



## **Guia d'instal·lació**

# **Eina de suport al programador per desplegar una aplicació web amb Docker**

Joan R. Hernández Chaves

**Màster d'Aplicacions Multimèdia**

**Universitat Oberta de Catalunya**

**Curs 2019 - 2020**

## INDEX

INTRODUCCIÓ.....	2
MÀQUINA VIRTUAL .....	2
MÀQUINA VIRTUAL: Ubuntu server.....	4
DOCKER: Instal·lació .....	12
DOCKER: Comandes .....	15
PLANTILLA WEB: Captures de pantalla .....	16
BIBLIOGRAFIA.....	18

## INTRODUCCIÓ

Per realitzar aquest projecte s'han fet servir eines locals per fer proves i eines externes que després facilitaràn el funcionament del projecte.

Per un costat, les eines locals són:

- **Oracle VM VirtualBox:** Software de virtualització de sistemes operatius gratuïta desenvolupat per Oracle.
- **Ubuntu server 18.04 LTS (64 bits):** Sistema operatiu Linux amb suport fins l'abril del 2023, orientat principalment a servidors i sobre tot per serveis web.

Per altre costat, les eines externes són:

- **GitHub:** Lloc web on allotjar projectes desenvolupats fent servir el sistema de control de versions Git.
- **Docker:** Eina de desplegament d'aplicacions dintre de contenidors de software independents entre sí.
- **Laradock:** Entorn de desenvolupament basat en Docker que suporta una gran varietat de serveis comuns i preconfigurats per proporcionar un entorn de desenvolupament complet de PHP mitjançant contenidors llicenciada sota MIT.

Un cop definies les eines que s'utilitzen en aquest projecte, a continuació es podrà veure una guia d'instal·lació d'ajuda a l'usuari que vulgui fer servir l'eina de suport al programador. Cal destacar que s'adjunta un script creat en bash per executar dins del servidor amb la intenció de fer les instal·lacions de forma desatesa anomenat "**jhernandezch\_script.sh**".

## MÀQUINA VIRTUAL

Per fer la màquina virtual per el qual es farà la instal·lació i proves de l'eina, primer de tot s'ha d'instal·lar **VirtualBox** (figura 1 i figura 2) que es pot descarregar des de l'enllaç oficial: <https://www.virtualbox.org>

Per instal·lar, només cal seguir l'assistent i fer "*següent*" a totes les pantalles fins finalitzar. No fa falta considerar cap opció extra i cal deixar per defecte tots els paràmetres fins al final.

Algunes captures de l'assistent:



Figura 1: Instal·lació de VirtualBox.

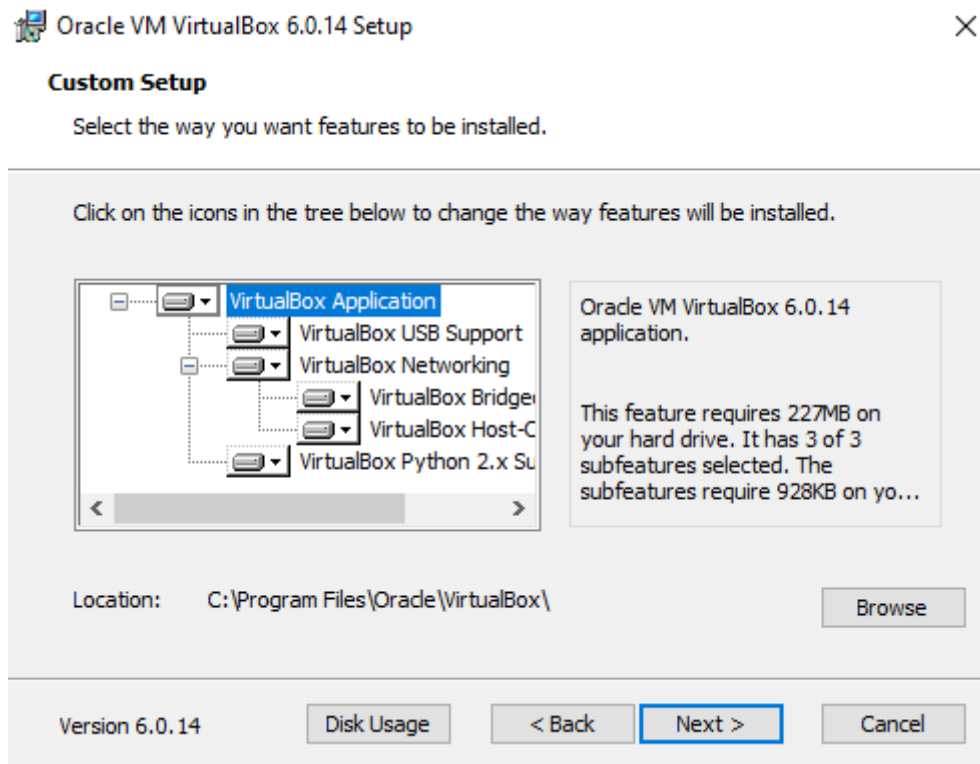
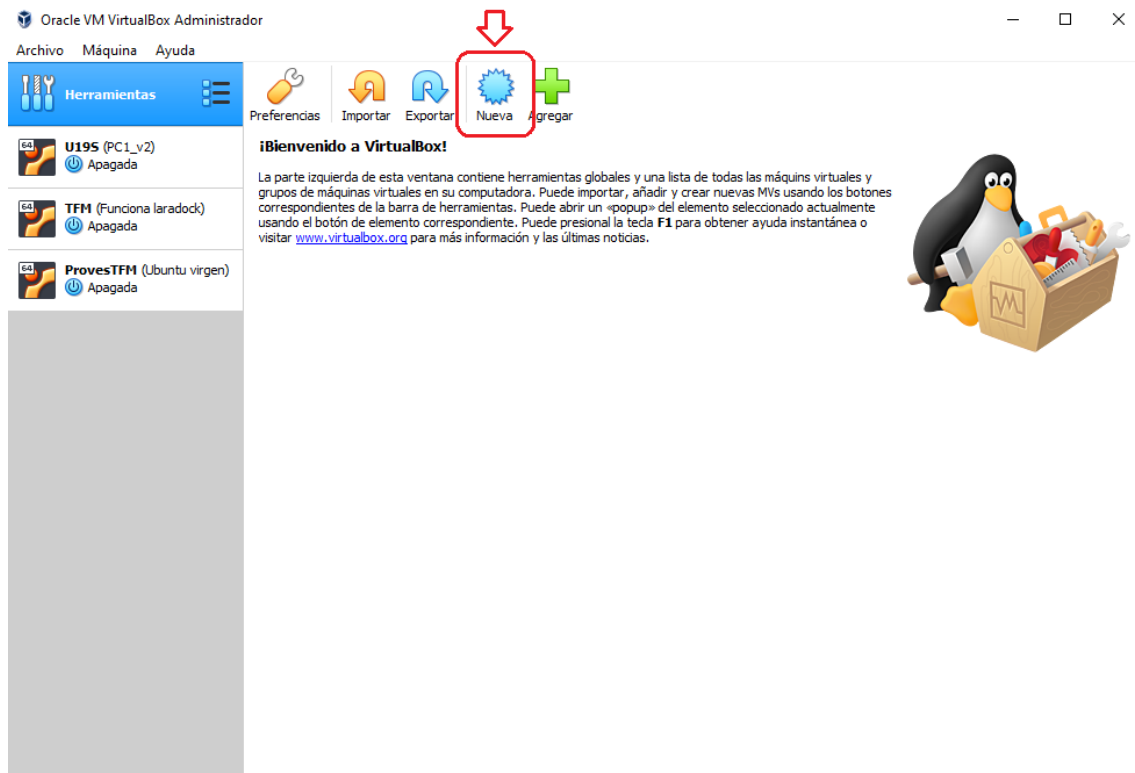


Figura 2: Instal·lació de VirtualBox.

## MÀQUINA VIRTUAL: Ubuntu server

A continuació, un cop instal·lat virtualbox, el següent pas és crear una nova màquina virtual amb sistema operatiu Linux Ubuntu de 64 bits, que es pot descarregar des de l'enllaç oficial: <https://ubuntu.com/download/server>

Obrint el programa, el que s'ha de fer primer és crear una nova màquina virtual clicant sobre el botó "Nueva":



**Figura 3:** *VirtualBox.*

A continuació, s'ha de posar nom a la màquina virtual, seleccionar el sistema operatiu Linux Ubuntu de 64 bits, un total de 2GB de memòria RAM per assegurar el bon rendiment i finalment seleccionar la opció de crear un nou disc dur virtual (Figura 4).

Configurada la primera pantalla, a continuació consisteix en agafar la ubicació on es crearà la màquina virtual i configurar el disc dur (Figura 5). Ara, el que es farà serà agafar un total de 30 GB per aquest projecte i fer proves, amb un tipus d'arxiu VDI i emmagatzematge reservada dinàmicament (que com en el cas contrari de mida fixa, aquest no reserva directament els 30 GB al disc dur físic i deixar que ocupi memòria dinàmicament):

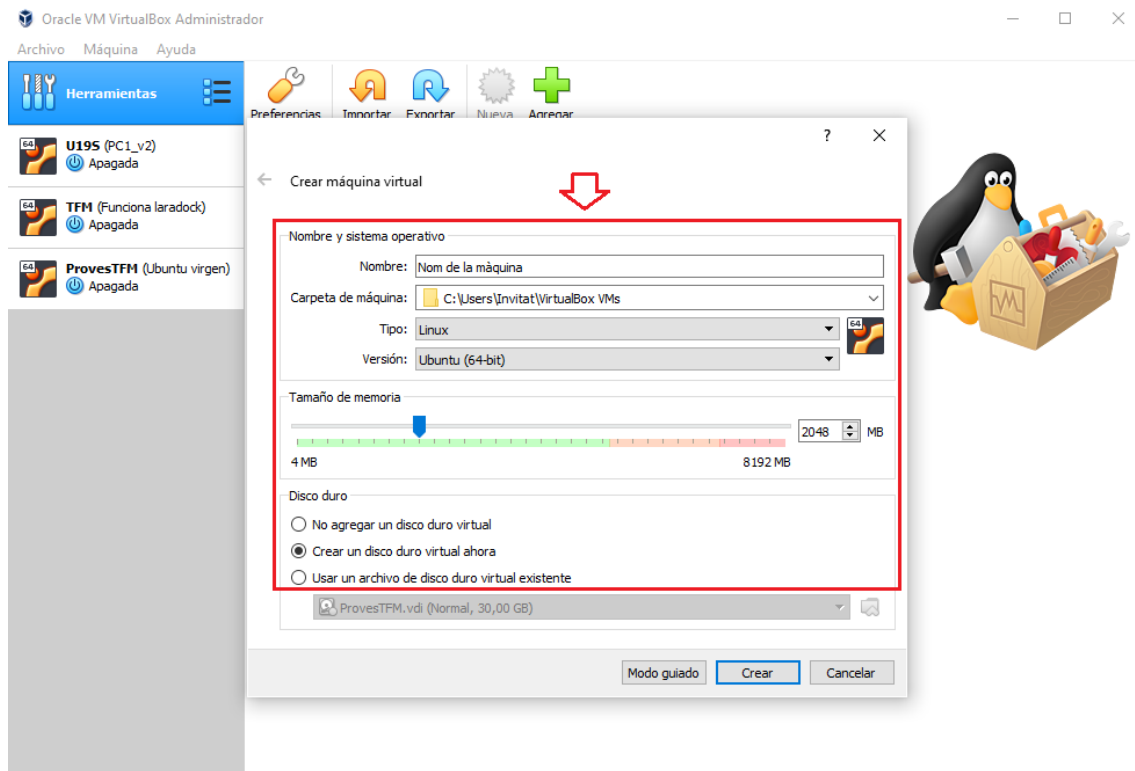


Figura 4: Creació de la màquina virtual.

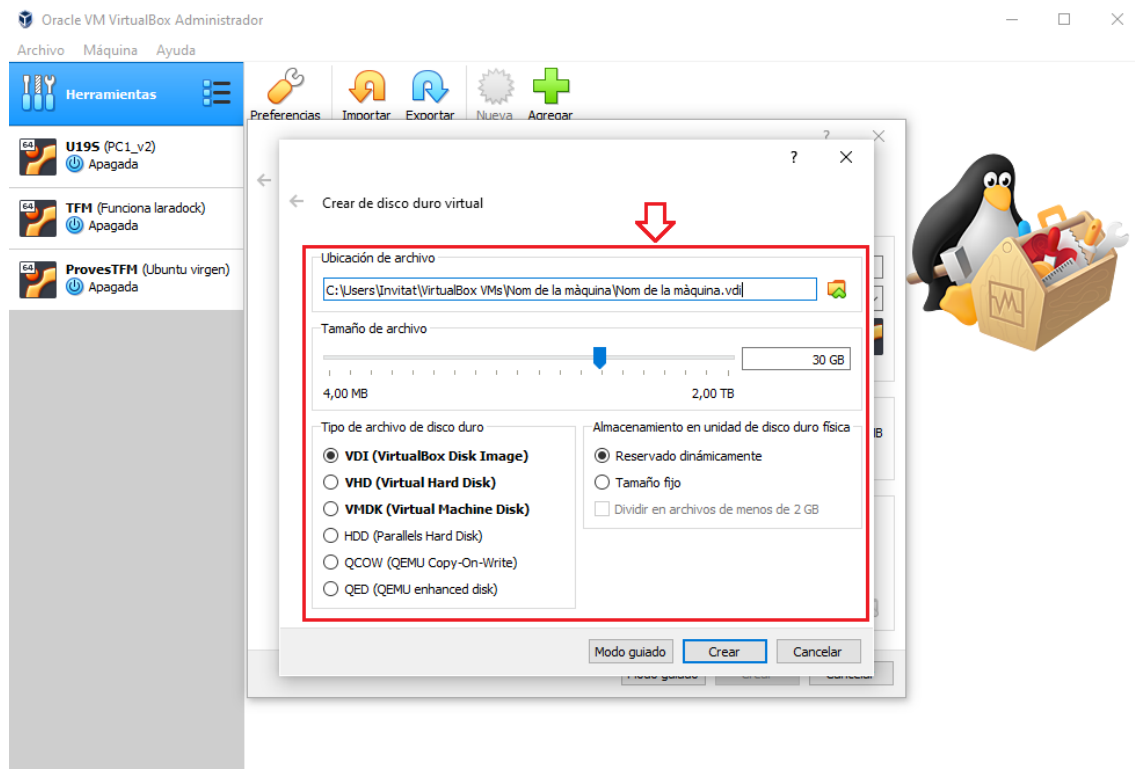
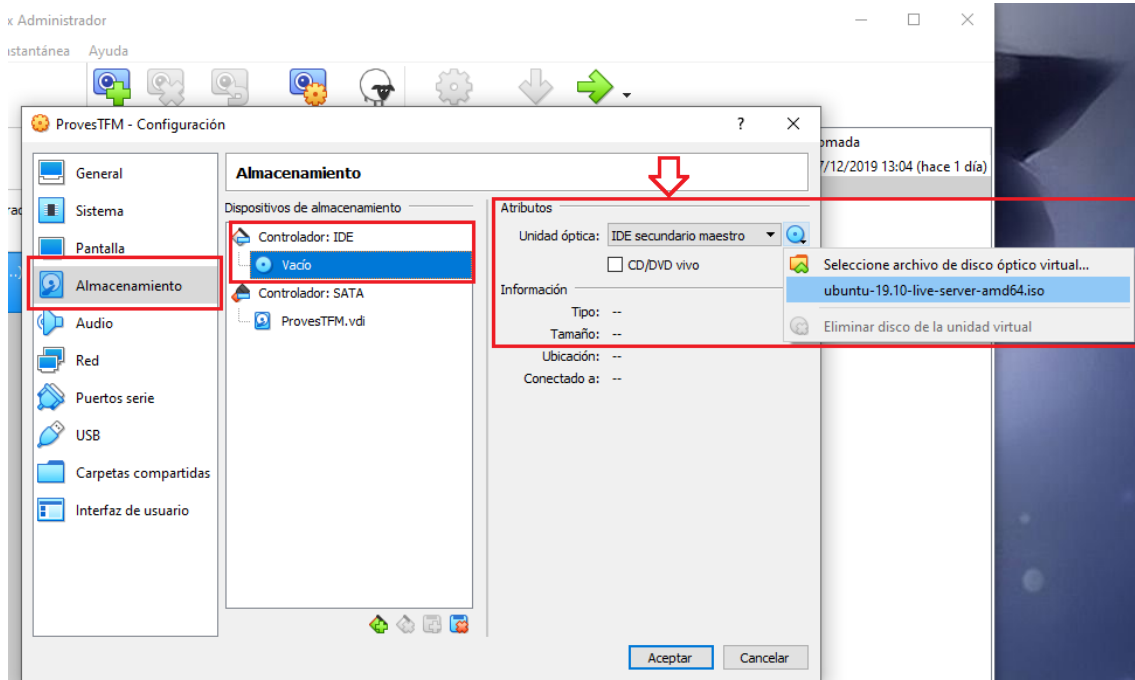


Figura 5: Configuració del disc dur.

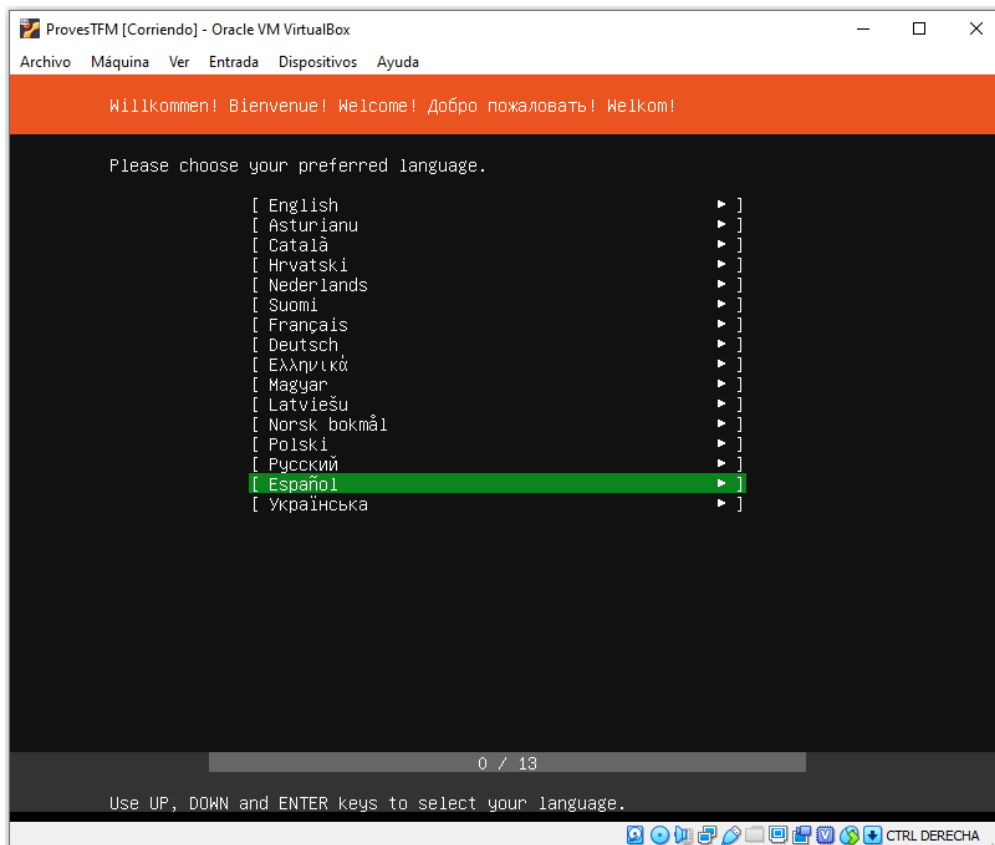
Un cop finalitzat tot el procés, a continuació s'ha d'agafar la imatge del sistema operatiu Ubuntu Server i afegir-ho com a disc per que llegeixi la ISO a l'arrencada (Figura 6):



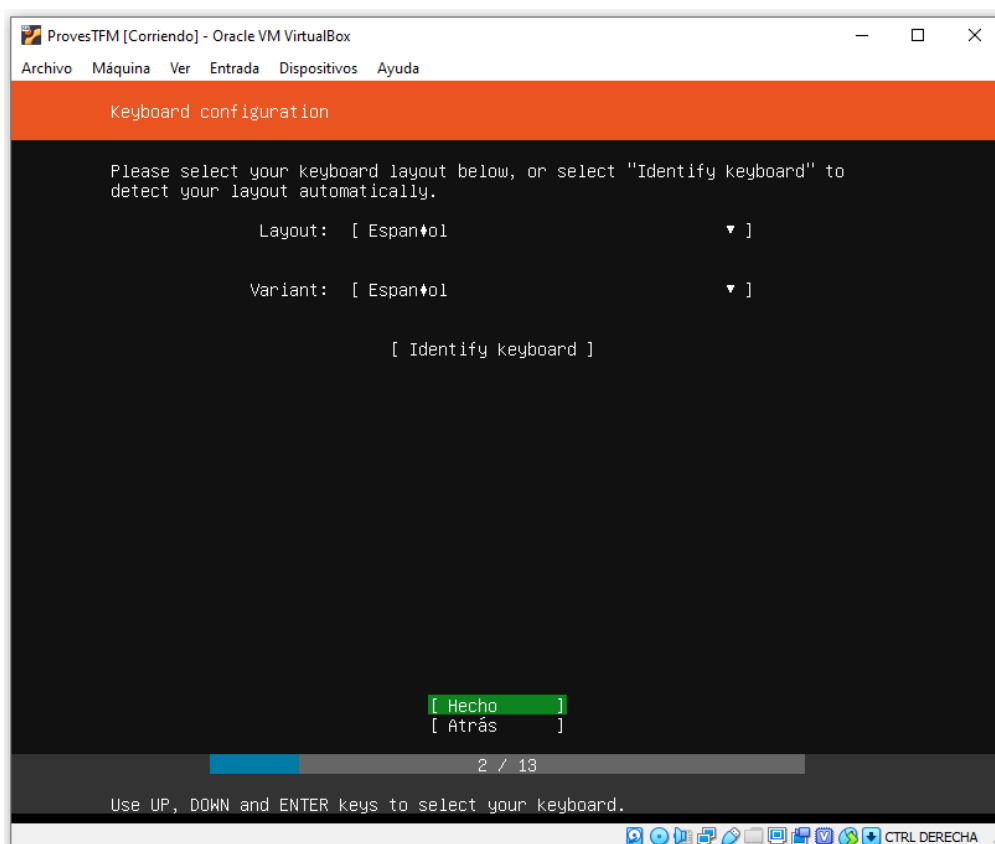
**Figura 6:** Afegir imatge ISO del sistema operatiu.

En aquest punt ja s'ha creat la màquina virtual i es procedeix a encendre i seguir els passos de l'assistent que ofereix Ubuntu.

Primer de tot, comença per seleccionar l'idioma tant del sistema operatiu com del teclat:



**Figura 7:** Instal·lació del sistema operatiu.



**Figura 8:** Configuració idioma de teclat.

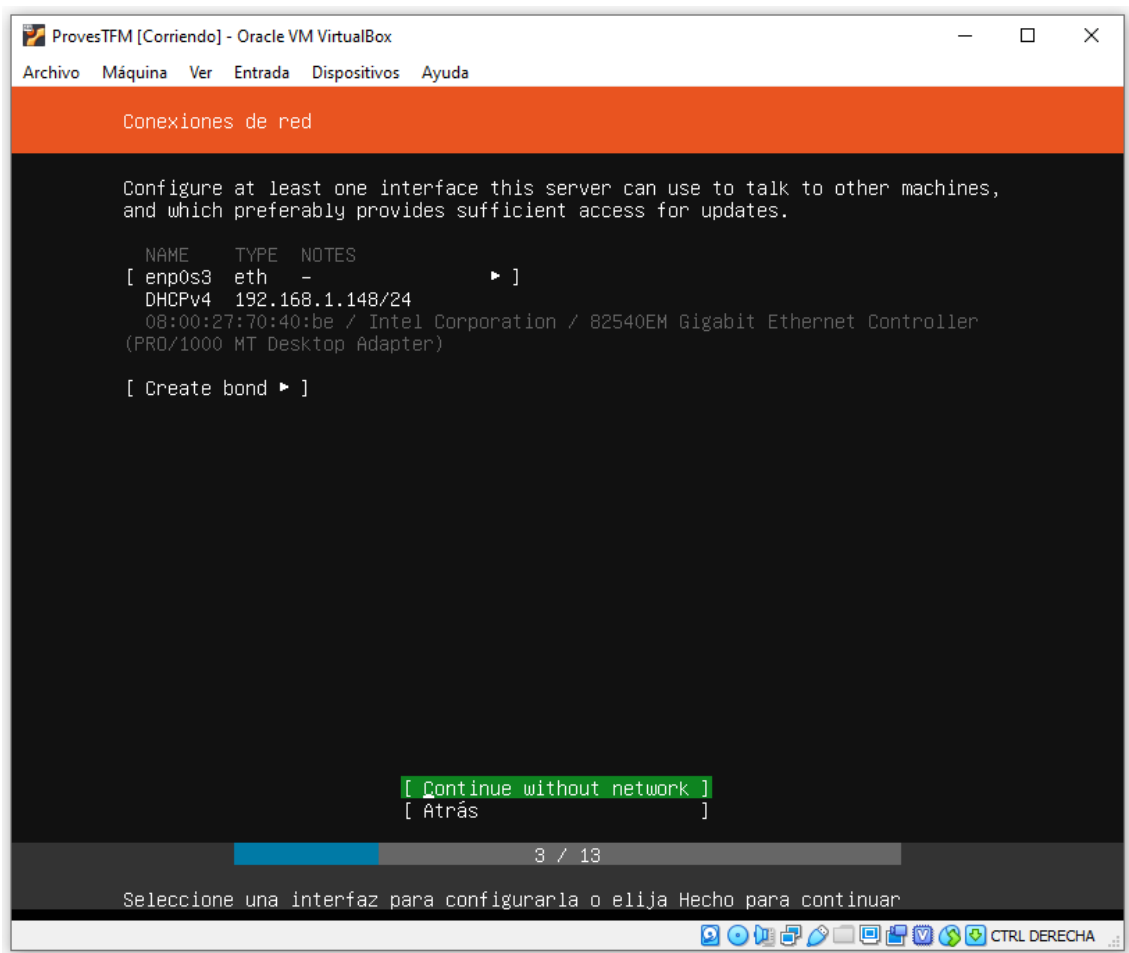


A continuació, demanarà configurar la xarxa del servidor.

Si el cas és fer proves, el que es pot fer és deixar la configuració tal i com ve per defecte amb la opció de DHCP activat i que doni la IP que vulgui que no hi haurà cap impediment per més endavant.

Si el cas és deixar com a servidor permanent, es recomana treure la opció de DHCP i configurar una IP estàtica seguint l'assistent d'Ubuntu.

Per les proves, el que es farà és deixar la opció per defecte tal i com es mostra a la figura 9:



**Figura 9:** Configuració de la xarxa del servidor.

Arribats a aquest punt, passa a configurar el disc dur. Igual que el pas anterior, si es vol fer proves només cal seguir l'assistent i deixar la configuració que dona per defecte. Però, si es vol fer una configuració més acurada, en aquest punt Ubuntu dona la opció de configurar el disc al gust de l'usuari, especificant la mida de les particions que el servidor proporciona: la partició d'arrencada "Grub", la partició de dades i la partició "swap" d'intercanvi de dades de memòria virtual, que fa funció de RAM.

A continuació es mostren les captures que respecta a la configuració per defecte:

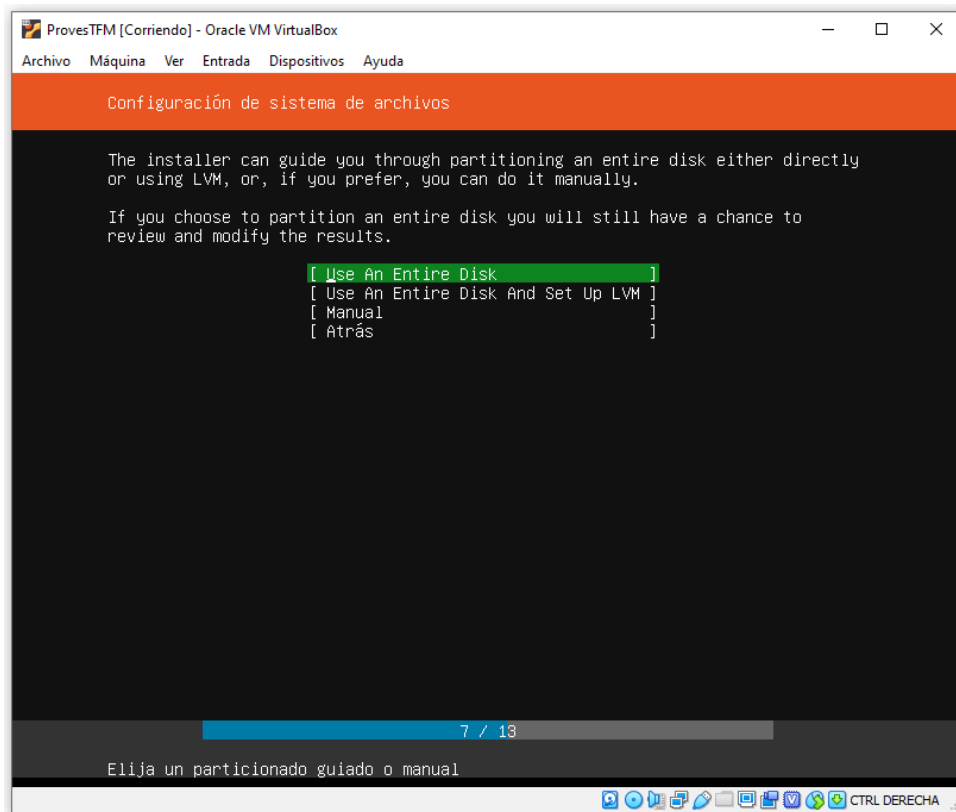


Figura 10: Selecció de l'ús del disc dur.

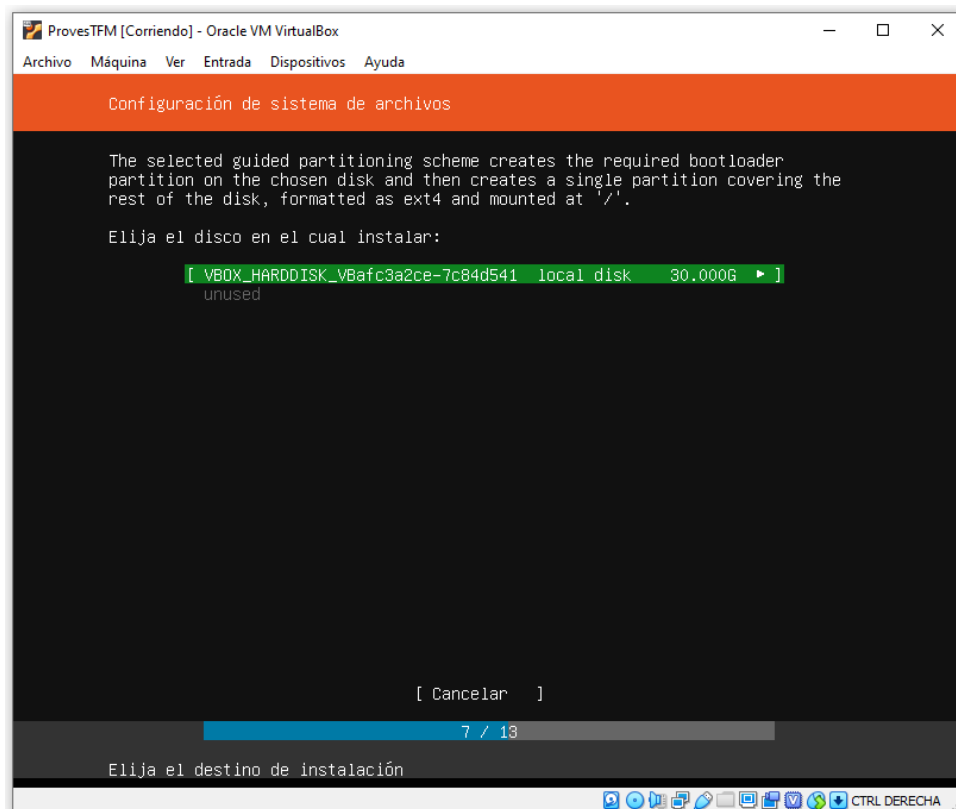


Figura 11: Selecció del disc dur existent.

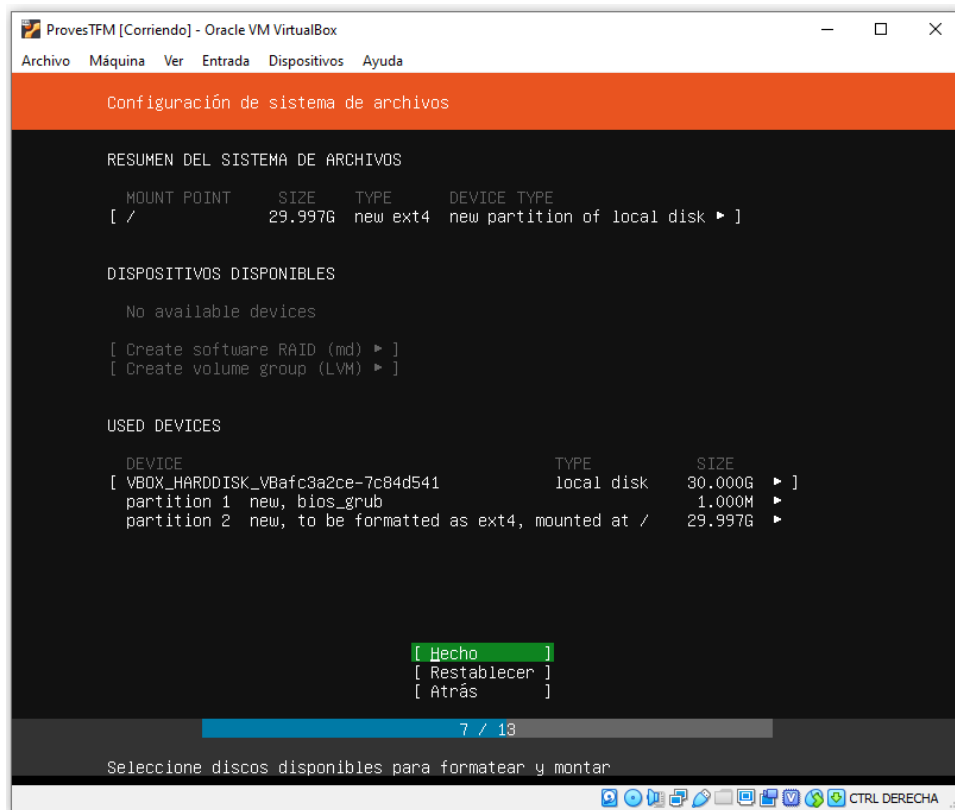


Figura 12: Resum de les particions fetes per defecte.

Feta la configuració del disc, a continuació l'assistent mostra un petit formulari el qual s'ha d'omplir per donar nom al servidor i crear un usuari per accedir al sistema:

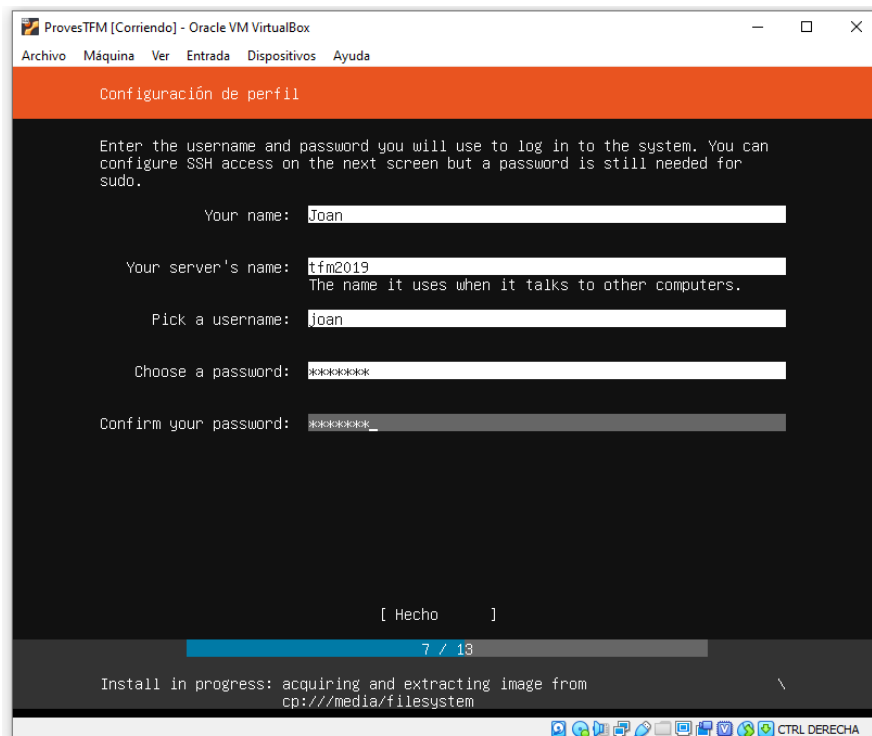
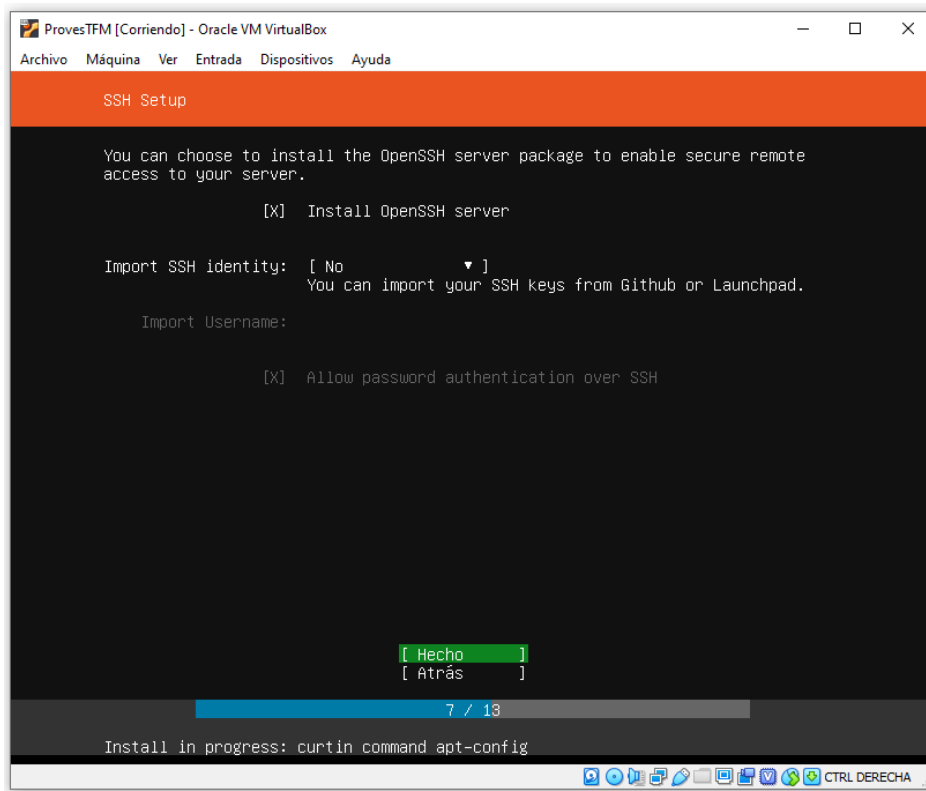


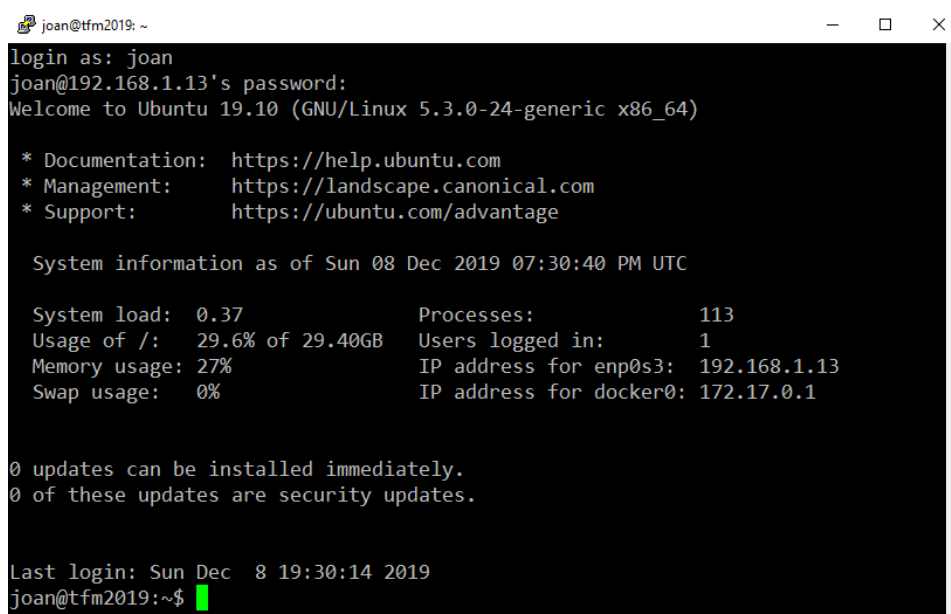
Figura 13: Formulari de noms i usuari.

Finalment l'últim pas, l'assistent dona opció a instal·lar SSH. Donat que si en un futur es vol accedir al servidor mitjançant eines com "FileZilla" o "Putty", openSSH ha d'estar instal·lat, ja que mitjançant el port 22 crea una connexió segura i dona accés al sistema amb usuari i contrasenya, detallat al formulari anterior:



**Figura 14:** Instal·lació de l'openSSH.

Finalitzat tot el procés, al accedir al sistema demanarà usuari i contrasenya i quedarà tal que així:



**Figura 15:** Accés al sistema un cop instal·lat.

## DOCKER: Instal·lació

Donat que l'objectiu de l'eina de suport al programador és generar i configurar un servidor per un entorn web predefinit, i de manera desatesa, Docker és i serà la peça clau per la qual faci que les plantilles web siguin operatives.

Per fer-la funcionar, s'ha creat un script anomenat "*script\_laravel.sh*" el qual conté totes les comandes necessàries (executar: **sudo bash ./jhernandezch\_script.sh**):

1 – Primer de tot, descarrega les planes web de GitHub i dona els permisos necessaris que Laravel necessita per funcionar (directoris "**storage**" i "**cache**"):

```

1  #!/bin/bash
2  echo "---> 1 - Instal·lació laravel Admin"
3  sudo git clone https://github.com/jrhernandezch/laravel_admin.git /var/www/laravel_admin
4  sudo chmod 777 -R /var/www/laravel_admin/storage
5  sudo chmod 777 -R /var/www/laravel_admin/bootstrap/cache
6  echo "Completada"
7
8  echo "---> 2 - Instal·lació laravel Template"
9  sudo git clone https://github.com/jrhernandezch/laravel_template.git /var/www/laravel_template
10 sudo chmod 777 -R /var/www/laravel_template/storage
11 sudo chmod 777 -R /var/www/laravel_template/bootstrap/cache
12 echo "Completada"

```

**Figura 16:** Clonació i assignació de permisos de les planes web.

2 – Quan finalitzi el procés, el següent pas és instal·lar Docker i Docker-compose al servidor. En primer lloc, Docker és el software que es farà servir per generar els contenidors necessaris que després Laradock executarà. I en segon lloc, Docker-compose és el software que llegirà i executarà cada arxiu de configuració que tindrà lloc cada contenidor. Aquest arxiu serveix per especificar els paràmetres que necessita un contenidor concret, com per exemple la IP, el port, variables d'entorn o volum de dades per la persistència:

```

14 echo "---> 3 - Instal·lació Docker"
15 sudo apt-get update
16 sudo apt-get remove docker docker-engine docker.io
17 sudo apt install docker.io
18 sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
19 sudo chmod +x /usr/local/bin/docker-compose
20 sudo systemctl start docker
21 sudo systemctl enable docker
22 echo "Completada"

```

**Figura 17:** Comandes d'instal·lació de Docker i docker-compose.

3 – Després de les instal·lacions anteriors, el següent pas és clonar el paquet de recursos de Laradock[1] per la posterior configuració i execució dels contenidors que necessita, que per aquest cas són: "*apache2*", "*mysql*", "*phpmyadmin*" i "*workspace*". Aquest darrer contenidor conté l'espai de feina del projecte Laravel.

```

24 echo "---> 4 - Instal·lació Laradock a Laravel template"
25 sudo git clone https://github.com/Laradock/laradock.git /var/www/laradock
26 echo "Completada"

```

**Figura 18:** Clonació del paquet Laradock.

4 – Fetes totes les instal·lacions pertinents i abans d'executar els contenidors Docker, dins del projecte de la plantilla Laravel “**Laravel\_template**”, s'allotja un directori anomenat “**laradock\_resources**” que conté tots els arxius de configuració que prèviament s'ha configurat per donar servei a les planes web:

- **.env**: Arxiu que especifica les variables d'entorn que Laradock necessita per ser executat.
- **createtable.sql**: Arxiu de configuració de MySQL que modifica la base de dades per a que s'accedeixi amb l'usuari root.
- **my.cnf**: Arxiu de configuració de MySQL que indica quin és l'usuari d'accés per defecte.
- **Laravel\_admin.conf**: Arxiu de configuració de lloc web d'apache2 mitjançant “virtualhost” per a la pàgina del backoffice.
- **Laravel\_template.conf**: Arxiu de configuració de lloc web d'apache2 mitjançant “Virtualhost” per a la pàgina del front-end.
- **Laravel\_business\_04122019.sql**: Base de dades de la pàgina web que s'haurà d'importar mitjançant phpmyadmin amb port 8080 a la URL(temporalment abans de ser automàtic).

Aquest arxius han de ser copiats als seus directoris corresponents (figura 19) per a que a la crida dels contenidors Docker llegeixi la configuració pertinent:

```
28 echo "--> 5 - Instal·lació recursos laradock"
29 sudo cp /var/www/laravel_template/laradock_resources/.env /var/www/laradock/.env
30 sudo cp /var/www/laravel_template/laradock_resources/my.cnf /var/www/laradock/mysql/my.cnf
31 sudo cp /var/www/laravel_template/laradock_resources/createdb.sql /var/www/laradock/mysql/docker-entrypoint-initdb.d/createdb.sql
32 sudo cp /var/www/laravel_template/laradock_resources/laravel_admin.conf /var/www/laradock/apache2/sites/laravel_admin.conf
33 sudo cp /var/www/laravel_template/laradock_resources/laravel_template.conf /var/www/laradock/apache2/sites/laravel_template.conf
34 echo "Completada"
```

**Figura 19:** Còpia de la configuració predefinida.

5 – Finalment, l'script finalitza amb la crida dels contenidors Docker de la següent manera:

```
36 echo "--> Instal·lació finalitzada. Inicialitzant contenidors Docker"
37 cd /var/www/laradock
38 sudo docker-compose up -d apache2
39 sudo docker-compose up -d mysql
40 sudo docker-compose up -d phpmyadmin
41 sudo docker-compose up -d workspace
42 sudo docker ps
```

**Figura 20:** Executar contenidors Docker.

Ha de quedar una pantalla similar a la següent:

```
Creating network "laradock_frontend" with driver "bridge"
Creating network "laradock_backend" with driver "bridge"
Creating network "laradock_default" with the default driver
Creating laradock_docker-in-docker_1 ... done
Creating laradock_mysql_1 ... done
Creating laradock_workspace_1 ... done
Creating laradock_phpmyadmin_1 ... done
Creating laradock_php-fpm_1 ... done
Creating laradock_apache2_1 ... done
joan@tfm2019: /var/www/laradock$
```

Figura 21: Contenedors carregats correctament.

Un cop que funcioni tots els contenidors, s'ha d'accedir a la URL de "phpmyadmin":

- [http://IP\\_SERVIDOR:8080](http://IP_SERVIDOR:8080)
- <http://laravel.template:8080>

Amb l'objectiu d'anar a la base de dades "Laravel\_business" i importar l'arxiu de taules "Laravel\_business\_07122019.sql" amb usuari i contrasenya "root":



Figura 21: Web phpMyAdmin.

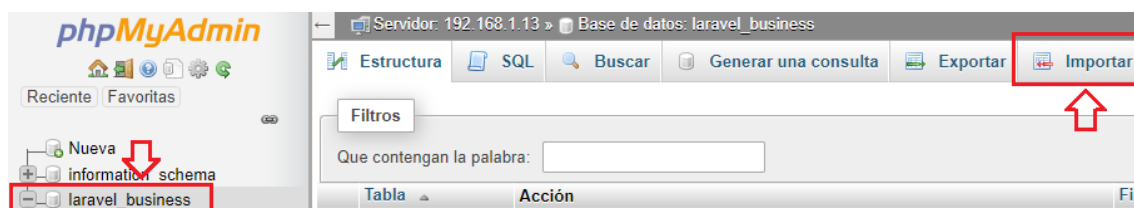


Figura 22: Importar arxiu base de dades.

## DOCKER: Comandes

Dins de Docker, existeixen comandes[2] bàsiques i útils que poden ajudar a gestionar els contenidors:

- **sudo docker ps** : Llista els contenidors actius amb informació bàsica.

```
joan@tfm2019:/var/www/laradock$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS                               NAMES
96d595a9caca       laradock_apache2   "/opt/docker/bin/ent..." 10 minutes ago    Up 10 minutes      0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp   laradock_apache2_1
f54dbef3a0         laradock_php-fpm   "docker-php-entrypoi..." 10 minutes ago    Up 10 minutes      9000/tcp                             laradock_php-fpm_1
d9d67bcec5f9       laradock_phpmyadmin "/docker-entrypoint..." 10 minutes ago    Up 10 minutes      0.0.0.0:8080->80/tcp                 laradock_phpmyadmin_1
792787175da0       laradock_workspace "/sbin/my_init"         10 minutes ago    Up 10 minutes      0.0.0.0:2222->22/tcp                 laradock_workspace_1
fb5bd6835317       laradock_mysql     "docker-entrypoint.s..." 10 minutes ago    Up 10 minutes      0.0.0.0:3306->3306/tcp, 3306/tcp     laradock_mysql_1
1a17cfcb7fd8       docker:dind        "dockerd-entrypoint..." 10 minutes ago    Up 10 minutes      2375-2376/tcp                       laradock_docker-in-docker_1
joan@tfm2019:/var/www/laradock$
```

Figura 21: Contenedors actius.

- **sudo docker-compose up -d apache2 mysql phpmyadmin workspace** : Inicialitza / executa tots els contenidors nomenats a la comanda sense veure logs.
- **sudo docker-compose up apache2 mysql phpmyadmin workspace** : Igual que l'anterior però amb els logs visibles a temps real.

```
joan@tfm2019:/var/www/laradock$ sudo docker-compose up -d apache2 mysql phpmyadmin workspace
Creating network "laradock_frontend" with driver "bridge"
Creating network "laradock_backend" with driver "bridge"
Creating network "laradock_default" with the default driver
Creating laradock_docker-in-docker_1 ... done
Creating laradock_mysql_1 ... done
Creating laradock_workspace_1 ... done
Creating laradock_phpmyadmin_1 ... done
Creating laradock_php-fpm_1 ... done
Creating laradock_apache2_1 ... done
joan@tfm2019:/var/www/laradock$
```

Figura 22: Activar contenidors.

- **sudo docker-compose down** : Desactiva tots els contenidors actius.

```
joan@tfm2019:/var/www/laradock$ sudo docker-compose down
Stopping laradock_apache2_1 ... done
Stopping laradock_php-fpm_1 ... done
Stopping laradock_phpmyadmin_1 ... done
Stopping laradock_workspace_1 ... done
Stopping laradock_mysql_1 ... done
Stopping laradock_docker-in-docker_1 ... done
Removing laradock_apache2_1 ... done
Removing laradock_php-fpm_1 ... done
Removing laradock_phpmyadmin_1 ... done
Removing laradock_workspace_1 ... done
Removing laradock_mysql_1 ... done
Removing laradock_docker-in-docker_1 ... done
Removing network laradock_frontend
Removing network laradock_backend
Removing network laradock_default
joan@tfm2019:/var/www/laradock$
```

Figura 23: Desactivar contenidors.



- **sudo docker-compose restart** [nom\_contenedor] : Reinicia un contenidor concret.

```
joan@tfm2019:/var/www/laradock$ sudo docker-compose restart apache2
Restarting laradock_apache2_1 ... done
joan@tfm2019:/var/www/laradock$
```

**Figura 24:** Reiniciar contenidor.

## PLANTILLA WEB: Captures de pantalla

Pel que fa a les direccions URL de proves per accedir a les planes web, a la configuració d'**apache2** s'han configurat 2 noms que s'han d'afegir de manera local a l'arxiu "**hosts**":

```
22 192.168.1.13 laravel.template.cat
23 192.168.1.13 laravel.admin.cat
```

**Figura 25:** Arxiu hosts de Windows.

En els dos casos, les planes web tenen accés al port 80:

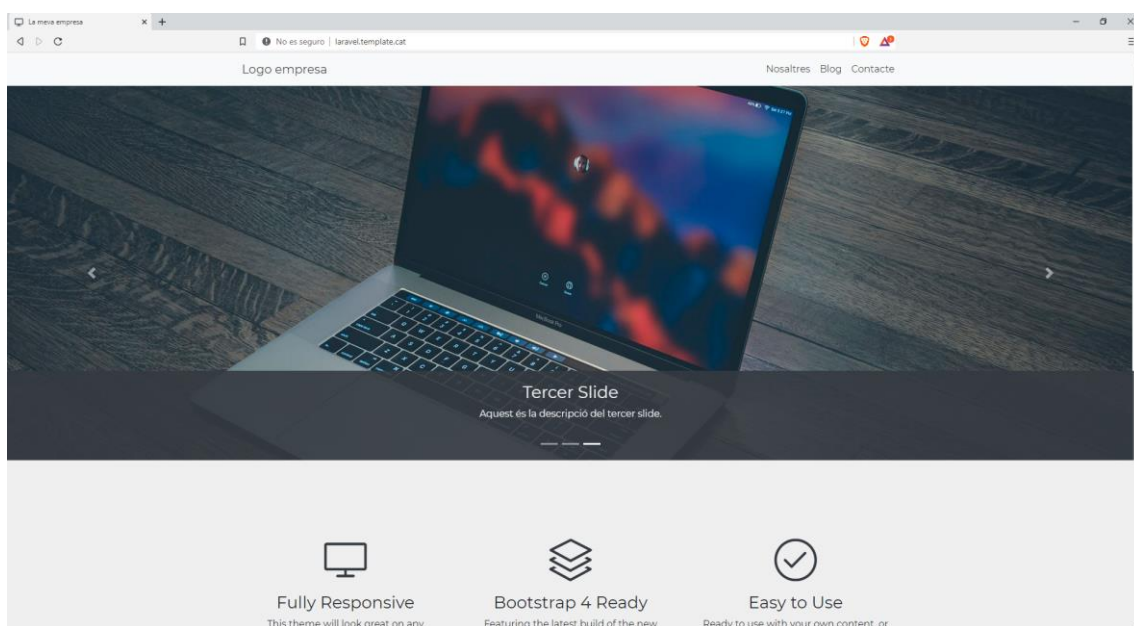
```
laravel_admin.conf x
1 <VirtualHost *:80>
2   ServerName laravel.admin.cat
3   DocumentRoot /var/www/laravel_admin/public/
4   Options Indexes FollowSymLinks
5
6   <Directory "/var/www/laravel_admin/public/">
7     AllowOverride All
8     <IfVersion < 2.4>
9       Allow from all
10    </IfVersion>
11    <IfVersion >= 2.4>
12      Require all granted
13    </IfVersion>
14  </Directory>
15
16 </VirtualHost>
```

**Figura 26:** Arxiu de configuració laravel\_admin.conf (backoffice).

```
laravel_template.conf x
1 <VirtualHost *:80>
2   ServerName laravel.template.cat
3   DocumentRoot /var/www/laravel_template/public/
4   Options Indexes FollowSymlinks
5
6   <Directory "/var/www/laravel_template/public/">
7     AllowOverride All
8     <IfVersion < 2.4>
9       Allow from all
10    </IfVersion>
11    <IfVersion >= 2.4>
12      Require all granted
13    </IfVersion>
14  </Directory>
15
16 </VirtualHost>
```

**Figura 27:** Arxiu de configuració *laravel\_template.conf* (front-end).

Amb tot configurat, hauria de sortir com a resultat una pantalla com la següent:



**Figura 28:** Plantilla web.

## BIBLIOGRAFIA

[1] Laradock, Paquet Docker per Laravel <https://laradock.io> Vist per última vegada el 9 de desembre del 2019.

[2] Docker, Base de comandes per Docker <https://docs.docker.com/engine/reference/commandline/docker> Vist per última vegada el 0 de desembre del 2019.