PROGRAMACIÓN BACKEND

Introducción a Node.js

Node.js es un entorno de ejecución para JavaScript construido sobre el motor V8 de Google Chrome. Se lanzó en 2009 por Ryan Dahl, quien buscaba crear un entorno que permitiera ejecutar JavaScript fuera del navegador, especialmente en el servidor.

A diferencia de otros entornos de ejecución, Node.js no solo permite desarrollar aplicaciones en el frontend, sino también en el backend, utilizando el mismo lenguaje: JavaScript.

¿ Qué es Node.js ?

Es un entorno de ejecución que permite a los desarrolladores escribir aplicaciones del lado del servidor en JavaScript. Se diferencia de otros entornos en que está diseñado para ser asíncrono y no bloqueante-

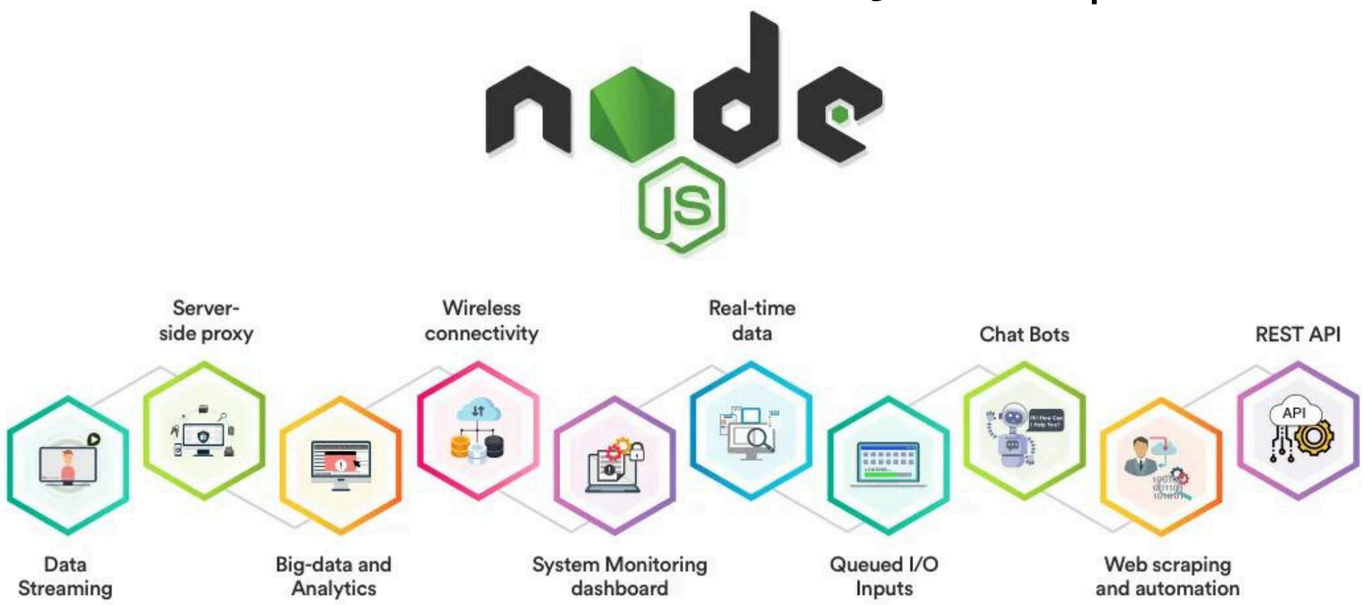
Caracteristicas principales

1. Motor V8 de Google Chrome
2. Asincronía y no bloqueo
3. Single-threaded, pero altamente escalable
4. Sistema de módulos
5. npm (Node Package Manager)

¿Por qué usar Node.js?

Node.js es especialmente útil para

1. Aplicaciones en tiempo real
2. APIs RESTful
3. Aplicaciones de alto rendimiento
4. Desarrollo Full-Stack con JavaScript



Ventajas de Node.js en el backend

1. **E/S no bloqueante**: Permite manejar numerosas conexiones simultáneamente, lo que es ideal para aplicaciones en tiempo real como chats, streaming, y APIs.
2. **JavaScript en el servidor**: Permite a los desarrolladores utilizar un solo lenguaje para el frontend y el backend, facilitando la comunicación y el mantenimiento del código.
3. **Ecosistema de módulos**: Con npm (Node Package Manager), Node.js cuenta con una amplia gama de bibliotecas y módulos que pueden integrarse fácilmente en cualquier proyecto.

Creación de un Servidor Básico con Node.js

Implica establecer un entorno donde Node.js actúe como el backend de una aplicación, capaz de recibir solicitudes (requests) y enviar respuestas (responses) a los clientes, como navegadores web o aplicaciones móviles. Node.js facilita esto utilizando su módulo integrado http, que permite crear y configurar un servidor web simple.

¿Por qué es útil crear un servidor básico con Node.js

- Fundamento para aplicaciones más complejas: Un servidor básico es el primer paso para construir aplicaciones web y APIs más complejas.

- Node.js te da control total sobre cómo manejas las solicitudes y respuestas, lo que lo hace altamente flexible y adecuado para una amplia variedad de aplicaciones.

Guía de instalación de Node.js y Configuración en Visual Studio Code

1. Ve a la página oficial de Node.js y descarga la versión LTS ***https://nodejs.org***
2. Ejecuta el instalador y sigue las instrucciones, asegurándote de que la opción para agregar Node.js al PATH esté seleccionada
3. Verificación de la instalación: Abre una terminal y ejecuta los siguientes comandos para asegurarte de que Node.js y npm se instalaron correctamente:

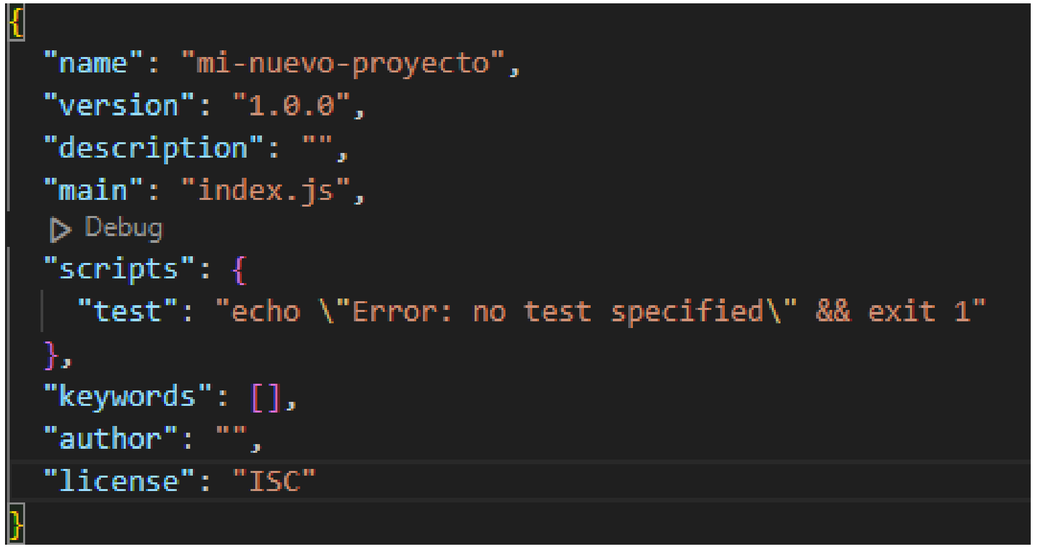
   node -v

   npm -v

Creación de un Servidor Básico con Node.js
En Visual Studio Code

1. Crea un nuevo proyecto, en la ubicación determinada por ustedes, de manera que se creara una carpeta por medio de la terminal ***mkdir name_project***
2. Abre el proyecto desde Visual Studio Code
3. Inicializar un proyecto Node.js:

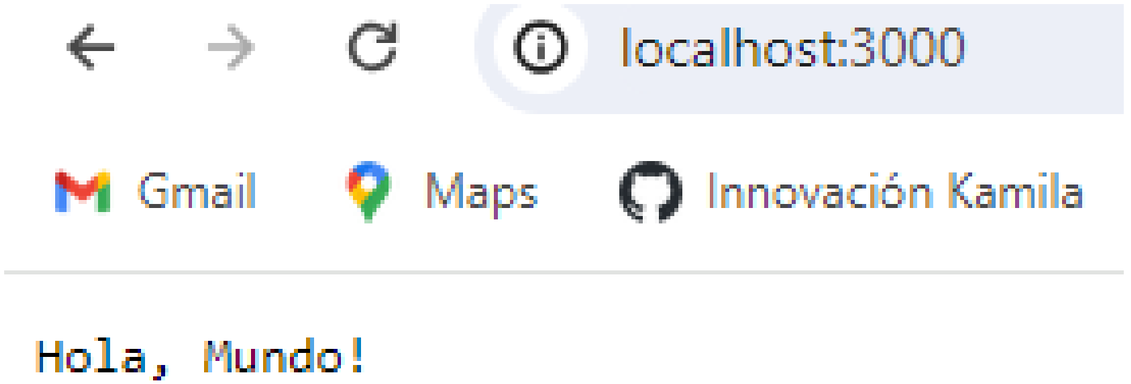
    - En la terminal integrada de Visual Studio Code, navega a la carpeta de tu proyecto y ejecuta: **npm init -y**, esto generará un archivo **package.json**, que es el archivo de configuración para tu proyecto Node.js.



4. En tu proyecto, crea un archivo llamado **server.js** o cualquier nombre que prefieras, en este se utiliza el módulo *http* para crear un servidor. Este servidor puede manejar solicitudes *HTTP* y enviar respuestas de vuelta al cliente.

```
1   const http = require('http');
2
3   const server = http.createServer((req, res) => {
4       res.writeHead(200, { 'Content-Type': 'text/plain' });
5       res.end('Hola, Mundo!\n');
6   });
7
8   const PORT = 3000;
9   server.listen(PORT, () => {
10      console.log(`Servidor escuchando en el puerto ${PORT}`);
11  });
12
```

5. Ejecutar el servidor: En la terminal, ejecuta el servidor con el siguiente comando node **server.js** ve a ***http://localhost:3000*** en tu navegador para ver el servidor en funcionamiento y puede recibir tus peticiones, en este caso sería ***"Hola, Mundo!"***.



Repositorio del manejo del servidor de node.js

Ejemplos adicionales de lo que puedes enviar al servidor y cómo responder a diferentes tipos de solicitudes utilizando el módulo http en Node.js ***https://github.com/Biarcosb/Creaci-n-servidor-Node.js.git***