

EJERCICIOS PRÁCTICOS TEMA 1

1. Documentate sobre las siguientes herramientas y crea y documenta un manual de instalación y uso de estas:

a. Node.js

Node.js es un entorno de tiempo de ejecución multiplataforma de JavaScript que te permite ejecutar JavaScript en el lado del servidor. Aquí tienes un manual básico de instalación y uso de Node.js:

Instalación:

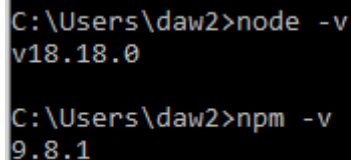
1. Ve al sitio web oficial de Node.js en <https://nodejs.org/>.
2. Descarga la versión LTS (Long Term Support) recomendada para tu sistema operativo (Windows, macOS o Linux).
3. Sigue las instrucciones de instalación en pantalla para completar la instalación.

Uso:

1. Abre tu terminal o línea de comandos.
2. Verifica la instalación de Node.js y npm ejecutando los siguientes comandos:

```
node -v
```

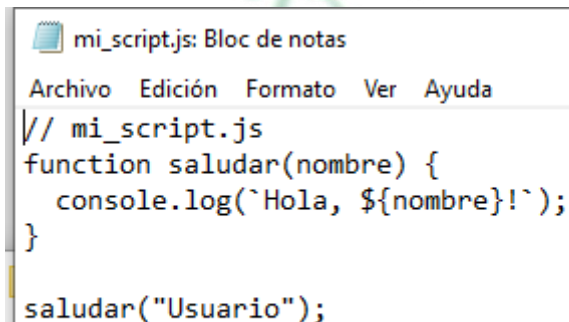
```
npm -v
```



```
C:\Users\daw2>node -v
v18.18.0

C:\Users\daw2>npm -v
9.8.1
```

3. Crea un archivo JavaScript (por ejemplo, "mi_script.js") y escribe tu código JavaScript en él.



```
mi_script.js: Bloc de notas
Archivo Edición Formato Ver Ayuda
// mi_script.js
function saludar(nombre) {
  console.log(`Hola, ${nombre}!`);
}

saludar("Usuario");
```

4. Ejecuta tu archivo JavaScript con Node.js usando el siguiente comando:

`node mi_script.js`

```
C:\Users\daw2>node mi_script.js
Hola, Usuario!
C:\Users\daw2>
```

b. Nvm

Nvm es una herramienta que te permite gestionar múltiples versiones de Node.js en tu sistema. Aquí tienes un manual básico de instalación y uso de Nvm:

Instalación:

1. Abre tu navegador web y ve al repositorio oficial de NVM para Windows en GitHub: <https://github.com/coreybutler/nvm-windows>.
2. En la página de GitHub, desplázate hacia abajo hasta encontrar la sección "Getting Started". Desde allí, puedes descargar el instalador de NVM para Windows. Debería haber un enlace con el nombre "Download Now" o similar.
3. Haz clic en el enlace de descarga para obtener el instalador.
4. Ejecuta el instalador descargado. Durante la instalación, verás una ventana de configuración que te permitirá personalizar algunas opciones. Puedes dejar las opciones predeterminadas o ajustarlas según tus preferencias.
5. Sigue las instrucciones del instalador para completar la instalación de NVM en tu sistema Windows.
6. Una vez finalizada la instalación, abre una nueva ventana de línea de comandos.
7. Para verificar que NVM se instaló correctamente, ejecuta el siguiente comando en la línea de comandos:

`nvm -version`

```
8.15.0
C:\WINDOWS\system32>nvm --version
1.1.11
C:\WINDOWS\system32>
```

Uso Básico:

1. Para instalar una versión específica de Node.js, utiliza el siguiente comando (por ejemplo, para instalar Node.js v8.15.0):

nvm install 8.15.0

```
C:\WINDOWS\system32>nvm install 8.15.0
Downloading node.js version 8.15.0 (64-bit)...
Complete
Creating C:\Users\admin\AppData\Roaming\nvm\temp

Downloading npm version 6.4.1... Complete
Installing npm v6.4.1...

Installation complete. If you want to use this version, type

nvm use 8.15.0

C:\WINDOWS\system32>nvm list

* 18.18.0 (Currently using 64-bit executable)
  8.15.0
```

2. Para usar una versión específica de Node.js, ejecuta el siguiente comando (por ejemplo, para usar Node.js v8.15.0):

nvm use 8.15.0

```
C:\WINDOWS\system32>nvm use 8.15.0
Now using node v8.15.0 (64-bit)

C:\WINDOWS\system32>
```

c. Npm

Npm es el administrador de paquetes de Node.js que te permite descargar y gestionar bibliotecas y módulos de JavaScript. Aquí tienes un manual básico de instalación y uso de npm:

Instalación:

Npm se instala automáticamente cuando instalas Node.js. No es necesario instalarlo por separado.

Uso Básico:

Supongamos que deseas instalar un paquete llamado "lodash", que es una biblioteca popular para realizar operaciones en objetos y arreglos en JavaScript.

Sigue estos pasos:

1. Abre tu terminal o línea de comandos.
2. Navega hasta el directorio de tu proyecto si no estás allí. Puedes usar el comando `cd` para cambiar de directorio.
3. Una vez que estés en el directorio de tu proyecto, ejecuta el siguiente comando para instalar el paquete "lodash":
`npm install lodash`
4. npm descargará e instalará automáticamente el paquete "lodash" y todas sus dependencias en tu proyecto.

```
C:\WINDOWS\system32>npm install lodash
added 1 package in 5s
npm notice
npm notice New major version of npm available! 9.8.1 -> 10.1.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.1.0
npm notice Run npm install -g npm@10.1.0 to update!
npm notice
C:\WINDOWS\system32>
```

2. Realiza un trabajo de investigación sobre la historia de JavaScript y céntrate en la estandarización del lenguaje, los distintos estándares y la situación actual del mismo.

JavaScript: Una Breve Historia

- **Década de 1990:** JavaScript fue creado por Brendan Eich en 1995 mientras trabajaba en Netscape Communications Corporation. Originalmente, se llamaba "Mocha", pero luego se renombró a "LiveScript" antes de adoptar el nombre "JavaScript". Su objetivo inicial era proporcionar interactividad en las páginas web.
- **1996:** Microsoft lanzó su propia versión de JavaScript llamada "JScript" en Internet Explorer, lo que llevó a una fragmentación del lenguaje y la necesidad de estándares.
- **1997:** Netscape presentó JavaScript a Ecma International (anteriormente conocida como European Computer Manufacturers Association) para su estandarización. El estándar resultante se llamó "ECMAScript" en honor a Ecma International. ECMAScript 1 (ES1) se convirtió en la primera especificación oficial de JavaScript en junio de 1997.
- **2000-2005:** Durante este período, ECMAScript 3 (ES3) se convirtió en la versión más ampliamente adoptada de JavaScript en los navegadores. Introdujo mejoras y correcciones de errores en el lenguaje.

Desarrollo y Estandarización Continua

- **2005-2008:** Se formó un comité técnico dentro de Ecma International para revisar y actualizar ECMAScript. Se publicó la sexta edición (ES4), pero fue controvertida y no se

implementó ampliamente en los navegadores debido a diferencias significativas con ES3.

- **2009:** Se lanzó ECMAScript 5 (ES5), que trajo mejoras importantes y funciones adicionales, como "Object.create" y "Array.forEach". ES5 se convirtió en la versión dominante durante varios años.
- **2011:** Con el nombre de "ECMAScript Harmony" y luego "ECMAScript 6" (ES6), se introdujo una revisión importante que agregó muchas características nuevas, incluyendo clases, módulos y funciones de flecha.
- **2015:** ECMAScript 6 (ES6) se convirtió en ECMAScript 2015, marcando un nuevo enfoque en la estandarización de versiones anuales en lugar de esperar años entre lanzamientos.
- **2015-2021:** Se lanzaron versiones adicionales de ECMAScript cada año, lo que permitió una evolución continua del lenguaje. Estas versiones incluyeron características como "async/await", "destructuring", "spread/rest operators" y muchas más.

La Situación Actual

Hoy en día, JavaScript es un lenguaje de programación fundamental para el desarrollo web y más allá. La estandarización continua y la adopción de prácticas recomendadas han hecho que JavaScript sea más robusto y confiable.

Los navegadores modernos son compatibles con las últimas versiones de ECMAScript, lo que permite a los desarrolladores aprovechar las características más nuevas en sus aplicaciones web. Además, JavaScript también se ha expandido más allá del navegador y se utiliza en servidores (Node.js), aplicaciones móviles y más, lo que lo convierte en uno de los lenguajes de programación más versátiles y ampliamente utilizados en la actualidad.

3. Busca información sobre la guerra de los navegadores.

La "Guerra de los Navegadores" fue un período en la historia de la tecnología que tuvo lugar en la década de 1990 y principios de la década de 2000. Esta batalla se libró entre varios fabricantes de navegadores web por la supremacía en el mercado de navegadores. La lucha se centró en dos actores principales: Netscape Communications Corporation, que desarrolló Netscape Navigator, y Microsoft Corporation, que lanzó Internet Explorer.

1. **Auge de Internet:** A medida que Internet ganaba popularidad en la década de 1990, la necesidad de un software que permitiera a las personas explorar la web se volvió evidente. Las páginas web eran en su mayoría estáticas y simples en ese momento.
2. **La Entrada de Netscape:** En 1994, Netscape Communications lanzó Netscape Navigator, uno de los primeros navegadores web populares. Netscape Navigator permitió a los usuarios experimentar la web de manera más visual y dinámica, lo que lo convirtió en un éxito instantáneo.

3. **La Respuesta de Microsoft:** Viendo el potencial de la web, Microsoft lanzó Internet Explorer (IE) en 1995. Internet Explorer 3.0, lanzado en 1996, marcó el inicio de la competencia directa con Netscape Navigator.
4. **La Primera Batalla:** La verdadera batalla comenzó con el lanzamiento de Internet Explorer 3.0. Microsoft comenzó a ganar terreno rápidamente al integrar IE con su sistema operativo Windows, lo que lo convirtió en el navegador predeterminado en la mayoría de las PC.
5. **La Estrategia de Microsoft:** Microsoft adoptó una estrategia agresiva al vincular IE con Windows, lo que dificultó que los usuarios optaran por otros navegadores. Esto llevó a acusaciones de prácticas anticompetitivas.
6. **La 'Guerra del Explorador':** La lucha se intensificó con el lanzamiento de nuevas versiones de ambos navegadores. Cada compañía intentó superar a la otra mediante la adición de características y mejoras constantes.
7. **La Victoria de Internet Explorer:** A finales de la década de 1990, Internet Explorer emergió como el navegador dominante con una cuota de mercado abrumadora. Netscape Navigator comenzó a perder terreno.
8. **Las Secuelas:** La batalla tuvo consecuencias significativas. Microsoft enfrentó juicios antimonopolio por su conducta anticompetitiva, y Netscape Communications fue adquirida por AOL.
9. **La Nueva Competencia:** A pesar de la victoria de IE, la competencia no desapareció. Mozilla Firefox surgió como una alternativa de código abierto en 2004 y Google Chrome se lanzó en 2008.
10. **Evolución de la Web:** La intensa competencia durante la "Guerra de los Navegadores" llevó a una rápida innovación en tecnología web y la adopción de estándares más sólidos. Esto allanó el camino para la evolución continua de la web y la creación de aplicaciones web más avanzadas.

La "Guerra de los Navegadores" dejó un legado duradero en la forma en que usamos Internet y cómo se desarrollan los navegadores web. La competencia entre los navegadores ha seguido impulsando la innovación y garantizando estándares web más sólidos en beneficio de los usuarios y desarrolladores de todo el mundo.