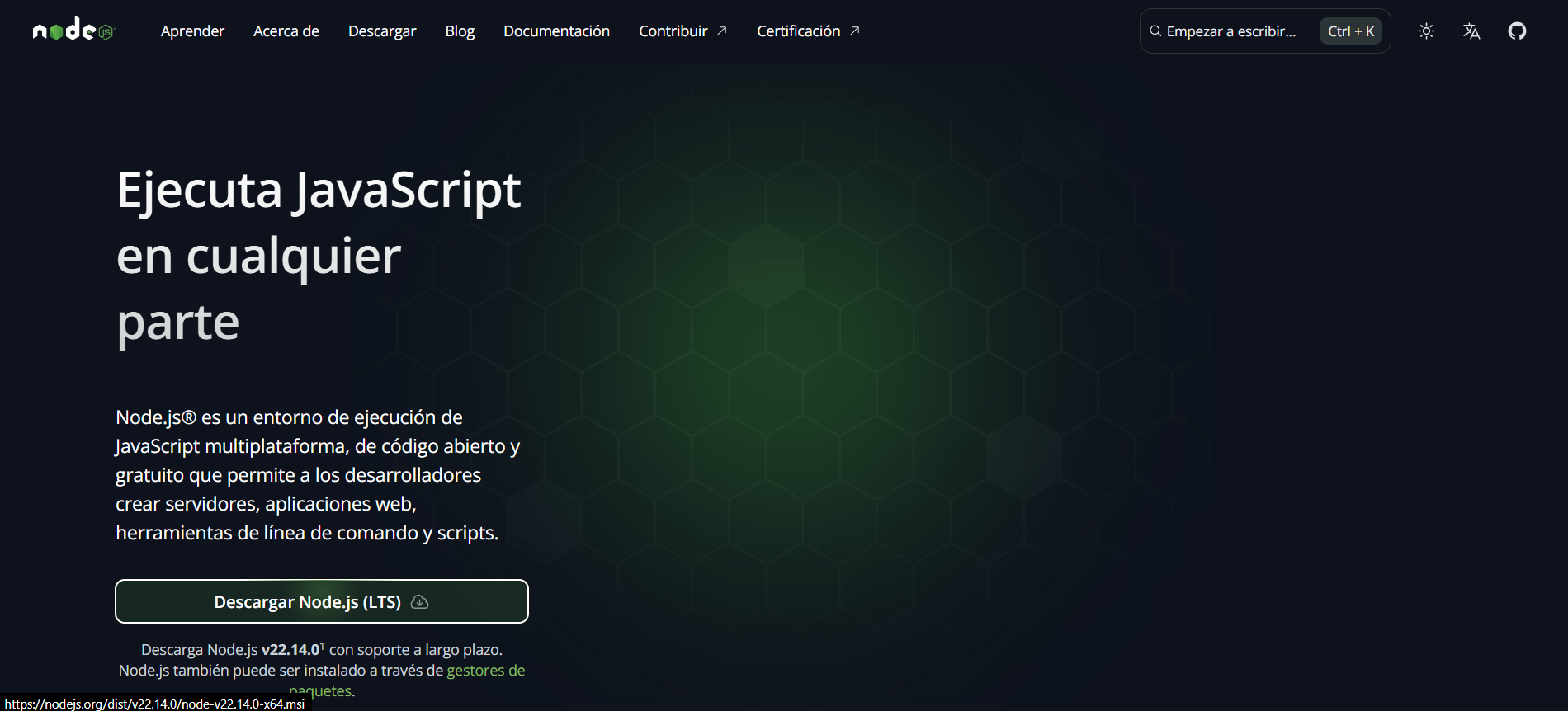
# Trabajo Práctico: Instalación de herramientas

### **Descargar Node.js**

1. Abre tu navegador web y dirígete al sitio oficial de Node.js:<https://nodejs.org>
2. En la página de inicio, verás un boton:
   * **Descargar Node.js LTS (Long Term Support):** Recomendado para la mayoría de los usuarios, ya que es más estable.
3. Descarga la versión que prefieras según tu sistema operativo (Windows, macOS o Linux).



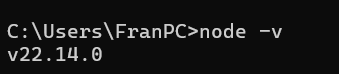
### **Instalar Node.js**

#### **En Windows:**

1. Ejecuta el archivo descargado (.msi).
2. Sigue el asistente de instalación:
   * Acepta los términos y condiciones.
   * Selecciona las opciones predeterminadas.
   * Espera a que finalice la instalación.
3. Una vez instalado, abre la terminal (cmd o PowerShell) y ejecuta:

node -v

Si muestra un número de versión, significa que Node.js se ha instalado correctamente.



npm -v

Si muestra un número de versión, significa que npm se ha instalado correctamente.



#### **En macOS:**

1. Abre el archivo .pkg descargado y sigue los pasos del asistente.
2. Una vez completada la instalación, abre la Terminal y ejecuta:

node -v

Si muestra una versión, la instalación fue exitosa.

#### **En Linux (Ubuntu/Debian):**

1. Abre la terminal y ejecuta los siguientes comandos para instalar Node.js desde el repositorio oficial

curl -fsSL https://deb.nodesource.com/setup\_lts.x | sudo -E bash -

sudo apt install -y nodejs

1. Verifica la instalación con:

node -v

1. Verificar la Instalación

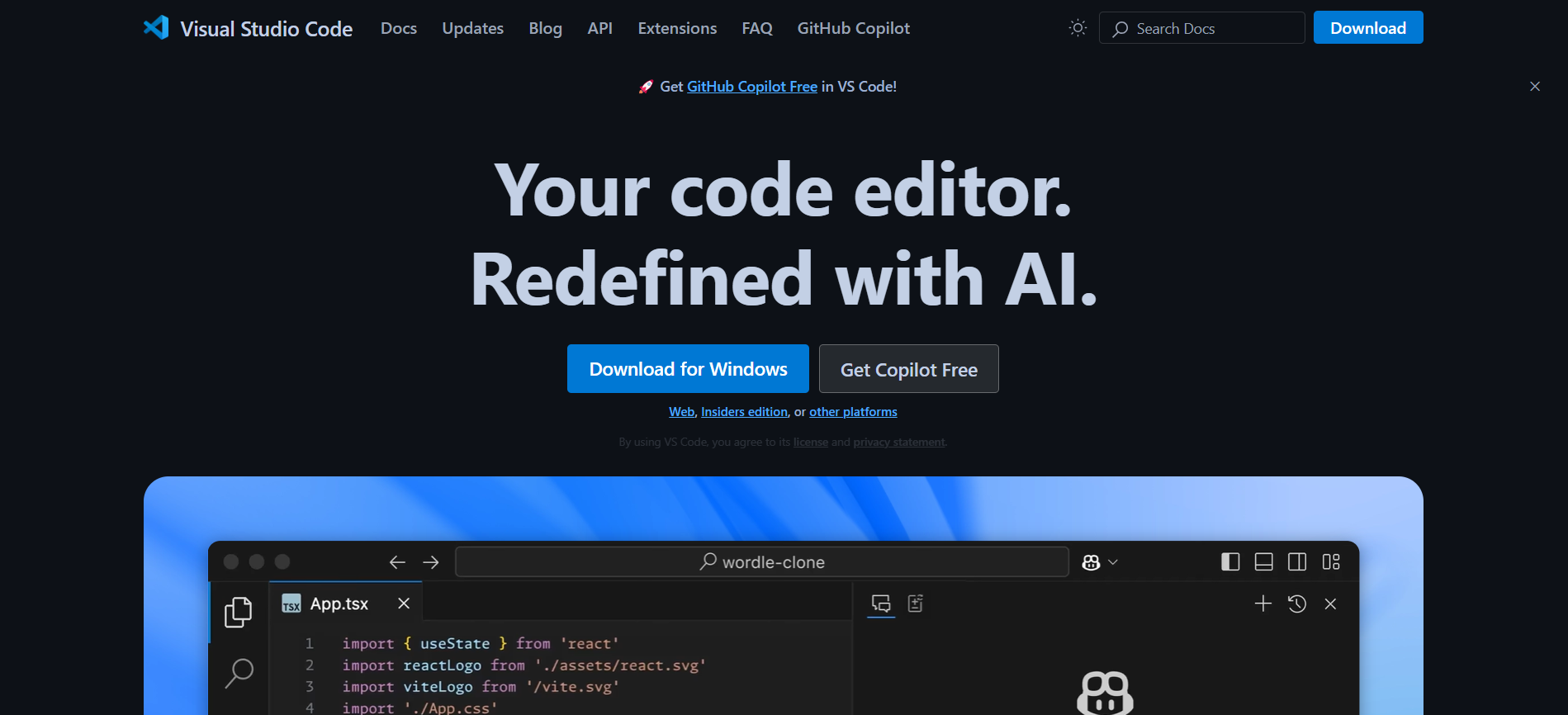
Después de la instalación, verifica también la versión de npm (Node Package Manager) con:

npm -v

Si ves un número de versión, significa que npm también se instaló correctamente.

**Descargar e Instalar Visual Studio Code**

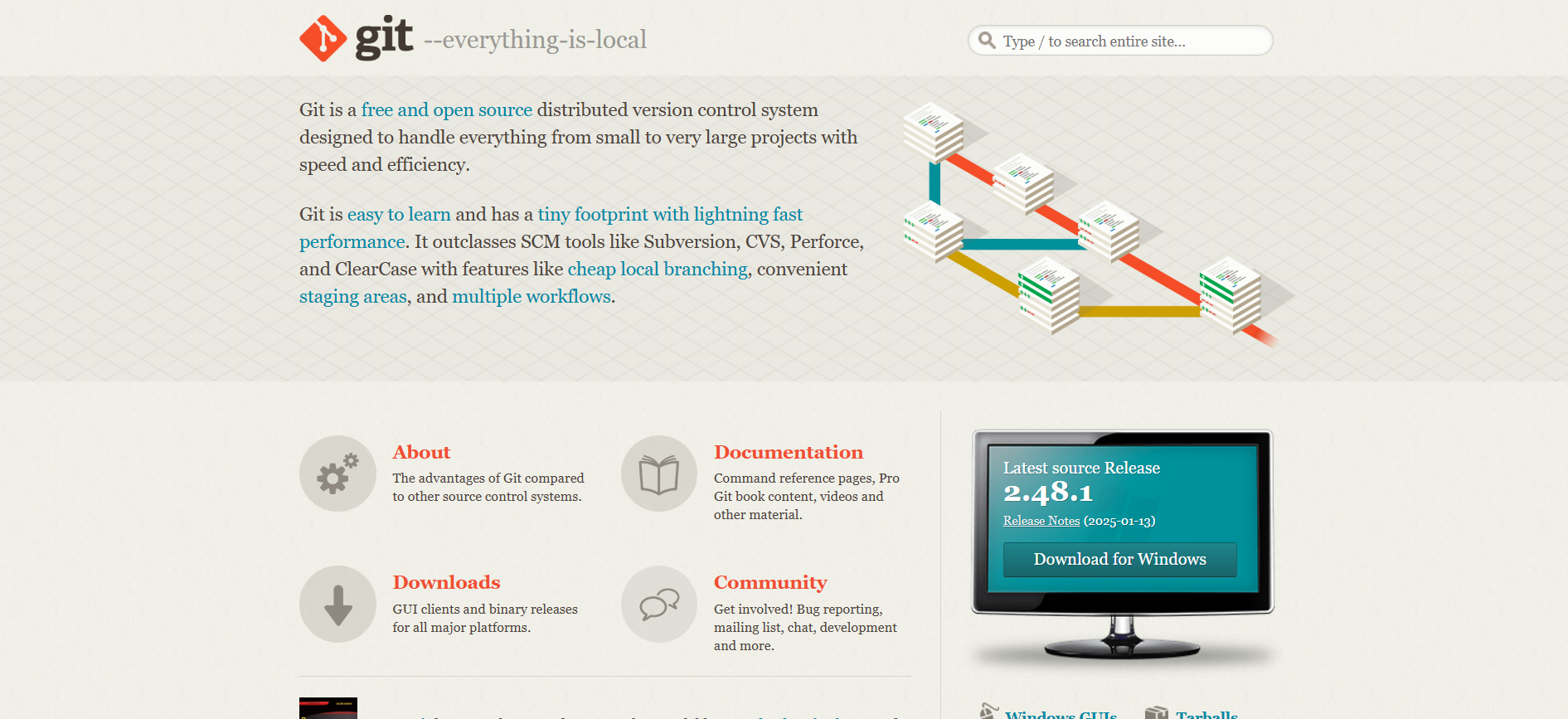
1. Abre tu navegador web y dirígete al sitio oficial de Visual Studio Code:<https://code.visualstudio.com>



1. Haz clic en el botón de descarga y selecciona la versión correspondiente a tu sistema operativo (Windows, macOS o Linux).
2. Una vez descargado el archivo de instalación:
   * **En Windows:** Ejecuta el archivo .exe, acepta los términos y sigue el asistente de instalación.
   * **En macOS:** Abre el archivo .zip descargado y mueve Visual Studio Code a la carpeta Aplicaciones.
   * **En Linux:** Dependiendo de la distribución, instala el archivo .deb o .rpm usando la terminal.
3. Abre Visual Studio Code y verifica que funciona correctamente.

### **Descargar e Instalar Git**

1. Abre tu navegador web y dirígete al sitio oficial de Git:<https://git-scm.com>



1. Descarga la versión adecuada para tu sistema operativo (Windows, macOS o Linux).
2. Instala Git siguiendo los pasos correspondientes:
   * **En Windows:** Ejecuta el instalador .exe, acepta los términos y sigue el asistente dejando las opciones predeterminadas.
   * **En macOS:** Instala Git usando Homebrew ejecutando en la terminal:

brew install git

* + **En Linux (Ubuntu/Debian):** Ejecuta en la terminal:  
    sudo apt install git

1. Verifica la instalación ejecutando en la terminal:  
   git --version

Si ves un número de versión, significa que Git se ha instalado correctamente.

### **Descargar e Instalar MongoDB Compass**

1. Abre tu navegador web y dirígete al sitio oficial de MongoDB Compass:<https://www.mongodb.com/try/download/compass>
2. Descarga la versión adecuada para tu sistema operativo (Windows, macOS o Linux).
3. Instala MongoDB Compass siguiendo los pasos correspondientes:
   * **En Windows:** Ejecuta el instalador .exe, acepta los términos y sigue el asistente de instalación.
   * **En macOS:** Abre el archivo .dmg descargado y mueve MongoDB Compass a la carpeta Aplicaciones.
   * **En Linux:** Dependiendo de la distribución, instala el archivo .deb o .rpm usando la terminal.
4. Abre MongoDB Compass y verifica que funciona correctamente.

### **Descargar e Instalar Docker Desktop**

1. Abre tu navegador web y dirígete al sitio oficial de Docker Desktop: https://www.docker.com/products/docker-desktop
2. Descarga la versión adecuada para tu sistema operativo (Windows o macOS).
3. Instala Docker Desktop siguiendo los pasos correspondientes:
   * **En Windows:** Ejecuta el instalador .exe, acepta los términos y sigue el asistente de instalación. Asegúrate de habilitar la opción de integración con WSL 2 si usas Windows.
   * **En macOS:** Abre el archivo .dmg descargado y arrastra Docker a la carpeta Aplicaciones.
4. Inicia Docker Desktop y espera a que se configure correctamente.
5. Verifica la instalación abriendo una terminal y ejecutando:  
   docker --version  
   Si muestra un número de versión, significa que Docker se ha instalado correctamente.

**Actividad**

Responder las siguientes preguntas sobre node

1. ¿Qué es Node.js y en qué se diferencia de JavaScript en el navegador?
2. ¿Qué son las versiones LTS?
3. Explica la diferencia entre un proceso síncrono y uno asíncrono
4. ¿Cómo se maneja la asincronía en Node.js? Explica las diferencias entre callbacks, Promises y async/await.
5. ¿Qué es el objeto global en Node.js y en qué se diferencia de window en el navegador?
6. ¿Node.js es multithreading?
7. ¿Qué es el Event Loop en Node.js y cuál es su papel en la ejecución de código asíncrono?
8. Mencionar tres ventajas de Node.js
9. menciona tres desventajas de Node.js.
10. ¿Podrías nombrar algunas bibliotecas que los desarrolladores utilizan frecuentemente con Node.js?

RESPUESTAS:

**1. ¿Qué es Node.js y en qué se diferencia de JavaScript en el navegador?**

Node.js es un entorno de ejecución de JavaScript basado en el motor V8 de Chrome, que permite ejecutar JavaScript fuera del navegador.

🔹 **Diferencias con JS en el navegador:**

* En el navegador, JavaScript se usa principalmente para manipular el DOM y manejar eventos.
* En Node.js, se usa para crear servidores, manejar archivos, conectarse a bases de datos y más.
* Node.js no tiene objetos como window o document, pero tiene global y módulos como fs para manejar archivos.

**2. ¿Qué son las versiones LTS?**

LTS (*Long-Term Support*) son versiones de Node.js que reciben soporte a largo plazo, incluyendo correcciones de seguridad y estabilidad durante al menos 30 meses.

🔹 **Ejemplo:**

* Una versión actual de Node.js puede ser 20.x, pero la versión LTS estable es 18.x.
* Se recomienda usar versiones LTS en proyectos en producción.

**3. Explica la diferencia entre un proceso síncrono y uno asíncrono**

🔹 **Síncrono:**

* Se ejecutan **línea por línea**, bloqueando el código hasta completar una tarea.
* Ejemplo:

javascript

Copiar código

console.log("Inicio");

const data = fs.readFileSync("archivo.txt", "utf8"); // Bloquea hasta leer el archivo

console.log(data);

console.log("Fin");

🔹 **Asíncrono:**

* No bloquea la ejecución; permite continuar con otras tareas mientras espera el resultado.
* Ejemplo:

javascript

Copiar código

console.log("Inicio");

fs.readFile("archivo.txt", "utf8", (err, data) => {

console.log(data);

});

console.log("Fin");

Aquí "Fin" se imprimirá antes que el contenido del archivo porque la lectura es asíncrona.

**4. ¿Cómo se maneja la asincronía en Node.js?**

Se maneja con **callbacks, Promises y async/await**.

🔹 **Callbacks:** Funciones pasadas como argumento que se ejecutan cuando la tarea se completa.

javascript

Copiar código

fs.readFile("archivo.txt", "utf8", (err, data) => {

if (err) console.error(err);

console.log(data);

});

🔹 **Promises:** Objetos que representan un valor disponible ahora, en el futuro o nunca.

javascript

Copiar código

fs.promises.readFile("archivo.txt", "utf8")

.then(data => console.log(data))

.catch(err => console.error(err));

🔹 **async/await:** Sintaxis más limpia para trabajar con Promises.

javascript

Copiar código

async function leerArchivo() {

try {

const data = await fs.promises.readFile("archivo.txt", "utf8");

console.log(data);

} catch (err) {

console.error(err);

}

}

leerArchivo();

**5. ¿Qué es el objeto global en Node.js y en qué se diferencia de window en el navegador?**

🔹 **global en Node.js**

* Es el equivalente a window en el navegador, pero para entornos fuera del navegador.
* Contiene funciones globales como console.log(), setTimeout(), process, etc.

🔹 **Diferencias con window:**

* window contiene el DOM y APIs del navegador (document, localStorage, etc.), mientras que global en Node.js maneja cosas como módulos y procesos del sistema.

**6. ¿Node.js es multithreading?**

Node.js **es single-threaded**, pero usa un modelo asíncrono basado en eventos.

Sin embargo, Node.js tiene el **módulo worker\_threads**, que permite crear múltiples hilos cuando es necesario para tareas pesadas.

**7. ¿Qué es el Event Loop en Node.js y cuál es su papel en la ejecución de código asíncrono?**

El **Event Loop** es el mecanismo que permite a Node.js manejar operaciones asíncronas sin bloquear el hilo principal.

🔹 **Ciclo del Event Loop:**

1. Ejecuta código síncrono.
2. Atiende operaciones asíncronas (callbacks, Promises, timers, etc.).
3. Repite el proceso en cada iteración del loop.

**8. Mencionar tres ventajas de Node.js**

✅ **Rápido y eficiente:** Usa el motor V8 y un modelo no bloqueante.  
✅ **Gran comunidad y ecosistema:** Miles de paquetes disponibles en npm.  
✅ **Cross-platform:** Funciona en Windows, Linux y macOS sin problemas.

**9. Mencionar tres desventajas de Node.js**

❌ **Single-threaded por defecto:** No es ideal para tareas CPU-intensivas.  
❌ **Callbacks anidados:** Puede generar el problema del *callback hell*.  
❌ **Manejo de memoria:** Puede ser menos eficiente en comparación con otros lenguajes.

**10. ¿Podrías nombrar algunas bibliotecas que los desarrolladores utilizan frecuentemente con Node.js?**

📌 **Express.js:** Framework para construir servidores web.  
📌 **Mongoose:** ODM para trabajar con MongoDB.  
📌 **bcrypt:** Para encriptación de contraseñas.  
📌 **dotenv:** Manejo de variables de entorno.  
📌 **axios:** Cliente HTTP para hacer peticiones a APIs.

**📌 Conclusión**

Node.js es un entorno potente para ejecutar JavaScript fuera del navegador, con un modelo asíncrono basado en eventos y una comunidad muy activa. Su uso es ideal para servidores web, APIs y aplicaciones en tiempo real.