



# **Curso complementario Desarrollo de Back-end con Node.js - MongoDB**

**ING - DIEGO CASALLAS**

**22810019-1**



[www.sena.edu.co](http://www.sena.edu.co)

# TABLA DE **CONTENIDO**

---



NODE.JS



# NODE.JS

En esta presentación, vamos a ver una introducción a Node.js y la librería Express para crear APIs REST.

Veremos los pasos para crear una API REST, así como un ejemplo práctico.

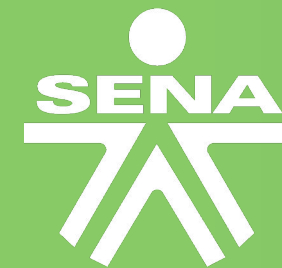




# NODE.JS

## ¿Qué es Node.js?

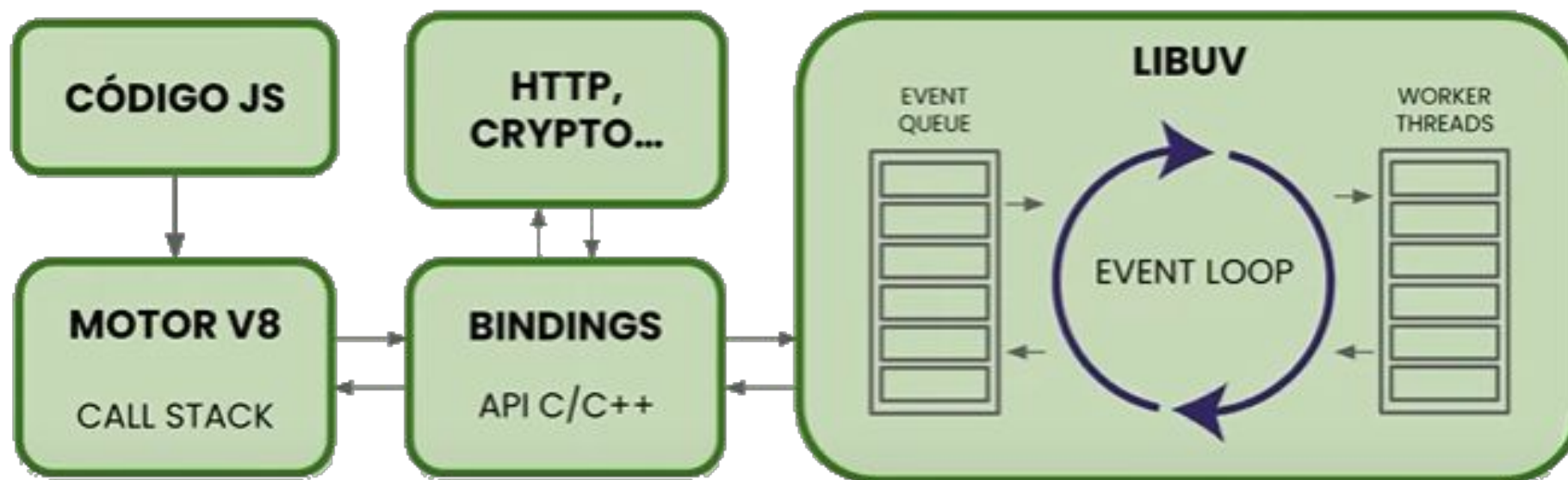
- Un entorno de ejecución de JavaScript de código abierto
- Permite ejecutar JavaScript fuera del navegador
- Basado en el motor V8 de Chrome
- Altamente escalable y eficiente
- Ideal para aplicaciones web en tiempo real y APIs





# NODE.JS

## ¿Cómo funciona Node.js?



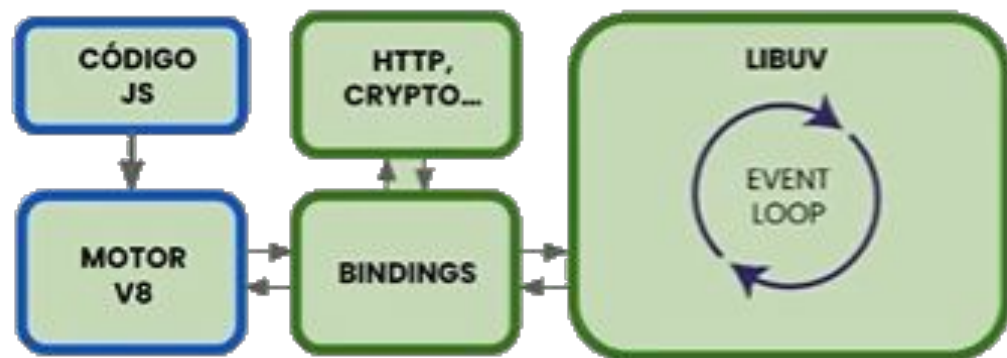
mismo motor



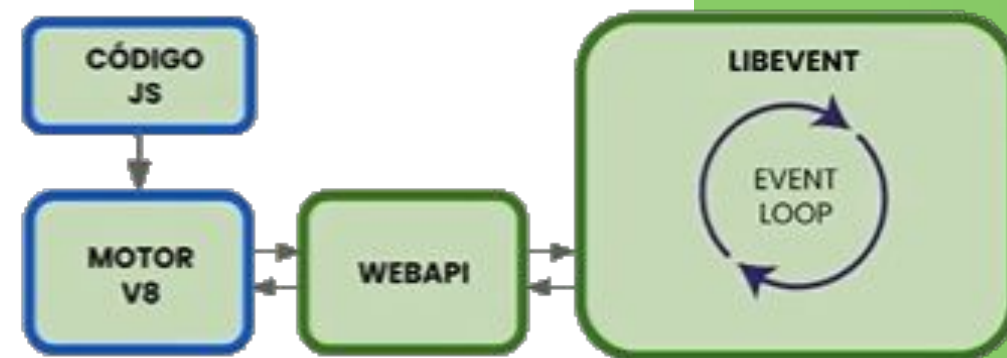


# NODE.JS

## ¿Cómo funciona Node.js?



Node.JS



Chrome



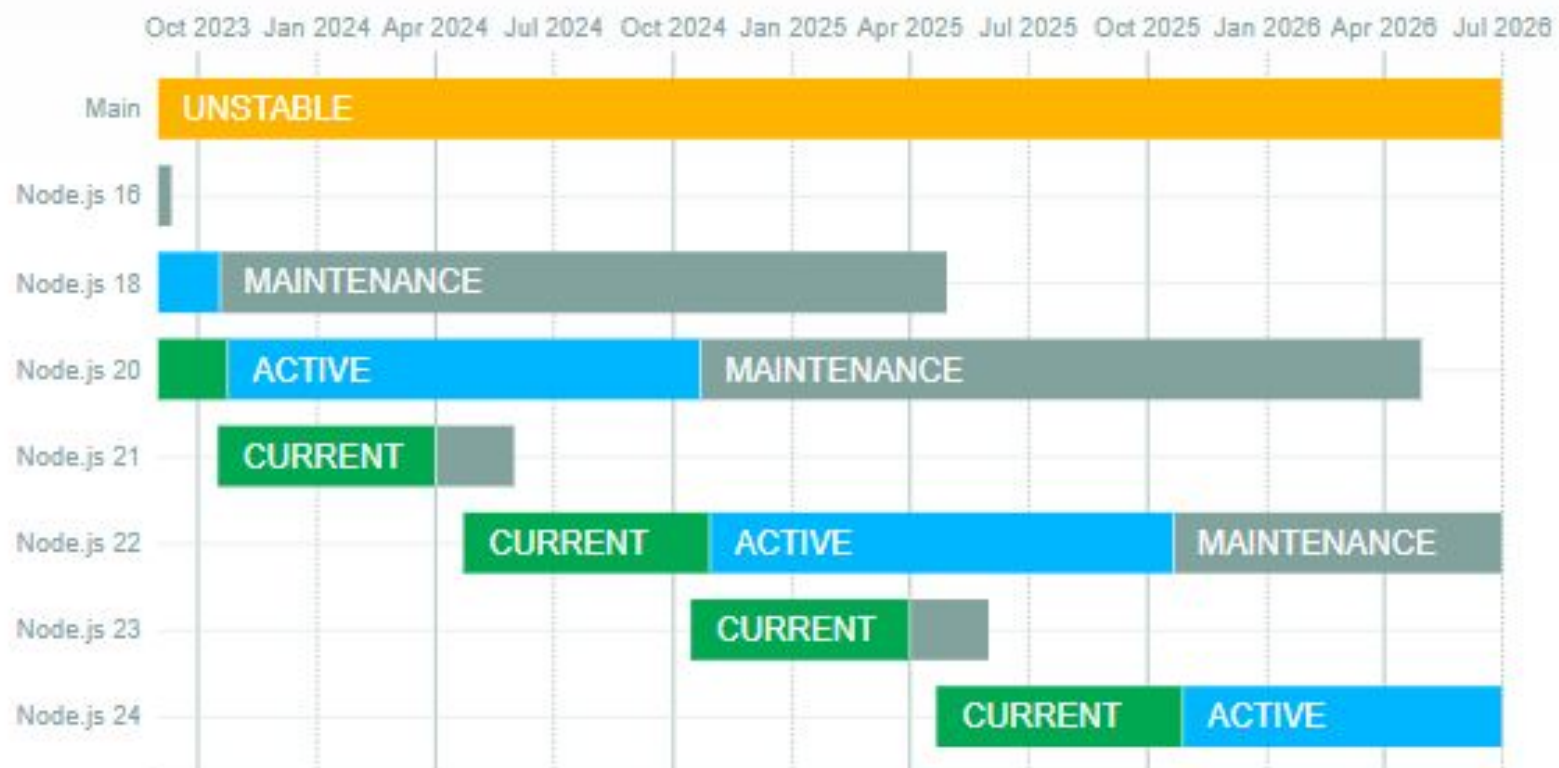


# NODE.JS

## Versión en NODE.JS



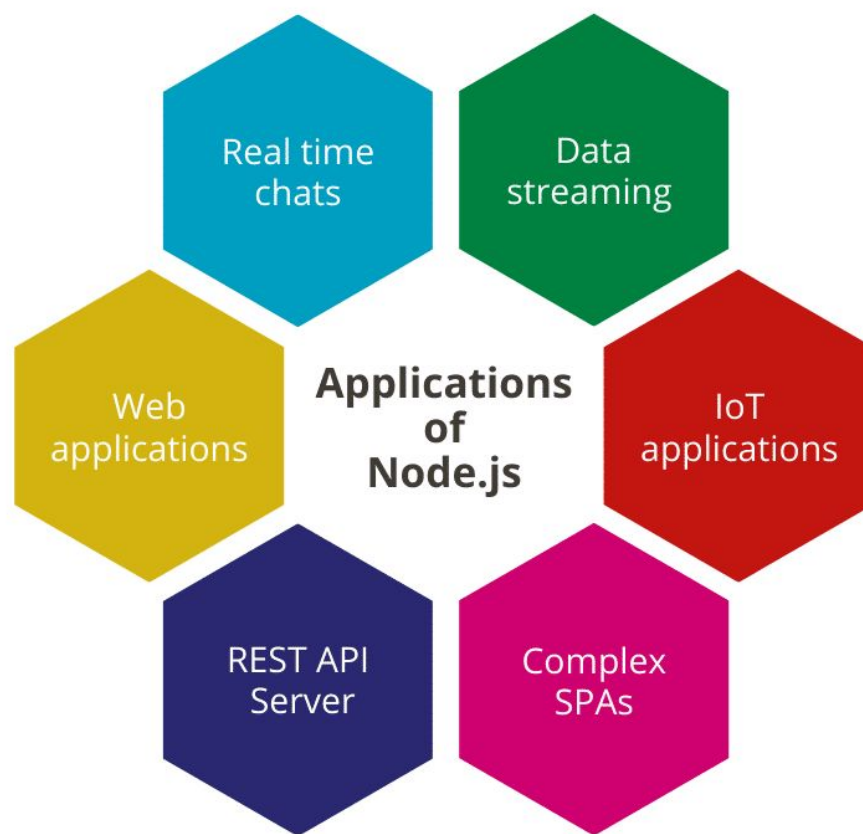
### Release Schedule





# NODE.JS

## ¿Para qué se utiliza Node.js?







# NODE.JS

## ¿Para qué se utiliza Node.js?

- Servidores web
- Aplicaciones web en tiempo real
- Chatbots
- Aplicaciones de IoT
- Herramientas de línea de comandos
- Juegos





# NODE.JS

## Cómo instalar Node.js





# NODE.JS

## Cómo instalar Node.js

- Descargar el instalador de Node.js desde <https://nodejs.org/en/download/prebuilt-installer>
- Ejecutar el instalador y seguir las instrucciones
- Verificar que Node.js está instalado correctamente abriendo una terminal y escribiendo `node -v`





# NODE.JS

## Cómo instalar Node.js

- Descargar el instalador de Node.js desde <https://nodejs.org/en/download/prebuilt-installer>

### Download Node.js®

Download Node.js the way you want.

Package Manager **Prebuilt Installer** Prebuilt Binaries Source Code

I want the  version of Node.js for  running

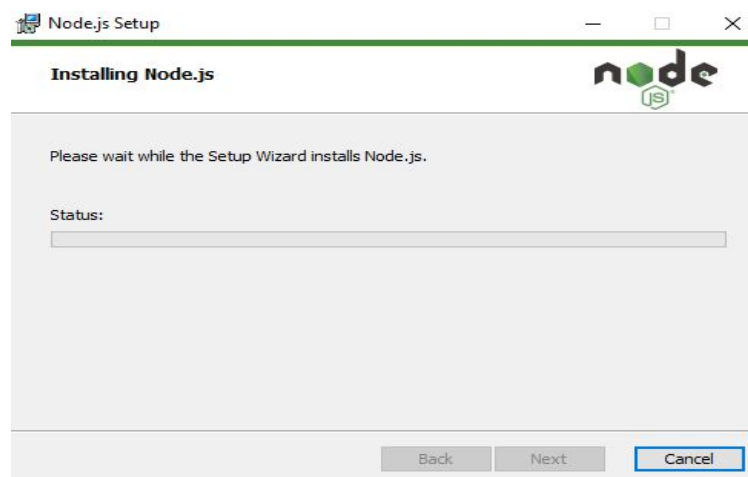
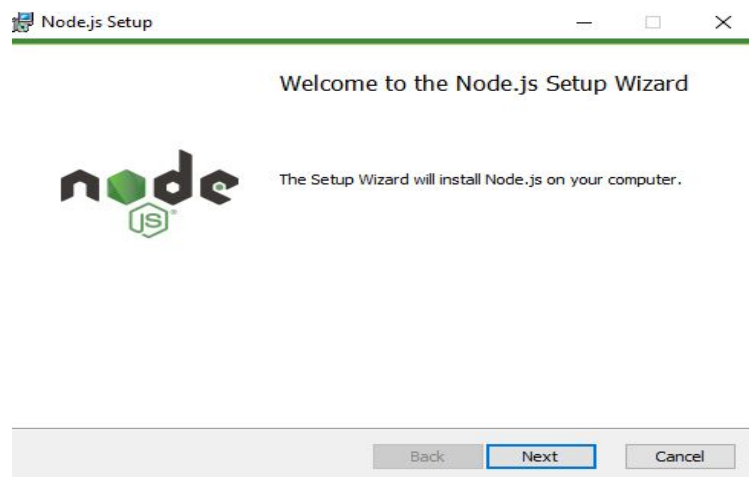
 **Download Node.js v22.2.0**





# NODE.JS

## Cómo instalar Node.js



```
Install Additional Tools for Node.js

Tools for Node.js Native Modules Installation Script

This script will install Python and the Visual Studio Build Tools, necessary
to compile Node.js native modules. Note that Chocolatey and required Windows
updates will also be installed.

This will require about 3 GiB of free disk space, plus any space necessary to
install Windows updates. This will take a while to run.

Please close all open programs for the duration of the installation. If the
installation fails, please ensure Windows is fully updated, reboot your
computer and try to run this again. This script can be found in the
Start menu under Node.js.

You can close this window to stop now. Detailed instructions to install these
tools manually are available at https://github.com/nodejs/node-gyp#on-windows

Presione una tecla para continuar . . .
```





# NODE.JS

## Cómo instalar Node.js

Esperamos que se descarguen las librerías necesarias

Ejecutar la instrucción node -v

```
PS C:\Windows\system32> node -v
v22.2.0
PS C:\Windows\system32>
```





# NODE.JS

## Ejemplo 1: Crear un servidor web simple



JavaScript

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('¡Hola mundo desde Node.js!');
});

server.listen(3000, () => {
  console.log('Servidor escuchando en el puerto 3000');
});
```





# NODE.JS

## Ejemplo 2: Leer un archivo

JavaScript

```
const fs = require('fs');

fs.readFile('archivo.txt', 'utf8', (err, data) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log(data);
});
```







# NODE.JS

## Ejemplo 3: Crear un módulo

JavaScript

```
// modulo.js
module.exports = function saludar(nombre) {
  return `¡Hola ${nombre}!`;
};
```

JavaScript

```
// app.js
const saludar = require('./modulo');

const nombre = 'Juan';
const saludo = saludar(nombre);
console.log(saludo);
```



# NODE.JS

## Conclusión

- Node.js es una herramienta poderosa y versátil que se puede utilizar para crear una amplia variedad de aplicaciones.
- Es fácil de aprender y usar, y tiene una gran comunidad de desarrolladores.
- Si estás buscando una nueva tecnología para aprender, Node.js es una excelente opción.





# NODE.JS

## Actividad

### Ejercicio 1: Leer un archivo y mostrar su contenido

Descripción: Este ejercicio consiste en leer el contenido de un archivo de texto y mostrarlo en la consola.

### Hacer uso:

```
const fs = require('fs');

fs.readFile('archivo.txt', 'utf8', (err, data) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log(data);
});
```





# NODE.JS

## Actividad Explicación:

Este ejercicio lee el contenido del archivo (archivo.txt) y lo imprime en la consola. Utiliza el módulo fs de Node.js para leer archivos. La función fs.readFile() toma tres argumentos:

**archivo:** La ruta del archivo que se desea leer.

**encoding:** La codificación del archivo. En este caso, se utiliza utf8 para leer el archivo como texto plano.

**callback:** Una función que se llamará cuando se haya leído el archivo. La función recibe dos argumentos:

err: Un objeto de error si se produjo un error al leer el archivo.

data: El contenido del archivo como una cadena de texto





# NODE.JS

## Actividad

### Ejercicio 2: Escribir contenido en un archivo de texto

#### Hacer uso:

JavaScript

```
const fs = require('fs');

const archivo = 'archivo.txt';
const contenido = '¡Hola mundo desde Node.js!';

fs.writeFile(archivo, contenido, 'utf8', (err) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log('Archivo escrito correctamente');
});
```



# NODE.JS

## Actividad

### Explicación:

Este ejercicio escribe el contenido ¡Hola mundo desde Node.js! en el archivo archivo.txt. Utiliza el módulo fs de Node.js para escribir archivos. La función fs.writeFile() toma cuatro argumentos:

**archivo:** La ruta del archivo en el que se desea escribir.

**contenido:** El contenido que se desea escribir en el archivo.

**encoding:** La codificación del archivo. En este caso, se utiliza utf8 para escribir el archivo como texto plano.

**callback:** Una función que se llamará cuando se haya escrito el archivo. La función recibe un argumento:

**err:** Un objeto de error si se produjo un error al escribir el archivo.





# NODE.JS

## Actividad

**Ejercicio 2:** Leer, mostrar y agregar contenido de un archivo JSON

**Hacer uso:**

JavaScript

```
const fs = require('fs');  
  
const archivo = 'datos.json';
```





# NODE.JS

## Ejecución

1. La ejecución de los archivos se realiza:
  - llamando la ruta del archivo
  - node “nombre del archivo”

## Ejemplo

```
✓ NODE.JS-MONGO  
> Task_management  
JS example_one.js
```



```
BACK-END\node.js-mongo> node example_one.js
```







# NODE.JS

## Ejecución

2. Ajustar el archivo (example\_one.js) y agregar lo siguiente.

▼ NODE.JS-MONGO

> 📁 Task\_management

JS example\_one.js

```
JS example_one.js
1  console.log("This is example one.");
```

3. Ejecutar nuevamente



```
BACK-END\node.js-mongo> node example_one.js
```

```
This is example one.
```





# GRACIAS

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



@SENAComunica

[www.sena.edu.co](http://www.sena.edu.co)