 02 de mayo de 2022.
- [3] Stack Overflow contributors. (2021). Node.js. 03 Jun 2022, de Ebook Sitio web: <https://riptutorial.com/Download/node-js-es.pdf>
- [4] Informática DP. (2020). 7 - Conectar MySQL desde Node.JS. 03 Jun 2022, de Youtube Sitio web: <https://www.youtube.com/watch?v=27JCqmykNdc>