

Tema 1. El teu servidor de desenvolupament

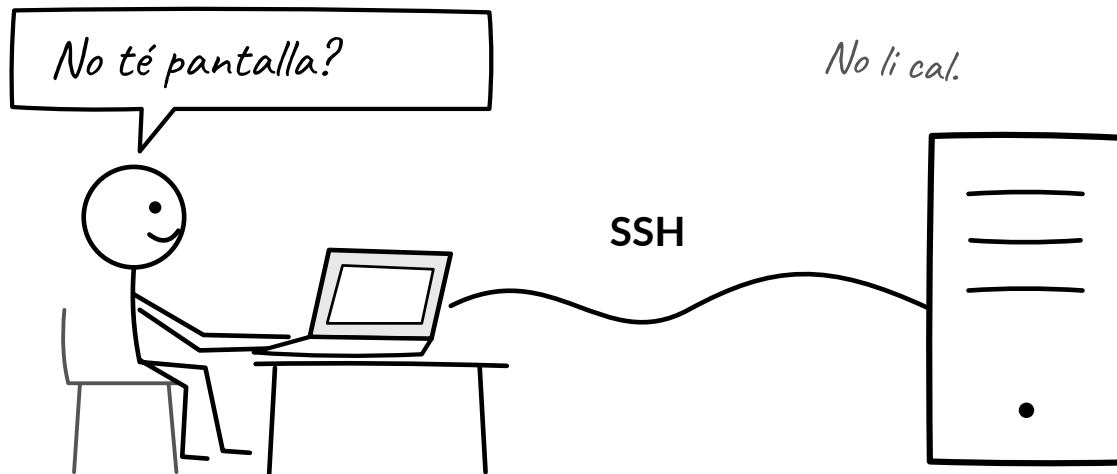


FIG 1.0 Un servidor sense pantalla i tu al teu portàtil. Els uneix SSH.

Aquest tema el fas **amb l'ordinador davant i el llibre obert al costat**. Cada apartat té dues parts: el que fas a la màquina i el que escrius aquí. Si només fas la primera, d'aquí a un mes no recordaràs com ho vas muntar. Quan alguna cosa es trenqui, aquestes pàgines seran el teu manual.

Què aprendràs

- Crear una màquina virtual amb VirtualBox, connectar-la a la xarxa de l'aula i instal·lar-hi Debian 13 sense escriptori.
 - Entrar al servidor per SSH **sense contrasenya**, amb una clau pública.
 - Connectar Visual Studio Code al servidor i treballar-hi amb el seu terminal.
 - Entendre el fitxer `.ssh/config` del teu portàtil i posar-hi un àlies per escriure menys.
 - Instal·lar Node.js 24 amb nvm i executar JavaScript al servidor i al navegador.
 - Instal·lar Docker, que faràs servir a Aplicacions Web.
-

La teva xuleta per a tot el tema

Copia-la o deixa aquesta pàgina marcada. **Fer-la servir no és fer trampa:** els professionals també tenen les seves.

Vull...	Escriu...	On?
Saber la IP de la MV	<code>hostname -I</code>	Servidor
Entrar al servidor per IP	<code>ssh alumne@192.168.1.50</code>	Terminal del portàtil
Entrar amb l'alias	<code>ssh debian13</code>	Terminal del portàtil
Fer les claus	<code>ssh-keygen -t ed25519</code>	Terminal del portàtil
Instal·lar una versió de Node	<code>nvm install 24</code>	Servidor
Saber la versió de Node	<code>node -v</code>	Servidor
Executar un fitxer JavaScript	<code>node fitxer.js</code>	Servidor
Aturar un programa que no acaba	Ctrl + C	On s'executa
Portar el port 3000 al portàtil	Pestanya Ports → Forward a Port → 3000	VS Code
Obrir la web del servidor	<code>http://localhost:3000</code>	Navegador del portàtil
Obrir la consola del navegador	F12, pestanya Console	Navegador del portàtil
Apagar el servidor	<code>sudo poweroff</code>	Servidor

Canvia alumne i 192.168.1.50 pel teu usuari i la IP de la teva MV.

Practica 1.1

La teva bitàcola. Omple aquesta taula a mesura que avancis pel tema. Són les dades que et demanaran quan alguna cosa falli.

Dada	Valor
IP del teu portàtil	
IP de la màquina virtual	
Nom d'usuari al servidor	
Àlies SSH	
Versió de Debian	
Versió de nvm	
Versió de Node.js	
Versió de Docker	
Nom de la primera instantània	

1. Què muntarem

Mira aquest esquema abans de tocar res.

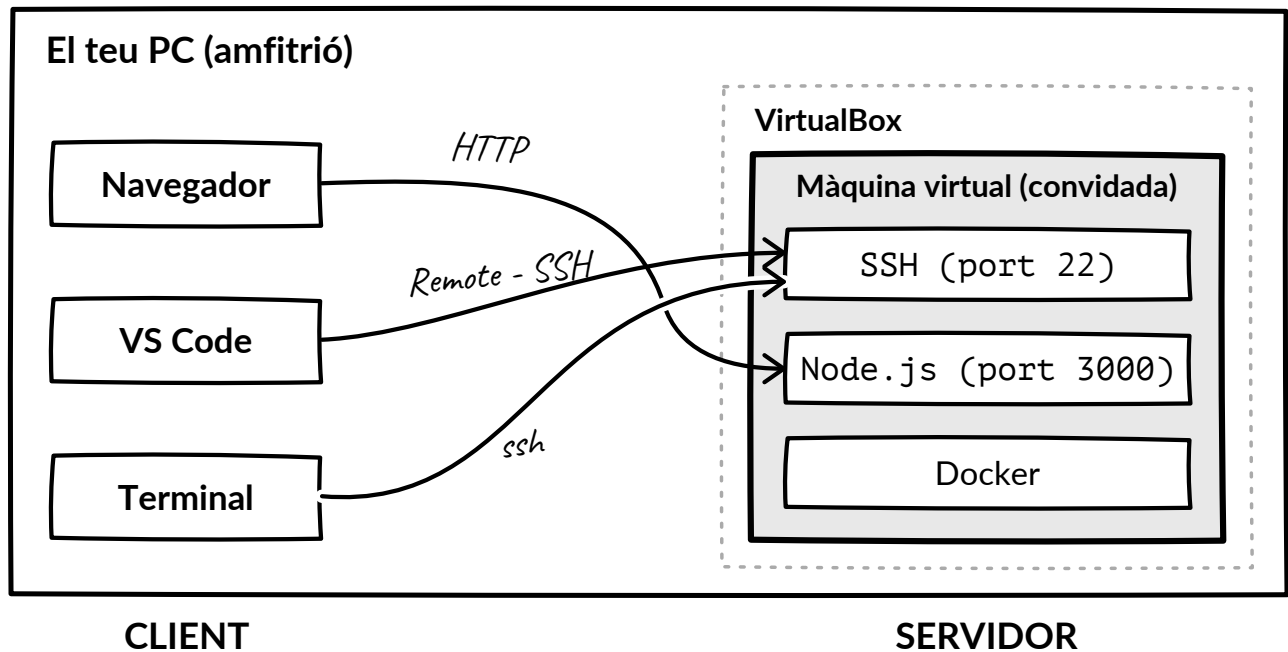


FIG 1.1 El teu portàtil fa de client. La màquina virtual fa de servidor.

Al teu portàtil tens el **navegador**, el **Visual Studio Code** i un **terminal**. Dins del portàtil, VirtualBox fa córrer una segona màquina: la **màquina virtual** (*virtual machine*), MV per abreviar.

El portàtil és l'**amfitrió** (*host*). La màquina virtual és la **convidada** (*guest*).

La MV **no té escriptori**. No li cal, i així gasta menys memòria i menys disc. Els servidors de veritat tampoc en tenen: viuen en un armari o al núvol, i s'hi entra a distància. Tu hi entraràs per **SSH**.

A la MV instal·laràs **Node.js**. Amb Node, JavaScript s'executa **al servidor**. En una pàgina web, JavaScript s'executa **al navegador**, és a dir, al client. Durant el curs faràs servir totes dues coses.

Practica 1.2

Escriu a cada fila si la cosa viu al **portàtil** (client) o a la **MV** (servidor).

Element	Portàtil o MV?
El navegador Firefox o Chrome	Portàtil
El programa node	
El servidor SSH que rep la teva connexió	
La finestra de Visual Studio Code	
El fitxer <code>.ssh/config</code>	
La consola de les eines de desenvolupador (F12)	

2. Crear la màquina virtual

Obre VirtualBox i crea una màquina nova amb aquests valors.

Opció	Valor	Per què
Nom	debian13	És el nom que farem servir a tot el llibre
Tipus i versió	Linux, Debian (64-bit)	
Memòria	4096 MB (3072 si el portàtil té 8 GB)	VS Code i Docker en necessiten
Processadors	2	
Disc	30 GB, reserva dinàmica	Les imatges de Docker ocupen
ISO	debian-13.x.x-amd64-netinst.iso	La mínima: baixa la resta d'internet
Xarxa, adaptador 1	Adaptador pont	La MV tindrà la seva pròpia IP a la xarxa de l'aula

No marquis la instal·lació desatesa (*Unattended Installation*). La faràs a mà.

L'adaptador pont

Ves a *Configuració* → *Xarxa* → *Adaptador 1*. A *Connectat a* tria **Adaptador pont** (*Bridged Adapter*). A *Nom* tria la targeta de xarxa que fa servir el portàtil ara mateix: la **Wi-Fi** si vas sense cable, o l'**Ethernet** si vas amb cable.

Amb l'adaptador pont, la MV es connecta a la xarxa de l'aula **com un ordinador més**. El router li dona una IP de la mateixa xarxa que el portàtil. Així pots entrar-hi directament des del portàtil, i també des de qualsevol altre ordinador de l'aula.

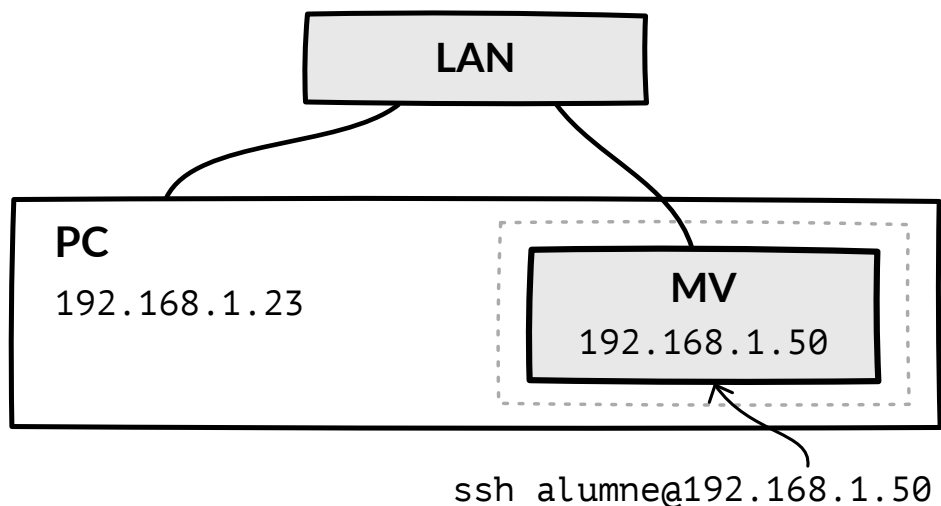


FIG 1.2 Amb l'adaptador pont, el portàtil i la MV són dos equips de la mateixa xarxa.

Fixa't en les dues IP de la figura: 192.168.1.23 i 192.168.1.50. Comencen igual, 192.168.1.: són **de la mateixa xarxa**.

Compte amb la IP 10.0.2.15. És la que dona VirtualBox quan l'adaptador és **NAT**, el que ve per defecte. Amb NAT, la MV surt a internet però el portàtil no hi pot entrar. Si la teva MV té aquesta IP, apaga-la i canvia l'adaptador a pont.

3. Instal·lar Debian 13

Engega la MV. L'instal·lador et fa moltes preguntes. Gairebé totes es contesten amb el valor que proposa. Aquestes quatre, no.

Pantalla	Què poses	Compte
Nom de la màquina	debian13	
Contrasenya de root	La deixes en blanc	Si la poses, no tindràs sudo
Usuari	El teu nom, en minúscules i sense accents	Apunta'l a la bitàcola
Selecció de programari	Només servidor SSH i utilitats estàndard	Desmarca l'entorn d'escriptori i GNOME

Per què la contrasenya de root en blanc? Si la deixes buida, Debian desactiva l'usuari root, instal·la sudo i dona permís d'administrador al teu usuari. Si poses contrasenya, sudo no s'instal·la i hauràs de fer-ho a mà.

Quan acaba, la MV es reinicia i veus una pantalla negra que et demana usuari. Entra-hi i escriu:

```
hostname -I
```

Surt la IP de la MV, per exemple 192.168.1.50. **Apunta-la a la bitàcola.** Comprova també que sudo funciona:

```
sudo whoami
```

Si respon root, tot va bé.

Exemple resolt. De la IP a l'ordre

La MV diu que la seva IP és 192.168.1.50 i l'usuari és alumne. Per entrar-hi, al terminal del portàtil escrius:

```
ssh alumne@192.168.1.50
```

Primer l'usuari, després una @, després la IP. Comprovació: la IP comença igual que la del portàtil, 192.168.1.. Són de la mateixa xarxa. Quadra.

Guiat 1.3

La MV té la IP 192.168.1.73. Hi ha un servidor web de Node al port 3000. L'adreça que escrius al navegador del portàtil és http://192.168.1.73:

Practica 1.4

- a) La IP de la MV és 10.0.20.15 i l'usuari és maria. Ordre per entrar-hi:
- b) A la mateixa MV hi ha un servidor web al port 3000. Adreça al navegador:
- c) El teu portàtil té la IP 192.168.1.23. hostname -I a la MV respon 10.0.2.15. Què passa i què fas?
.....

(Comprovació de la primera: ssh maria@10.0.20.15.)

4. Primera connexió per SSH

Minimitza la finestra de la MV. **A partir d'ara no la tornaràs a fer servir.** Obre un terminal al portàtil (a Windows, PowerShell) i escriu, amb la teva IP i el teu usuari:

```
ssh alumne@192.168.1.50
```

La primera vegada surt una pregunta com aquesta:

```
The authenticity of host '192.168.1.50 (192.168.1.50)' can't be established.
ED25519 key fingerprint is SHA256:Xk3f...
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Escriu yes. El teu portàtil acaba d'apuntar l'**empremta** (*fingerprint*) del servidor. És com el DNI de la màquina. Si un dia canvia, SSH et donarà l'alarma i no et deixarà entrar. Després et demana la contrasenya del teu usuari. **És l'última vegada que l'escriuràs.** Surt del servidor amb `exit`.

Practica 1.5

Explica què vol dir cada tros de l'ordre.

Tros	Què vol dir
<code>ssh</code>	El programa client que fa la connexió
<code>alumne</code>	
<code>@</code>	
<code>192.168.1.50</code>	

Quan no connecta

Aquests són els quatre errors que veuràs més. Llegeix el missatge sencer: gairebé sempre diu què passa.

Missatge	Què vol dir	Què fas
<code>Connection refused</code>	La MV hi és, però no hi ha res escoltant al port de SSH	Mira que la MV estigui engegada i que hagi instal·lat el servidor SSH
<code>Connection timed out</code>	La petició no arriba a la MV	Revisa la IP amb <code>hostname -I</code> . Si és <code>10.0.2.15</code> , la MV és en NAT
<code>Permission denied</code>	Arribes, però no et deixa entrar	Revisa l'usuari i la contrasenya
<code>REMOTE HOST IDENTIFICATION HAS CHANGED</code>	L'empremta del servidor no és la que el portàtil tenia apuntada	Si has reinstal·lat la MV, esborra l'antiga amb <code>ssh-keygen -R 192.168.1.50</code>

5. Entrar sense contrasenya: la clau pública

Escriure la contrasenya cada vegada cansa, i VS Code te la demanaria molts cops. La solució són les **claus SSH**. Fas una parella de claus **al portàtil**:

- La **clau privada** (*private key*) es queda al portàtil. No la dones mai a ningú.
- La **clau pública** (*public key*) la copies al servidor. La pot veure tothom.

Quan et connectes, el servidor comprova que tens la privada que fa parella amb la pública que ell guarda. Si és així, entres sense contrasenya. El servidor guarda les claus públiques que accepta en un fitxer que es diu `~/.ssh/authorized_keys`.

Primer, fes les claus. Al terminal del portàtil, no dins del servidor:

```
ssh-keygen -t ed25519
```

Prem Retorn a totes les preguntes. Es creen dos fitxers a la carpeta `.ssh` del teu usuari: `id_ed25519` (la privada) i `id_ed25519.pub` (la pública).

Després, copia la pública al servidor. Depèn del sistema del portàtil.

A **Linux** o **macOS**:

```
ssh-copy-id alumne@192.168.1.50
```

A **Windows**, a PowerShell. Són dues línies, però és una sola ordre: la primera acaba en `|`.

```
type $env:USERPROFILE\.ssh\id_ed25519.pub |  
ssh alumne@192.168.1.50 "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

Totes dues et demanen la contrasenya per última vegada. **Ara prova-ho:**

```
ssh alumne@192.168.1.50
```

Si entres sense que et demani la contrasenya, la clau funciona.

Practica 1.6

Després de fer les claus i copiar la pública, marca amb una X on és cada fitxer.

Fitxer	Al portàtil	A la MV
<code>.ssh/id_ed25519</code> (clau privada)		
<code>.ssh/id_ed25519.pub</code> (clau pública)		
<code>.ssh/authorized_keys</code>		

Practica 1.7

Un company diu que, per anar més ràpid, copiarà la seva clau **privada** a la MV. Per què és una mala idea?

6. Visual Studio Code dins del servidor

Ara connectaràs VS Code al servidor. Com que la clau ja funciona, **no et demanarà contrasenya**.

1. Obre Visual Studio Code **al portàtil**.
2. Instal·la l'extensió **Remote - SSH**, de Microsoft.
3. Prem F1 i escriu Remote-SSH: Add New SSH Host.
4. Escriu l'ordre sencera, com al terminal: `ssh alumne@192.168.1.50`.
5. VS Code pregunta en quin fitxer ho ha de guardar. Tria el primer: el `.ssh/config` del teu usuari.
6. Prem F1 un altre cop, escriu Remote-SSH: Connect to Host i tria 192.168.1.50.
7. Si pregunta el sistema del servidor, tria **Linux**. La primera vegada, VS Code instal·la un petit servidor dins la MV. Triga un minut.
8. A baix a l'esquerra ha de posar **SSH: 192.168.1.50**.

Obre el terminal amb el menú *Terminal* → *New Terminal*. **Aquest terminal ja és al servidor**. Crea la carpeta de treball:

```
mkdir -p ~/projectes/hola
```

Obre-la amb *File* → *Open Folder*. Els fitxers que hi crees es guarden a la MV, no al portàtil.

Practica 1.8

Al terminal de VS Code escriu `hostname`. Després obre un terminal normal del portàtil i escriu el mateix. Apunta les dues respostes i explica la diferència.

7. El fitxer .ssh/config i l'àlies

Al pas 5, VS Code ha escrit una cosa al teu portàtil. Mira-la. Prem F1, escriu Remote-SSH: Open SSH Configuration File i tria el mateix fitxer. Veuràs un bloc com aquest:

```
Host 192.168.1.50
  HostName 192.168.1.50
  User alumne
```

.ssh/config

El fitxer és a la carpeta del teu usuari: C:\Users\el_teu_nom\.ssh\config a Windows, ~/.ssh/config a Linux i macOS. Cada línia vol dir una cosa:

Línia	Què vol dir
Host	El nom curt que fas servir per connectar-te: l' àlies
HostName	L'adreça de veritat, la IP de la MV
User	L'usuari amb què entres

Ara mateix l'àlies és la mateixa IP. Això no estalvia res. **Canvia la primera línia** i desa el fitxer:

```
Host debian13
  HostName 192.168.1.50
  User alumne
```

.ssh/config

Obre un terminal normal del portàtil i prova-ho:

```
ssh debian13
```

Entres sense IP, sense usuari i sense contrasenya. SSH busca debian13 al fitxer, i hi troba la IP i l'usuari. A VS Code, a *Remote-SSH: Connect to Host*, ara també surt debian13.

Practica 1.9

Tens una segona MV que es diu proves, amb la IP 192.168.1.51 i el mateix usuari. Escriu el bloc que afegiries a .ssh/config per entrar-hi amb ssh proves.

1
2
3

Practica 1.10

Demà, el router dona a la teva MV la IP 192.168.1.77, i ssh `debian13` ja no connecta. Quina línia del fitxer canvies, i com queda?

8. Instal·lar Node.js 24 amb nvm

Tot això es fa al **terminal de VS Code**, que és al servidor.

Debian 13 porta Node.js 20 als seus repositoris. **No l'instal·lis**. Aquesta versió va deixar de rebre actualitzacions l'abril de 2026. Instal·laràs la **24**, que és una versió **LTS** (*Long Term Support*): rep correccions fins a l'abril de 2028.

No l'instal·laràs amb `apt`, sinó amb **nvm** (*Node Version Manager*). És un programa que instal·la Node **dins de la carpeta del teu usuari** i et deixa tenir diverses versions i canviar d'una a l'altra. Així:

- **No cal** sudo per instal·lar Node ni els paquets de npm.
- Si un projecte necessita una altra versió, n'instal·les una altra sense tocar la primera.

Primer, instal·la nvm. Només necessita `curl`:

```
sudo apt update
sudo apt install -y curl
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.7/install.sh | bash
```

L'última línia **no porta** sudo. L'script copia nvm a la carpeta `~/ .nvm` i afegeix unes línies al final del teu `~/ .bashrc`, el fitxer que el terminal llegeix cada vegada que s'obre.

Com que el terminal que tens obert ja havia llegit el `~/ .bashrc` abans, encara no coneix nvm. Fes-li llegir un altre cop:

```
source ~/.bashrc
nvm --version
```

Si respon un número, com `0.40.7`, nvm està a punt. Apunta'l a la bitàcola.

Després, instal·la Node 24:

```
nvm install 24
```

Escriu 24 i no --lts. --lts **instal·la la LTS més nova del dia que ho escrius**, i a l'octubre de 2026 passa a ser la 26. Amb 24, tota la classe té la mateixa versió tot el curs.

La primera versió que instal·les queda com a **versió per defecte**: és la que tindràs cada vegada que obris un terminal. Comprova-ho i apunta la versió a la bitàcola:

```
node -v
npm -v
```

Compte amb el `curl ... | bash`. Aquesta línia baixa un script d'internet i l'executa sense ensenyar-te'l. Aquí ho fem perquè l'adreça és la del projecte nvm, i la versió (v0.40.7) va escrita: sempre baixes el mateix script. No ho facis mai amb una adreça que no coneixes.

Quan node no es troba

nvm posa Node dins de `~/ .nvm`, i ho diu al terminal a través del `~/ .bashrc`. D'aquí surten els dos errors que veuràs més:

Missatge	Què passa	Què fas
<code>nvm: command not found</code>	El terminal es va obrir abans d'instal·lar nvm	<code>source ~/.bashrc</code> , o tanca el terminal de VS Code i obre'n un de nou
<code>sudo: node: command not found</code>	<code>sudo</code> executa com a root, i root no té el teu <code>~/ .nvm</code>	Treu el <code>sudo</code> . Amb nvm, Node no el necessita

Practica 1.11

Escriu què fa cada ordre. Totes s'escriuen al terminal del servidor.

Ordre	Què fa
<code>nvm install 24</code>	Baixa i instal·la l'última versió de Node 24
<code>nvm ls</code>	
<code>node -v</code>	
<code>source ~/.bashrc</code>	
<code>sudo node -v</code>	

QR

QR 1.1

Ordres actualitzades d'aquest tema, per si canvien les versions

9. JavaScript en dos llocs

Obre la carpeta de treball

Des que et connectes amb l'àlies, VS Code obre una finestra nova **sense cap carpeta**. Abans de crear fitxers, torna-la a obrir:

1. Prem F1, escriu Remote-SSH: Connect to Host i tria **debian13**.
2. Menú *File* → *Open Folder*. Escriu `/home/alumne/projectes/hola` i prem **OK**.
3. Comprova-ho: a l'esquerra, l'explorador mostra la carpeta **HOLA**, i a baix a l'esquerra posa **SSH: debian13**.

Obre el terminal amb *Terminal* → *New Terminal*. Com que la carpeta és oberta, el terminal ja hi comença: no cal fer `cd`. Tot el que facis en aquest apartat anirà a aquesta carpeta.

Al servidor: un script al terminal

Crea aquest fitxer a `~/projectes/hola`, amb el botó *New File* de l'explorador:

```
1 console.log('Hola des del servidor');
2 console.log('2 + 3 =', 2 + 3);
```

hola.js

Executa'l al terminal de VS Code:

```
node hola.js
```

Surt això:

```
Hola des del servidor
2 + 3 = 5
```

No hi ha navegador enlloc. Això és JavaScript al servidor.

Al navegador: una pàgina amb un fitxer .js

Ara el servidor enviarà una pàgina al navegador. El navegador fa una **petició** (*request*) i el servidor li torna una **resposta** (*response*). Faràs tres fitxers a ~/projectes/hola: la pàgina, el JavaScript que s'executa al navegador i el servidor.

La pàgina. Fixa't en la línia del <script>: carrega el fitxer app.js.

```
index.html
1 <!DOCTYPE html>
2 <html lang="ca">
3 <head>
4   <meta charset="utf-8">
5   <title>Hola</title>
6 </head>
7 <body>
8   <h1>Hola des del servidor</h1>
9   <p id="hora"></p>
10  <script src="app.js"></script>
11 </body>
12 </html>
```

El JavaScript del navegador:

```
app.js
1 console.log('Això surt a la consola del navegador');
2 document.getElementById('hora').textContent = new Date().toLocaleTimeString();
```

El servidor. **No cal que ho entenguis tot.** Fixa't només que, quan li demanen /, envia index.html, i quan li demanen /app.js, envia app.js.

servidor.js

```

1  const http = require('node:http');
2  const fs = require('node:fs');
3
4  const fitxers = {
5    '/': ['index.html', 'text/html; charset=utf-8'],
6    '/app.js': ['app.js', 'text/javascript; charset=utf-8'],
7  };
8
9  const servidor = http.createServer((peticio, resposta) => {
10   console.log('Petició rebuda:', peticio.url);
11   const fitxer = fitxers[peticio.url];
12   if (!fitxer) {
13     resposta.writeHead(404);
14     resposta.end('No trobat');
15     return;
16   }
17   resposta.writeHead(200, { 'Content-Type': fitxer[1] });
18   resposta.end(fs.readFileSync(__dirname + '/' + fitxer[0]));
19 });
20
21 servidor.listen(3000, () => {
22   console.log('Servidor escoltant al port 3000');
23 });

```

Engega'l al terminal de VS Code:

```
node servidor.js
```

El terminal es queda esperant. És normal: el servidor escolta. Però escolta al port 3000 **de la MV**, i el navegador és al **portàtil**. VS Code, que ja està connectat a la MV per SSH, pot fer de pont entre tots dos:

1. Al panell de baix, on tens el terminal, obre la pestanya **Ports**.
2. Clica **Forward a Port** i escriu 3000. Surt una fila amb el port 3000 i l'adreça localhost:3000.
3. Al navegador del portàtil, obre http://localhost:3000. Obre també la consola amb F12.

Això es diu **reenviament de ports** (*port forwarding*): el que el navegador demana al port 3000 del portàtil, VS Code ho porta per dins de la connexió SSH fins al port 3000 de la MV. De vegades VS Code detecta el port tot sol i l'afegeix a la llista. Si ja hi és, salta el pas 2.

I la IP de la MV? Amb l'adaptador pont, http://192.168.1.50:3000 també funciona si la xarxa de l'aula ho deixa. El reenviament funciona sempre que funcioni l'SSH, per això el fem servir.

Practica 1.12

Mira els tres fitxers i omple la taula **abans** de provar-ho. Després comprova-ho.

Missatge	Terminal de VS Code o consola del navegador?
<code>Servidor escoltant al port 3000</code>	Terminal de VS Code
Petició rebuda: /	
Això surt a la consola del navegador	

Practica 1.13

Mira el terminal de VS Code. Quines adreces apareixen després de Petició rebuda: ? N'hi ha alguna que tu no has escrit al navegador?

Ho has vist? Tu només has escrit l'adreça de la pàgina, però el navegador ha fet més peticions pel seu compte. Demana `/app.js` perquè el `<script>` de la pàgina el carrega. I demana `/favicon.ico` per buscar la icona de la pestanya. Com que el servidor no la té, respon 404, i a la consola del navegador surt un error en vermell. **No passa res:** és la icona.

Practica 1.14

L'hora que surt a la pàgina, la calcula el servidor o el navegador? Digues en quin fitxer és la línia que ho fa i com ho has sabut.

Per aturar el servidor, torna al terminal de VS Code i prem **Ctrl + C**.

10. Docker, per a Aplicacions Web

A Aplicacions Web desplegaràs WordPress i altres aplicacions dins de **contenidors** (*containers*). Un contenidor és una aplicació amb tot el que necessita, aïllada de la resta del sistema. Qui fa córrer els contenidors és **Docker**. L'instal·les ara, al terminal de VS Code, i així ja el tens.

No facis servir el paquet `docker.io` de Debian. Instal·la la versió oficial:

```
sudo apt install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
sudo tee /etc/apt/sources.list.d/docker.sources <<FI
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
FI
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
    docker-buildx-plugin docker-compose-plugin
```

Perquè el teu usuari pugui fer servir Docker sense sudo:

```
sudo usermod -aG docker $USER
```

Tanca la connexió i torna a entrar. El canvi de grup no s'aplica fins que tornes a entrar. A VS Code, prem F1 i tria Remote-SSH: Kill VS Code Server on Host, i torna't a connectar a `debian13`. Després comprova-ho:

```
docker run hello-world
```

Si surt `Hello from Docker!`, funciona. Apunta la versió de `docker --version` a la bitàcola.

11. Guarda el que has fet

Ara tens un servidor que funciona. Si demà el trenques, no vols repetir tot el tema.

Apaga la MV amb `sudo poweroff`. A VirtualBox, selecciona-la i fes una **instantània** (*snapshot*). Posa-li de nom `entorn-net` i apunta-ho a la bitàcola. Si un dia la MV queda inservible, tornes a aquest punt en deu segons.

El que has de saber sí o sí

- El portàtil és l'**amfitrió** i fa de client. La MV és la **convidada** i fa de servidor.
- Amb l'**adaptador pont**, la MV té la seva IP a la xarxa de l'aula i hi entres directament: `ssh alumne@192.168.1.50`.

- Si la MV té la IP 10.0.2.15, és en NAT i no hi podràs entrar.
- Si deixes en blanc la contrasenya de root, Debian et configura sudo.
- La **clau privada** no surt mai del portàtil. La **pública** va al servidor, a `authorized_keys`.
- VS Code guarda els servidors al fitxer `.ssh/config` del portàtil.
- A `.ssh/config`, Host és l'**àlies**, HostName és la IP i User és l'usuari.
- Amb l'àlies entres amb `ssh debian13`. Si la IP canvia, només canvies HostName.
- **Node s'instal·la amb nvm**, sense sudo, a la carpeta del teu usuari. Amb `nvm install 24` tota la classe té la mateixa versió.
- `node fitxer.js` executa JavaScript al **servidor**. Un `.js` carregat per una pàgina s'executa al **navegador**.
- Per obrir al portàtil un servidor de la MV, **reenvia el port** a la pestanya Ports de VS Code i obre `http://localhost:3000`.
- Abans de tocar res important, fes una **instantània**.

Paraules noves

Paraula	Què és	Les meves notes
Amfitrió (<i>host</i>)	L'ordinador real, on corre VirtualBox.	
Convidat (<i>guest</i>)	La màquina virtual.	
Adaptador pont (<i>bridged adapter</i>)	Mode de xarxa on la MV és un equip més de la xarxa, amb la seva IP.	
NAT	Mode de xarxa on la MV surt a internet a través de l'amfitrió, però no s'hi pot entrar.	
Adreça IP	Número que identifica un equip dins d'una xarxa.	
SSH	Protocol per treballar en un altre ordinador per terminal, de forma xifrada.	
Empremta (<i>fingerprint</i>)	Identificador únic de la clau d'un servidor.	
Clau privada (<i>private key</i>)	Meitat secreta d'una parella de claus. No es comparteix.	

Paraula	Què és	Les meves notes
Clau pública (<i>public key</i>)	Meitat que es copia als servidors on vols entrar.	
Àlies	Nom curt que poses a <code>.ssh/config</code> per connectar-te sense escriure la IP.	
Node.js	Programa que executa JavaScript fora del navegador.	
nvm (<i>Node Version Manager</i>)	Programa que instal·la i canvia versions de Node dins la carpeta de l'usuari.	
LTS (<i>Long Term Support</i>)	Versió que rep correccions durant anys.	
Petició (<i>request</i>)	Missatge que el navegador envia al servidor per demanar una cosa.	
Reenviament de ports (<i>port forwarding</i>)	Portar un port de la MV al portàtil a través de la connexió SSH.	
Contenedor (<i>container</i>)	Aplicació aïllada amb tot el que necessita per funcionar.	
Instantània (<i>snapshot</i>)	Còpia de l'estat d'una MV per tornar-hi més tard.	

Repàs del tema

A. Vertader o fals

Frase	V o F	Per què
Amb l'adaptador pont, la MV té una IP de la mateixa xarxa que el portàtil		
La clau privada s'ha de copiar al servidor		
El fitxer <code>.ssh/config</code> és a la MV		
A <code>.ssh/config</code> , Host és la IP del servidor		
El <code>console.log</code> de <code>servidor.js</code> surt a la consola del navegador		
Si poses contrasenya a root, <code>sudo</code> queda configurat igualment		
Node instal·lat amb <code>nvm</code> s'ha d'executar amb <code>sudo</code>		

B. Preguntes curtes

1. Per què la MV no porta escriptori?

2. Quina diferència hi ha entre executar `node hola.js` i obrir una pàgina que carrega `app.js`?

3. Què guanyes posant Host `debian13` a `.ssh/config`?

C. Problemes

Problema 1. Escris `ssh debian13` i surt `Connection timed out`. Ahir funcionava. Escriu, per ordre, tres coses que comprovaries.

Problema 2. Has fet `ssh-keygen` i t'has connectat, però cada vegada et continua demanant la contrasenya. Què t'has deixat, i com ho arregles?

Problema 3. Engegues `node servidor.js` i al terminal surt `Servidor escoltant al port 3000`. Al navegador escrius `http://localhost:3000` i diu que no pot connectar. A la pestanya Ports de VS Code no hi ha cap fila. Per què passa, i què has de fer?
