# NODEJS

Un entorno donde puedes ejecutar JavaScript, orientado a eventos. Utiliza el motor V8 el mismo que utiliza Chrome tanto NodeJS como V8 son de código abierto.

Solo tiene solo un Thread, de un solo hilo o mono-hilo, espera eventos para simular el multi-thread.

Utilizar administrador de versiones de NodeJS NVM, FNM.

Para instalar Node Version Manager (NVM) en Windows, debes seguir los siguientes pasos:

1. Descargar NVM para Windows:

Visita el repositorio oficial de NVM para Windows en GitHub: https://github.com/coreybutler/nvm-windows/releases

- En la sección de "Assets", descarga el archivo "nvm-setup.zip". Asegúrate de descargar la última versión disponible.

2. Instalar NVM:

- Descomprime el archivo ZIP que has descargado en una ubicación de tu elección. Puedes crear una carpeta llamada "nvm" en tu directorio personal o cualquier otro lugar que prefieras.

- Abre la carpeta "nvm" y ejecuta el archivo "nvm-setup.exe" para iniciar el instalador.

- Sigue las instrucciones del instalador para completar la instalación de NVM en tu sistema.

3. Configurar NVM:

- Abre una nueva ventana de línea de comandos (Cmd) o PowerShell para que los cambios surtan efecto.

- Puedes verificar si la instalación se realizó correctamente ejecutando el siguiente comando en tu terminal:

```shell

nvm version

```

- Luego, puedes instalar una versión de Node.js específica utilizando NVM. Por ejemplo, para instalar la última versión LTS de Node.js, ejecuta el siguiente comando:

```shell

nvm install --lts

```

- Puedes establecer la versión de Node.js que desees como la versión predeterminada con el comando:

```shell

nvm use <version>

```

Donde `<version>` es la versión específica de Node.js que deseas usar.

Con estos pasos, deberías haber instalado NVM en tu sistema Windows y estar listo para gestionar múltiples versiones de Node.js. Puedes instalar diferentes versiones de Node.js y cambiar entre ellas según tus necesidades con NVM.

**Usar el REPL, es usar JavaScript por consola: utilizando el comando *node*.**

Window Vs GlobalThis

Como variable global para exclusivamente de Node tenemos *global* para navegadores tenemos *window* y para ambos y mas recomendable usar para Node *globalthis.*

Patrón de diseño módulo

CommonJS Module

|  |
| --- |
| Function sum (a, b) { return a+ b }  // CommonJS Module Export  Module.exports = { sum }  // CommonJS require module  Const { sum } = require(‘./sum’)  Console.log(sum(1, 2)) |

ES Modules (Recomendado)

|  |
| --- |
| // .js -> por defecto utiliza CommonJS  // .mjs -> para utilizar ES Modules  // .cjs -> para utilizar CommonJS  export function sum (a, b) { return a+ b }  import { sum } from ‘./sum.mjs’  Console.log(sum(1, 2)) |

Modulos nativos