

Pràctica 1:

Crear l'entorn de desenvolupament

Autor: Roger Muntané
Professor: Alfredo Ballesteró
Data: 25/9/2025

Index:

1. Crear l'entorn de treball	3
1. Activa el firewall del sistema:	3
2. Instal·la la versió LTS	3
3. Instal·la l'editor de codi VSCode.	3
4. Crear un projecte amb Vite.	3
5. Inicia el servidor de desenvolupament	5
2. Deshabilitar JavaScript	6
3. Executar JavaScript en el terminal	8
4. Debug en el Navegador i en VSCode	10

1. Crear l'entorn de treball

1. Activa el firewall del sistema:

```
roger@RogerUbuntu:~$ sudo ufw enable
[sudo] contrasenya para roger:
El cortafuegos está activo y habilitado en el arranque del sistema
roger@RogerUbuntu:~$ sudo ufw status verbose
Estado: activo
Acceso: on (low)
Predeterminado: deny (entrantes), allow (salientes), deny (enrutados)
Perfiles nuevos: skip
```

2. Instal·la la versió LTS

El primer pas consisteix en instal·lar la versió LTS (Long-Term Support) de Node.js, que ofereix estabilitat i suport prolongat. Aquesta instal·lació inclou npm (Node Package Manager), l'eina fonamental per gestionar les dependències del projecte. És crucial utilitzar la versió LTS per garantir la compatibilitat amb les llibreries i frameworks moderns.

```
npm install --global npm
roger@RogerUbuntu:~$ nvm install 22
v22.19.0 is already installed.
Now using node v22.19.0 (npm v10.9.3)
roger@RogerUbuntu:~$ node -v
v22.19.0
```

3. Instal·la l'editor de codi VSCode.

```
roger@RogerUbuntu:~/M6$ code -v
1.103.2
6f17636121051a53c88d3e605c491d22af2ba755
x64
```

4. Crear un projecte amb Vite.

Seguidament, s'utilitza Vite per crear l'estructura inicial del projecte, ja que ofereix una configuració ràpida i un servidor de desenvolupament optimitzat. Vite genera automàticament els arxius bàsics i la configuració necessària.

```
roger@RogerUbuntu:~/M6$ npm create vite @latest ex1
Need to install the following packages:
create-vite@7.1.3
Ok to proceed? (y) y
```

```
> npx
> create-vite @latest ex1
```

```
◇ Package name:
  -ex1
```

```
◇ Select a framework:
  Vanilla
```

```
◇ Select a variant:
  JavaScript
```

```
◇ Scaffolding project in /home/roger/M6/@latest...
```

```
Done. Now run:
```

```
cd @latest
npm install
```

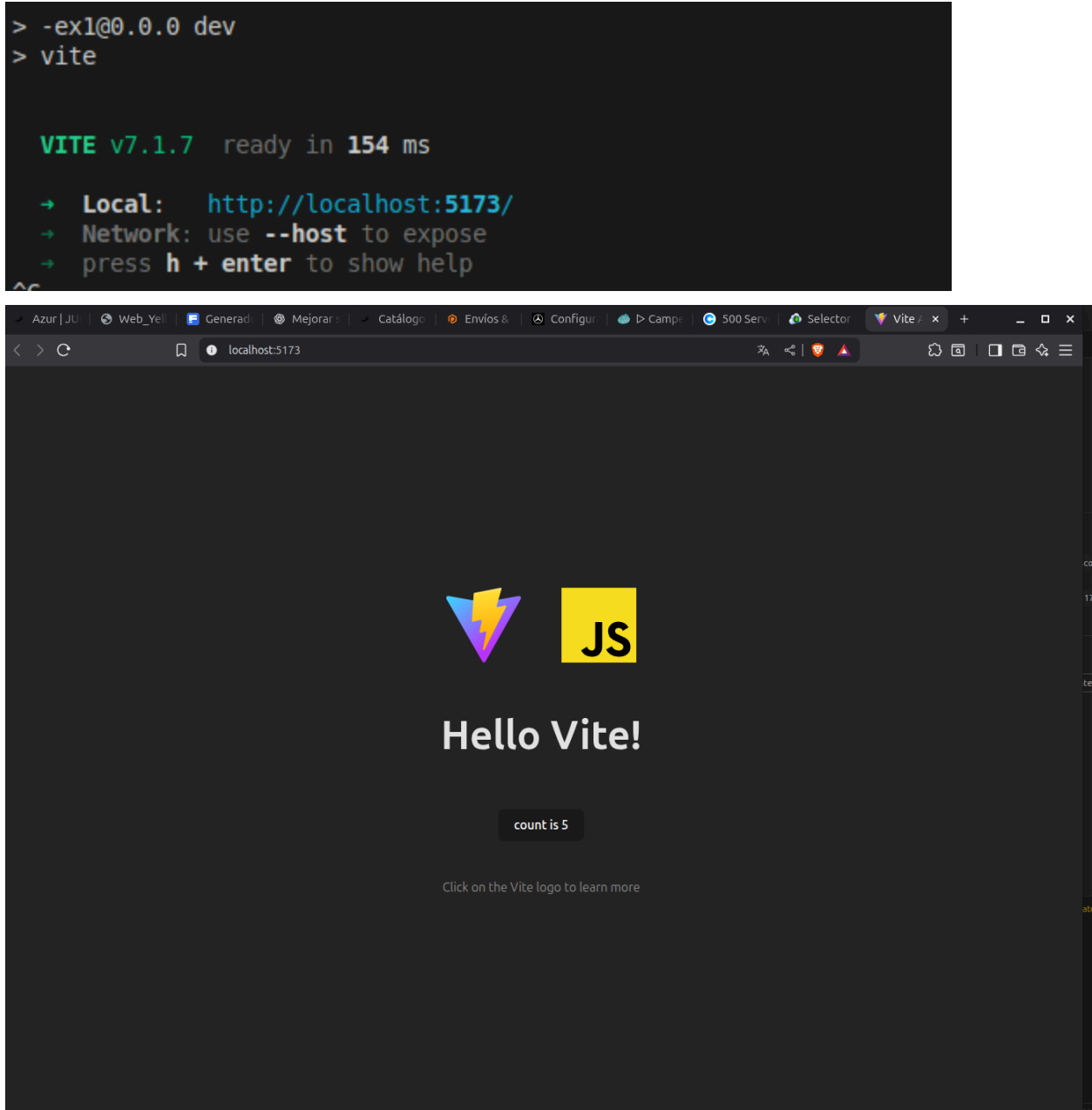
```
roger@RogerUbuntu:~/M6$ cd @latest/
roger@RogerUbuntu:~/M6/@latest$ npm install
npm run dev
```

```
added 13 packages, and audited 14 packages in 4s
```

```
5 packages are looking for funding
run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
> -ex1@0.0.0 dev
```



5. Inicia el servidor de desenvolupament

Video al comprimit

- Implementació d'ESLint

La instal·lació d'ESLint es realitza com a dependència de desenvolupament mitjançant npm. Es configura el fitxer .eslintrc.js amb regles personalitzades que defineixen l'estil de codi. Paral·lelament, s'instal·la l'extensió d'ESLint per a VSCode per habilitar la integració visual.

- Automatització de correccions

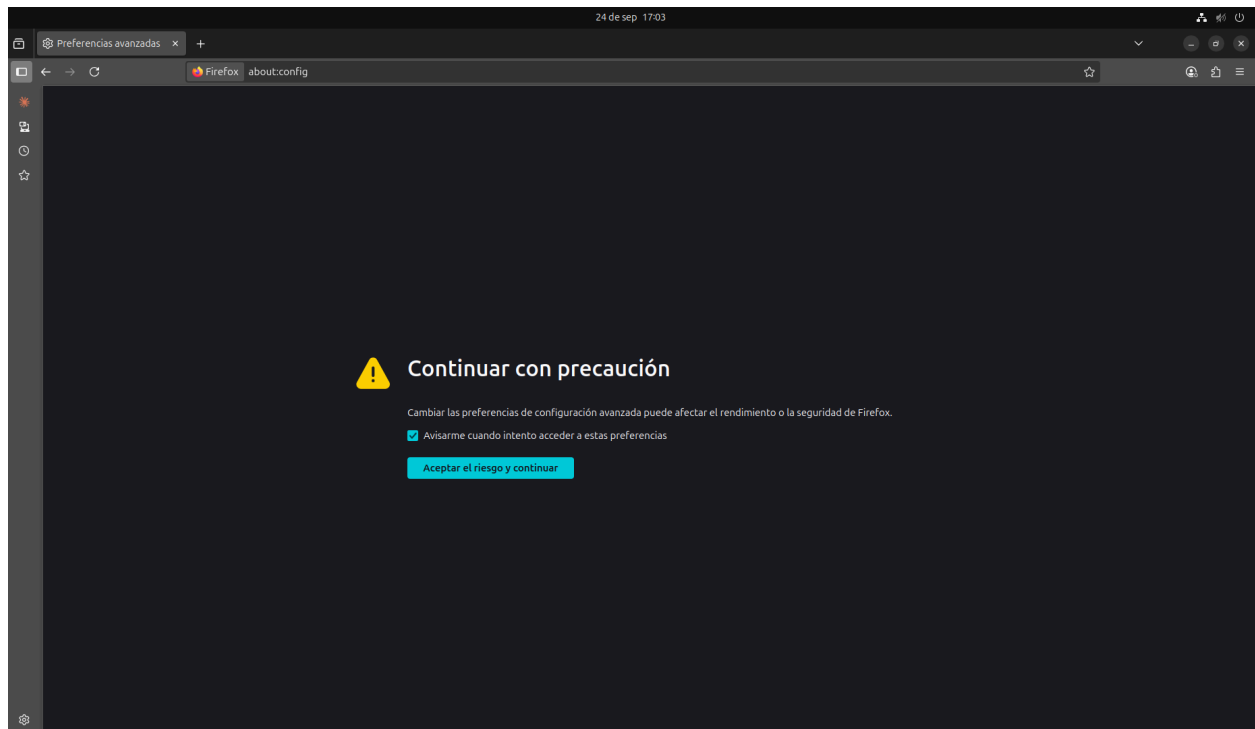
S'estableix la configuració perquè ESLint s'executi automàticament en desar els arxius, aplicant correccions immediates als errors detectats. Això s'aconsegueix mitjançant les opcions d'editor de VSCode, que permeten activar l'acció de correcció en el moment del desat.

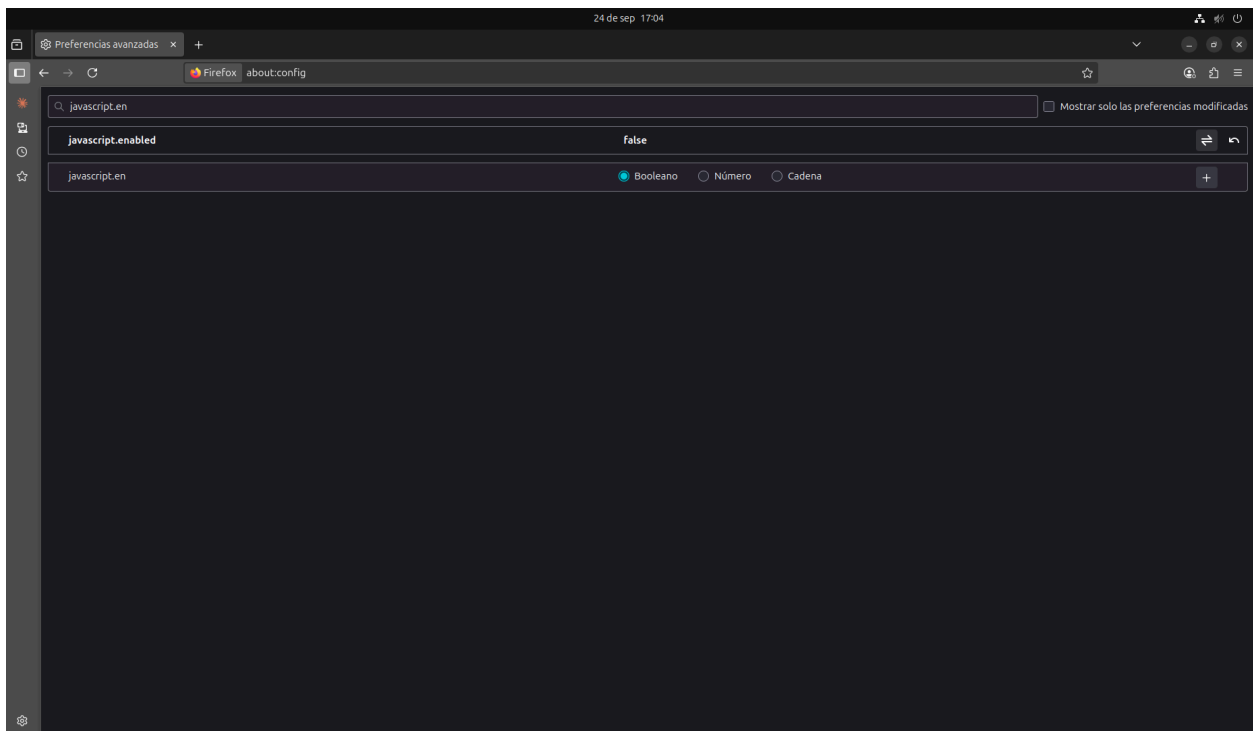
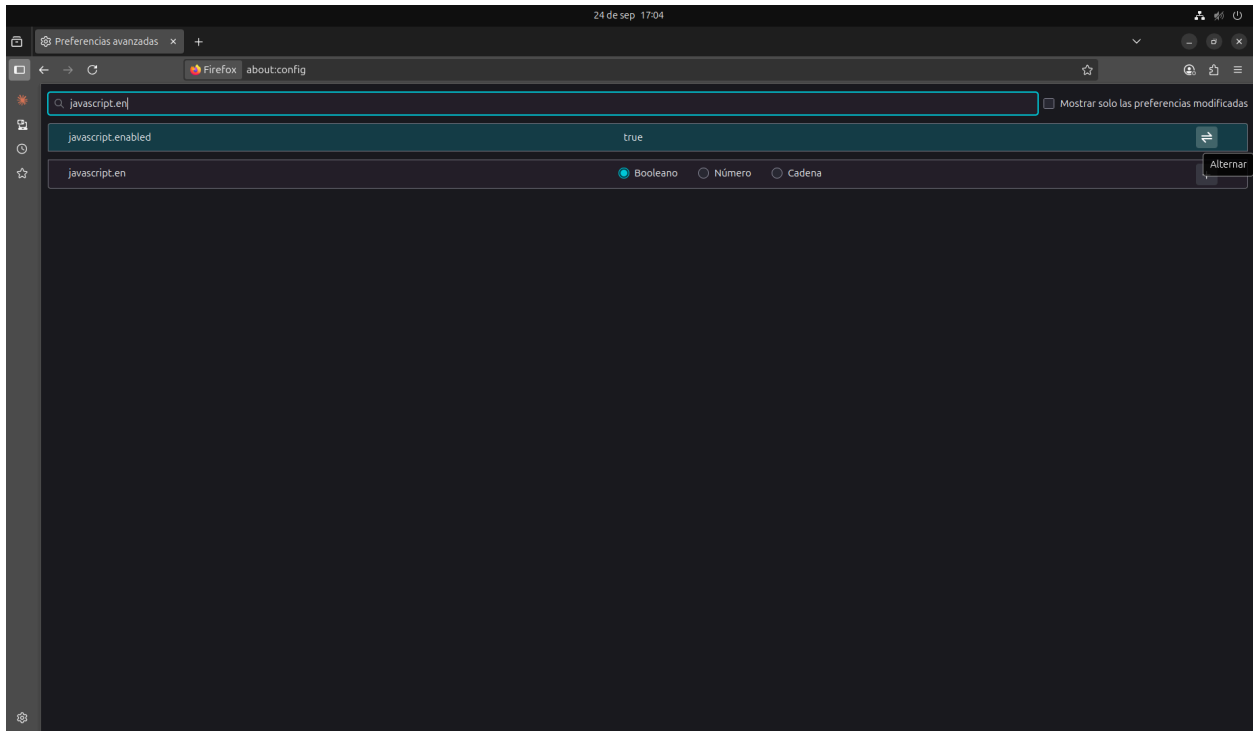
- Avaluació d'extensions

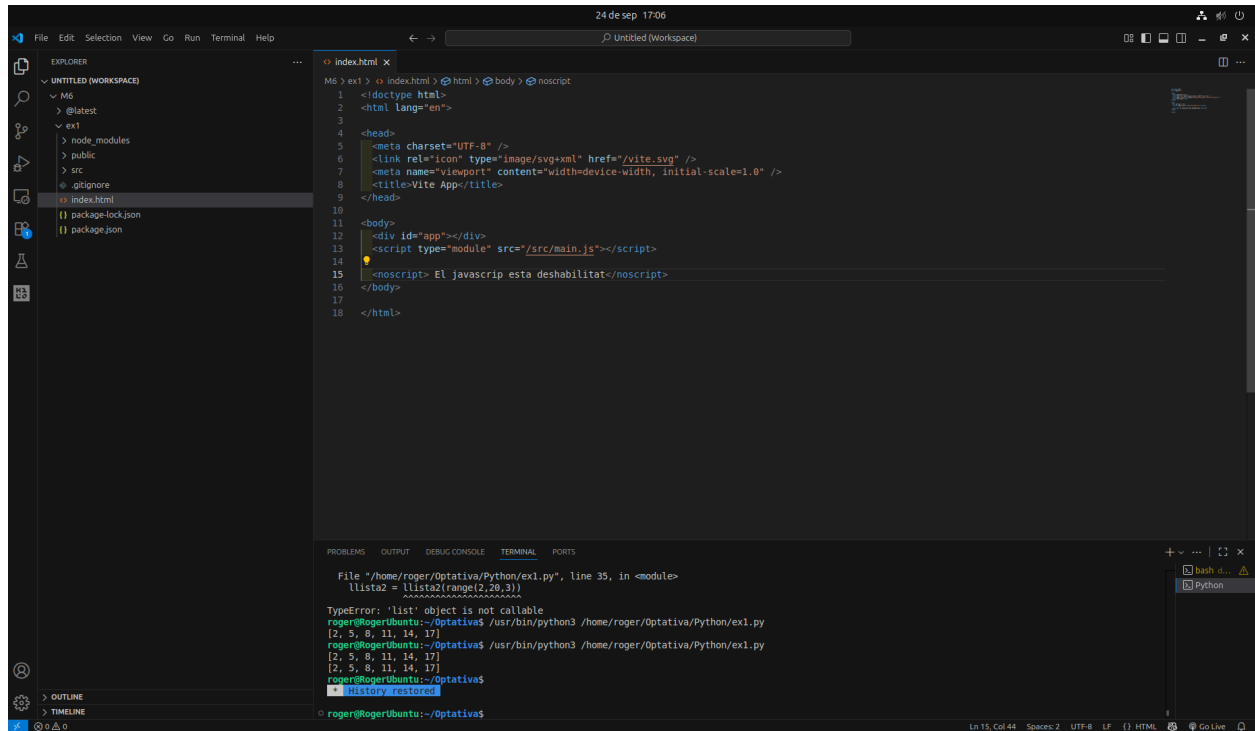
Finalment, s'analitza la conveniència d'incorporar plugins com `@stylistic/eslint-plugin` per millorar l'estil del codi. Aquesta avaluació considera factors com la consistència visual, l'adaptació a les necessitats del projecte i la reducció de decisions subjectives en l'estil de programació.

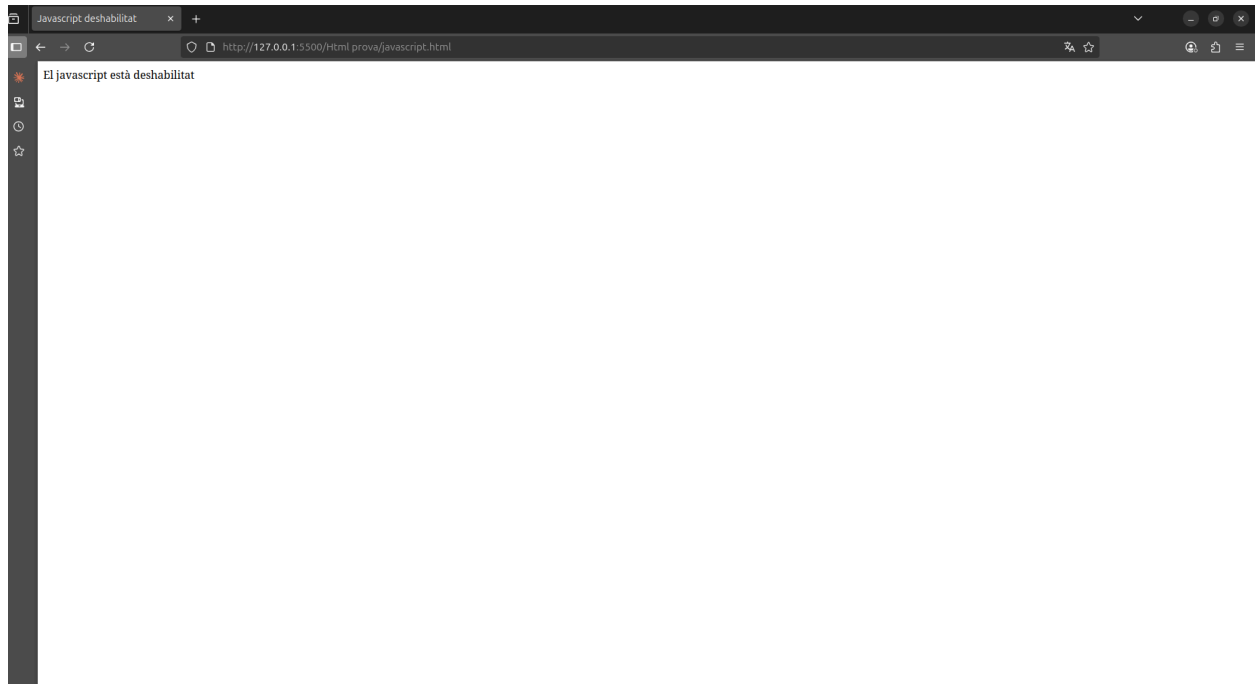
2. Deshabilitar JavaScript

Per deshabilitar JavaScript, s'accedeix a la configuració de seguretat del navegador. En Firefox es localitza a les opcions de privadesa, mentre que en Chrome es troba a la secció de configuracions de contingut. S'incorpora l'etiqueta `<noscript>` al document HTML, que mostra contingut alternatiu quan JavaScript està desactivat. Aquesta etiqueta s'utilitza per informar els usuaris sobre la necessitat d'activar JavaScript per al funcionament òptim del lloc web.



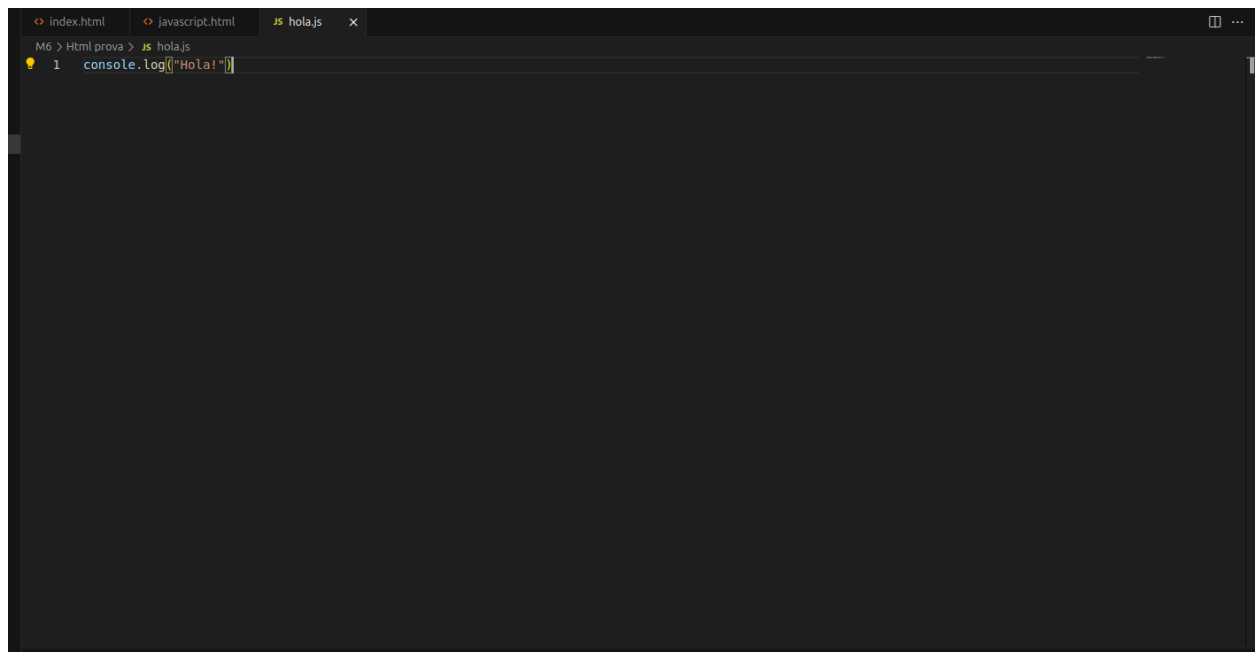






3. Executar JavaScript en el terminal

Es crea un arxiu JavaScript bàsic amb una comanda de sortida per consola. Mitjançant l'interpret de Node.js s'executa el script des de la línia de comandes. Aquest procés permet verificar el correcte funcionament del codi sense necessitat d'un entorn de navegador, utilitzant el terminal del sistema operatiu com a plataforma d'execució.



```
roger@RogerUbuntu: ~/M6/Html prova
roger@RogerUbuntu: /opt... x roger@RogerUbuntu: ~/M... x roger@RogerUbuntu: ~/M... x
roger@RogerUbuntu:~/M6/@latest$ cd /home/roger/M6/
ex1/      Html prova/ @latest/
roger@RogerUbuntu:~/M6/@latest$ cd /home/roger/M6/Html\ prova/
roger@RogerUbuntu:~/M6/Html prova$ node hola.js
Hola!
roger@RogerUbuntu:~/M6/Html prova$
```

4. Debug en el Navegador i en VSCode

Les eines de desenvolupador de Firefox ofereixen un entorn de depuració complet que s'activa amb la tecla F12, on el panell "Debugger" permet establir punts d'aturada al codi font per pausar l'execució i examinar l'estat de l'aplicació. Mitjançant els controls de pas a pas es pot avançar per les línies de codi, saltar funcions o entrar-hi, mentre s'observen els valors actuals de les variables i la pila de crides en temps real. El sistema inclou funcionalitats avançades com punts d'aturada condicionals i expressions de vigilància per a una anàlisi més precisa, tot integrant-se amb la resta d'eines del navegador per oferir una experiència de depuració cohesionada.

VSCode s'configura mitjançant un arxiu launch.json que defineix els paràmetres de depuració. S'estableix la connexió amb el navegador per depurar codi client-side directament des de l'editor. El sistema permet crear punts d'aturada visuals, inspeccionar variables i utilitzar les eines de control d'execució sense alternar entre aplicacions, millorant el flux de treball de desenvolupament.