

 gemartin99

2 weeks ago

last month

last month

2 years ago

Born2beroot-Tutorial

This guide has versions in different languages. Choose the one you prefer.

English version

 [CLICK HERE](#)

Versão portuguesa

 [CLIQUE AQUI](#)

Índice

1. [Descargar imagen de la maquina virtual](#) 
2. [Instalación de la maquina](#) 

 - 2.1 [Instalación de la maquina con Virtual Box](#) 
 - 2.2 [Instalación de la maquina con VMware](#) 

3. [Instalación Debian](#) 
4. [Configuración de la máquina virtual](#) 
 - 4.1 [Instalación de sudo y configuración de usuarios y grupos](#) 
 - 4.2 [Instalación y configuración de SSH](#) 
 - 4.3 [Instalación y configuración de UFW](#) 
 - 4.4 [Configurar contraseña fuerte para sudo](#) 
 - 4.5 [Configuración de política de contraseñas fuerte](#) 
 - 4.6 [Conectarse via SSH](#) 
5. [Script](#) 

 - 5.1 [Resultado total del script](#) 

6. [Crontab](#) 
7. [Signature.txt](#) 
8. [Bonus](#) 
 - 8.1 [Particionado manual del disco](#) 
 - 8.2 [Wordpress y configuración de servicios](#) 
 - 8.3 [Servicio adicional](#) 

9. Hoja de corrección ↴

9.1 [Respuestas de la evaluación 100](#)

9.2 [Comandos de la evaluación 📈](#)

10. [Tester OK](#)

’1- Descargar imagen de la maquina virtual ⚙

[Click aquí](#) para redireccionarte a la URL donde puedes descargar la ISO de manera segura.

’2- Instalacion de la maquina 🏠

Según el subject es OBLIGATORIO hacer este proyecto con Virtual Box. Pero si por algún problema técnico en tu campus no está disponible Virt Box. Este tutorial cuenta con una versión para VMware.

Si quieres hacer la instalación con VMware haz [Click aquí](#)

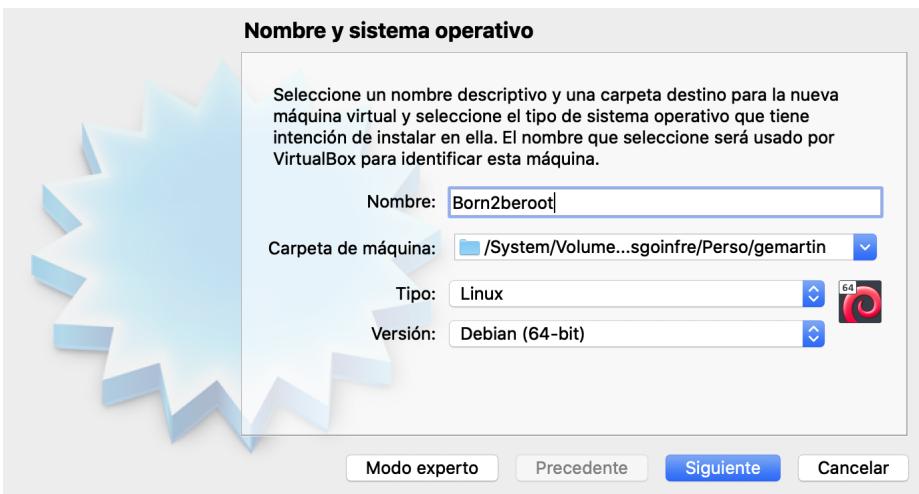
’2-1 Instalacion de la maquina con Virtual Box 🏠

Para realizar la instalación se requiere de un software de virtualización. En este tutorial haremos uso de [VirtualBox](#). Si ya tienes VirtualBox instalado dispones de la ISO Debian ya podemos empezar con el tutorial.

1 ° Debemos abrir VirtualBox y pinchar sobre [Nueva](#)



2 ° Escogemos el nombre de nuestra máquina y la carpeta donde estará ubicada. Importante introducir la máquina dentro de la carpeta sgoinf que si no la ubicamos ahí nos quedaremos sin espacio y fallará la instalación (dependiendo del campus la ruta de sgoinfre puede cambiar).



3º Seleccionamos la cantidad de memoria RAM que reservaremos para la máquina.



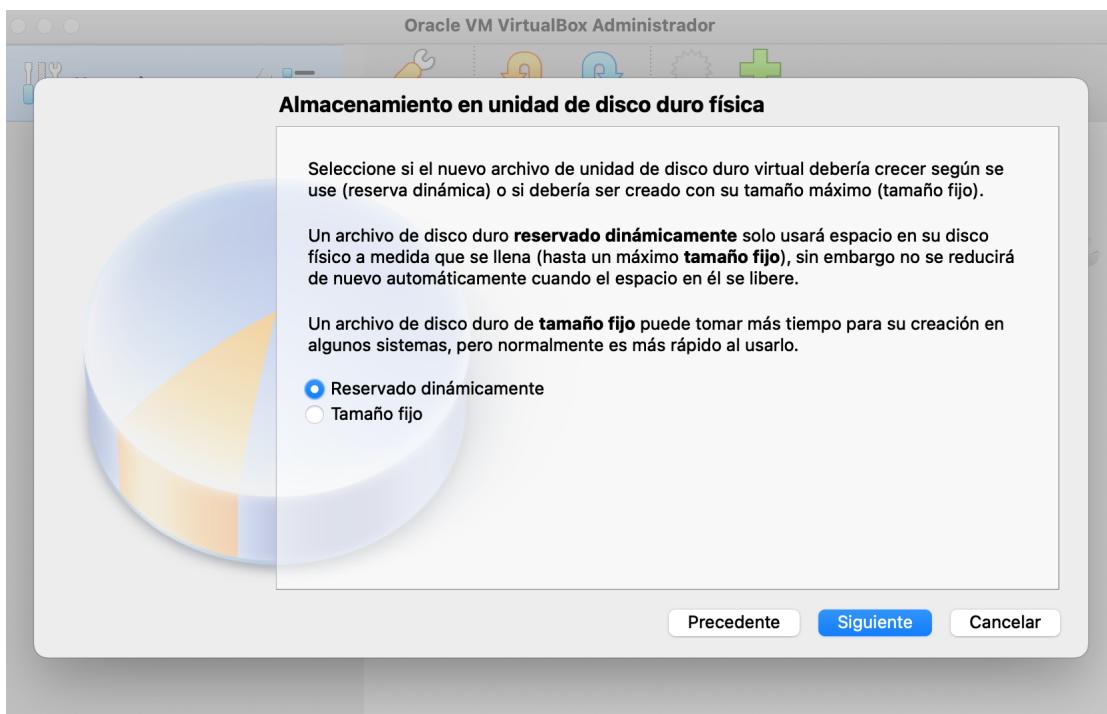
4º Seleccionamos la segunda opción para así crear un disco duro virtual ahora.



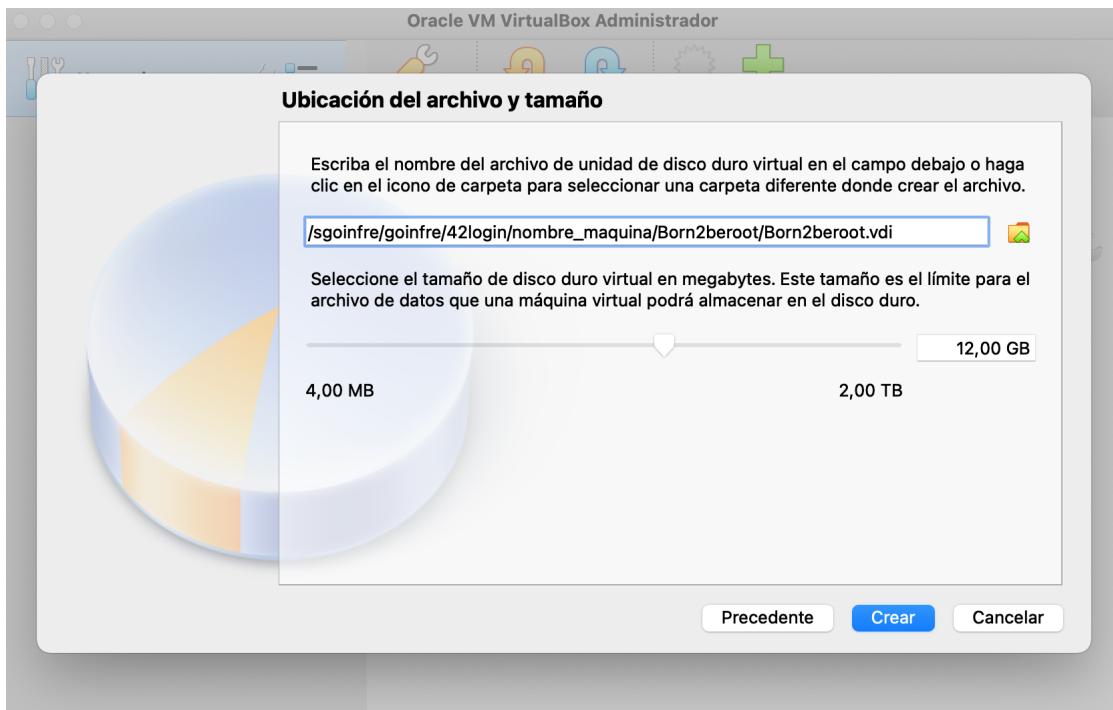
5º Escogemos la primera opción **VDI** ya que nos hemos descargado una imagen de disco.



6º Seleccionamos la primera opción **Reservado dinámicamente** para que así se vaya reservando memoria en la máquina real según vayamos utilizandola en la virtual hasta llegado al límite máximo disponible en la virtual.



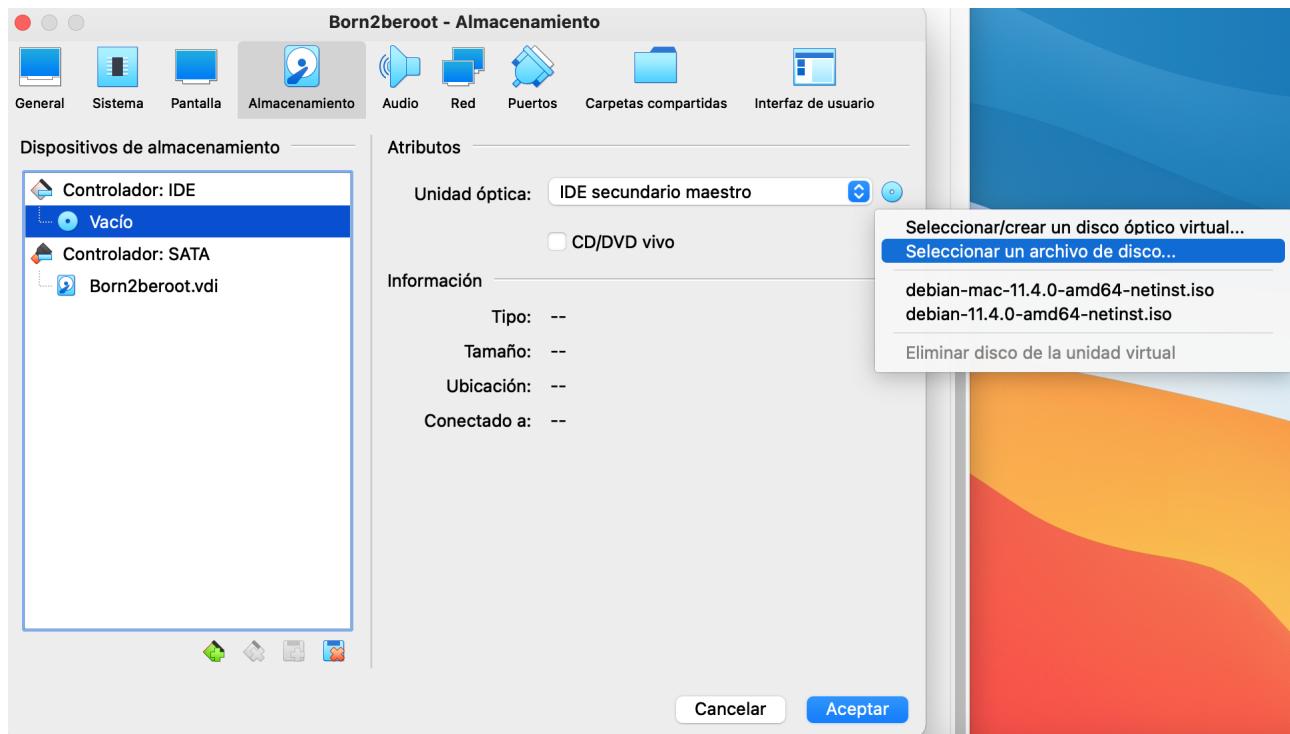
7º Una vez hayamos establecido la cantidad recomendada **12 GB** deberemos darle a **crear**. Si hacemos el bonus seleccionaremos **30 GB**.



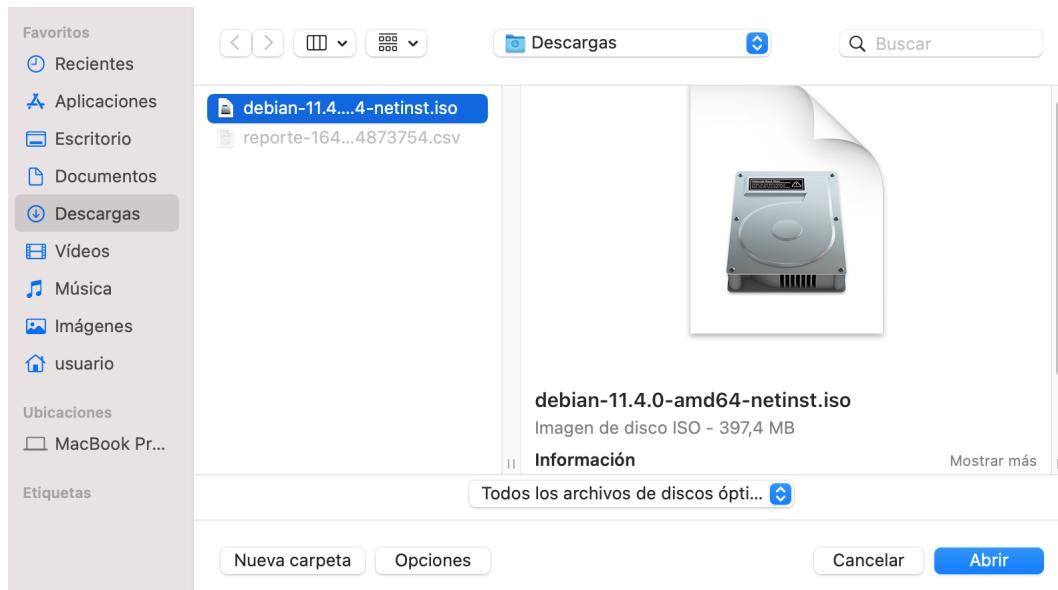
8 ° Puede parecer que ya hemos terminado la instalación , pero todavía faltan un par de pasos más. Debemos darle a configuración



9 ° Acto seguido pincharemos encima de **Almacenamiento** , volveremos a pinchar sobre el emoticono que se encuentra a la derecha y de nuev pincharemos sobre **Seleccionar un archivo de disco** .



10. Seleccionaremos la ISO que acabamos de descargar y le damos a **Abrir** y después le daremos a **Aceptar**.



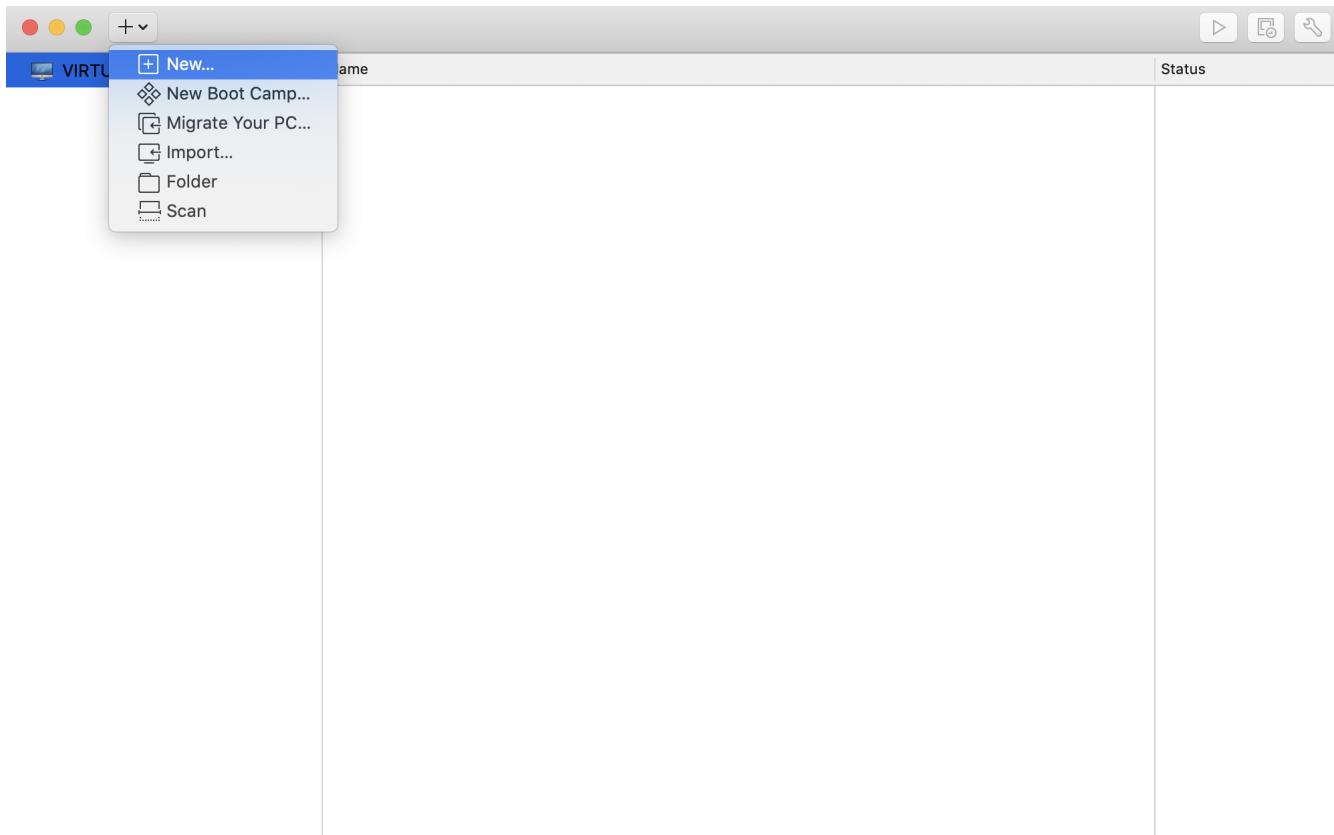
11. Una vez completados todos los pasos anteriores ya podemos **Iniciar** nuestra máquina virtual.



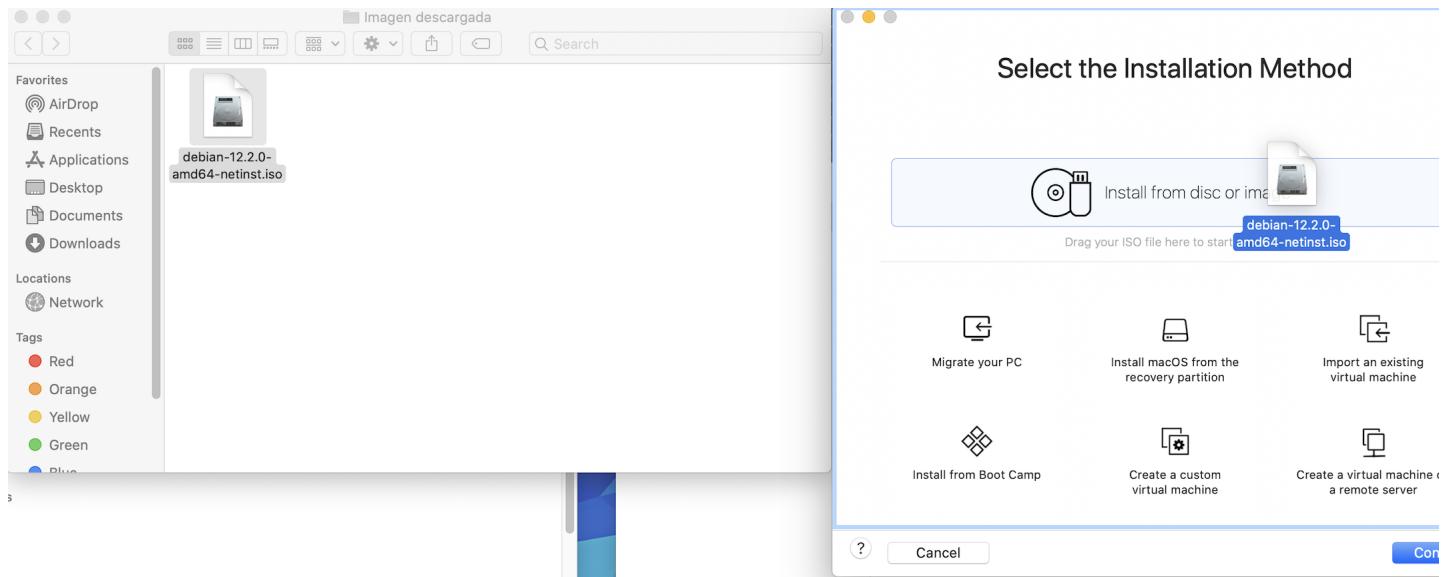
Para dirigirte a la Instalación de Debian directamente [Click aquí](#)

2-2 Instalacion de la maquina con VMware

1º Debemos abrir VMware y pinchar sobre **New**.



2º Se nos habrá abierto una pestaña. Ahora debemos arrastrar la ISO de Debian que hemos descargado en el paso 1.



3º Le damos a **continue** y indicamos el sistema operativo que usaremos.



Create a New Virtual Machine

This will guide you through installing Windows or another operating system in a virtual machine on your Mac.



Choose an operating system installation disc or image:

debian-12.2.0-amd64-netinst.iso [Show in Finder](#)

[Use another disc or disc image...](#)

4º Seleccionamos **Debian 10.x 64 bits**.



Choose Operating System

Select the operating system to be used in this virtual machine.



Select the operating system for this virtual machine:

Microsoft Windows	> Asianux Server 3
Linux	> CentOS 8 64-bit
Apple OS X	> CentOS 7 64-bit
VMware ESX	> CentOS 6 64-bit
Other	> CentOS 6 CentOS version 5 and earlier 64-bit CentOS version 5 and earlier Debian 10.x 64-bit Debian 10.x Debian 9.x 64-bit Debian 9.x

(?) Cancel

Go Back

Continue

5º Seleccionaremos `Legacy BIOS`. Este paso es importante ya que si escogemos UEFI o UEFI Secure Boot las particiones no quedarán como específica el subject ya que crea una partición nueva. Con la opción Legacy Bios no se creará ninguna partición específica.



Choose Firmware Type

Select the firmware type to be used to boot this virtual machine.



Specify the boot firmware:

- Legacy BIOS
 UEFI
 UEFI Secure Boot

(?) Cancel

Go Back

Continue

6º Antes de finalizar la instalación debemos escoger la ruta donde almacenaremos nuestra maquina virtual. Le daremos a `Customize Settings`

Finish

The configuration of the virtual machine is now complete.



Virtual Machine Summary

Guest Operating System Debian 10.x 64-bit
New Hard Disk Capacity 20 GB
Memory 2 GB
Networking Share with my Mac (NAT)
Device Summary CD/DVD, USB Controller, Printer, Sound Card

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

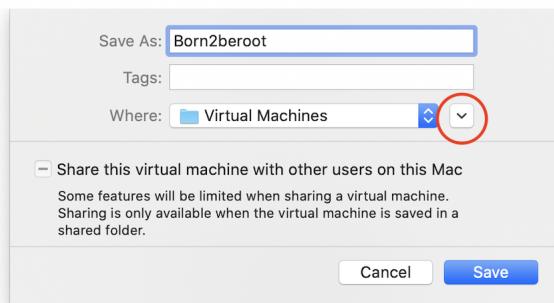
[Customize Settings](#)

? Cancel

Go Back

Finish

7 ° Cambiamos el nombre de la maquina a Born2beroot y pincharemos sobre la flecha para poder escoger la ruta donde almacenaremos la ma



Cancel Save

Networking Share with my Mac (NAT)

Device Summary CD/DVD, USB Controller, Printer, Sound Card

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

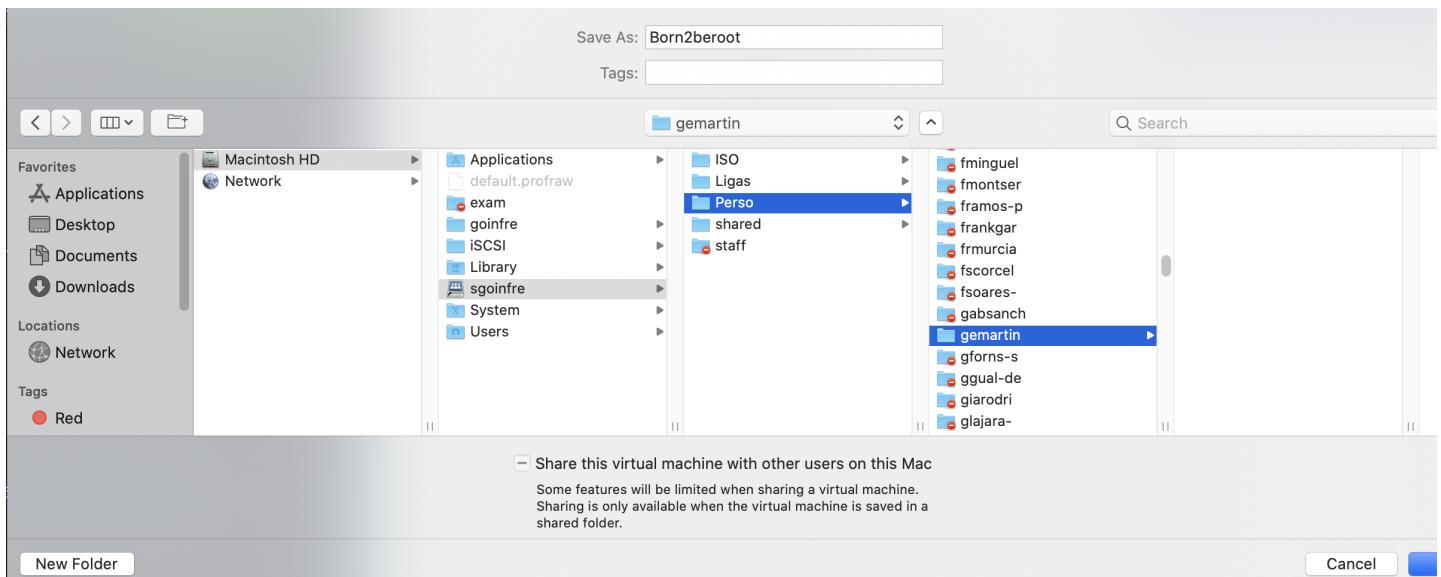
[Customize Settings](#)

? Cancel

Go Back

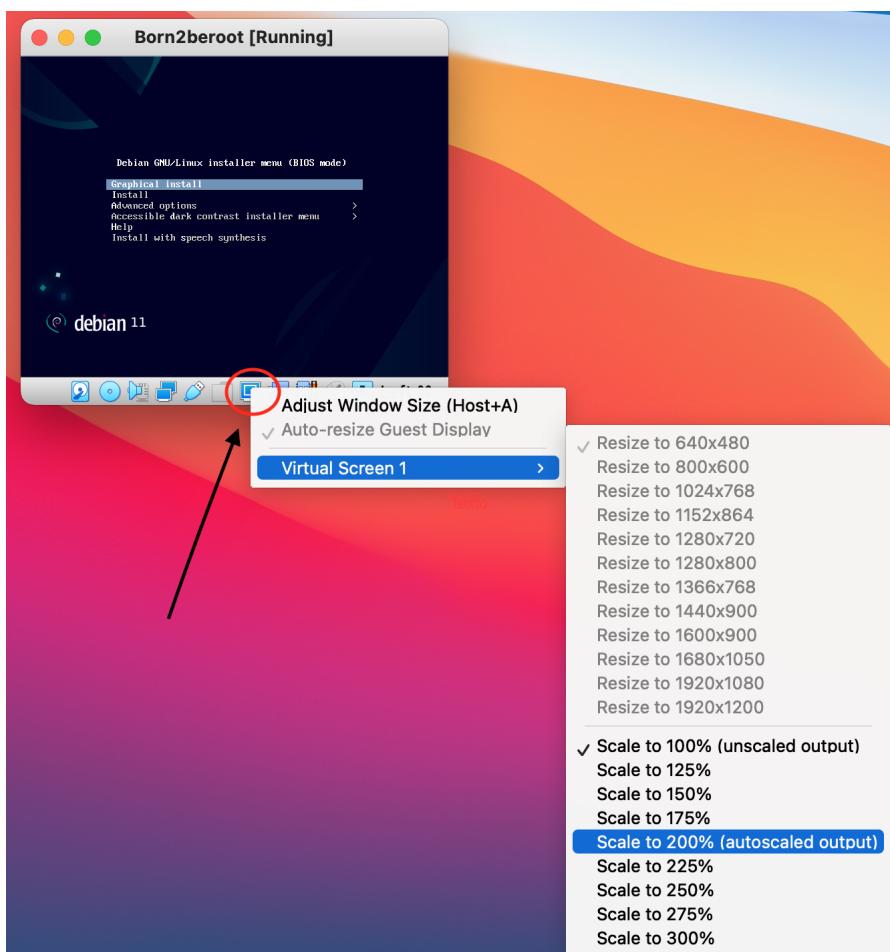
Finish

8 ° Para que no nos quite espacio de nuestro usuario la almacenaremos en el sgoinfre, es importante que crees una carpeta con tu login y que los permisos necesarios. Una vez la tengas almacenaremos nuestra maquina virtual en esa ruta. En mi caso esta es la ruta, quizás en tu campus diferente!



3- Instalación Debian 9

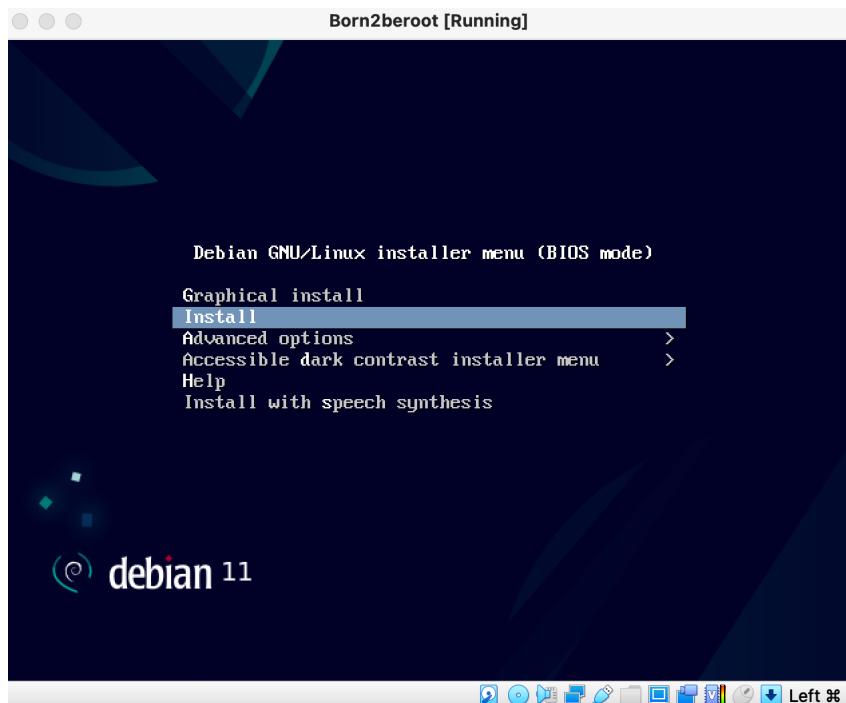
► Espera ! Tu vista es muy importante !! Para poder hacer la ventana más grande debes hacer lo siguiente:



Utiliza la tecla `command` para que la captura del ratón pase de la maquina real a la virtual y al revés.

’ Sigamos con la instalación 

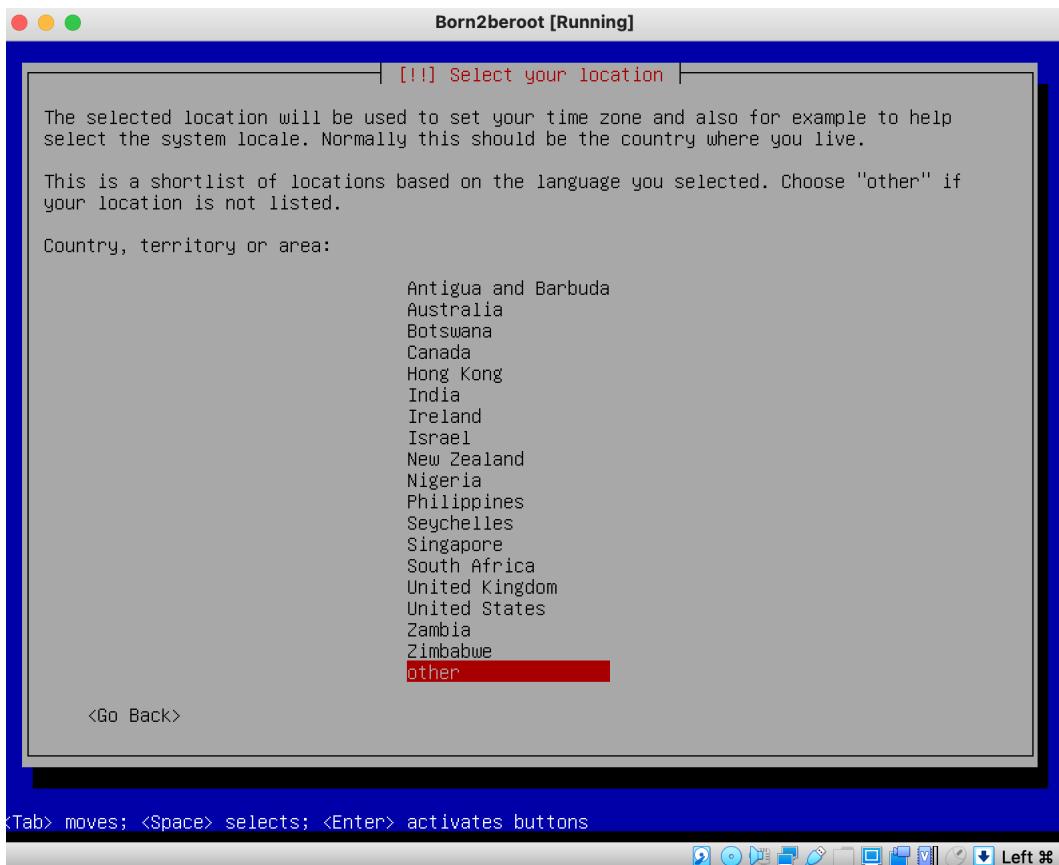
1 ° Escogeremos la versión sin interfaz gráfica `Install` ya que el subject indica que no se utilice ninguna. Cada vez que queramos confirmar algo presionaremos `Enter` y para movernos por las opciones utilizaremos las flechas.



2º Escogeremos el idioma que usaremos para la instalación y el predeterminado que se le quedará al sistema English .



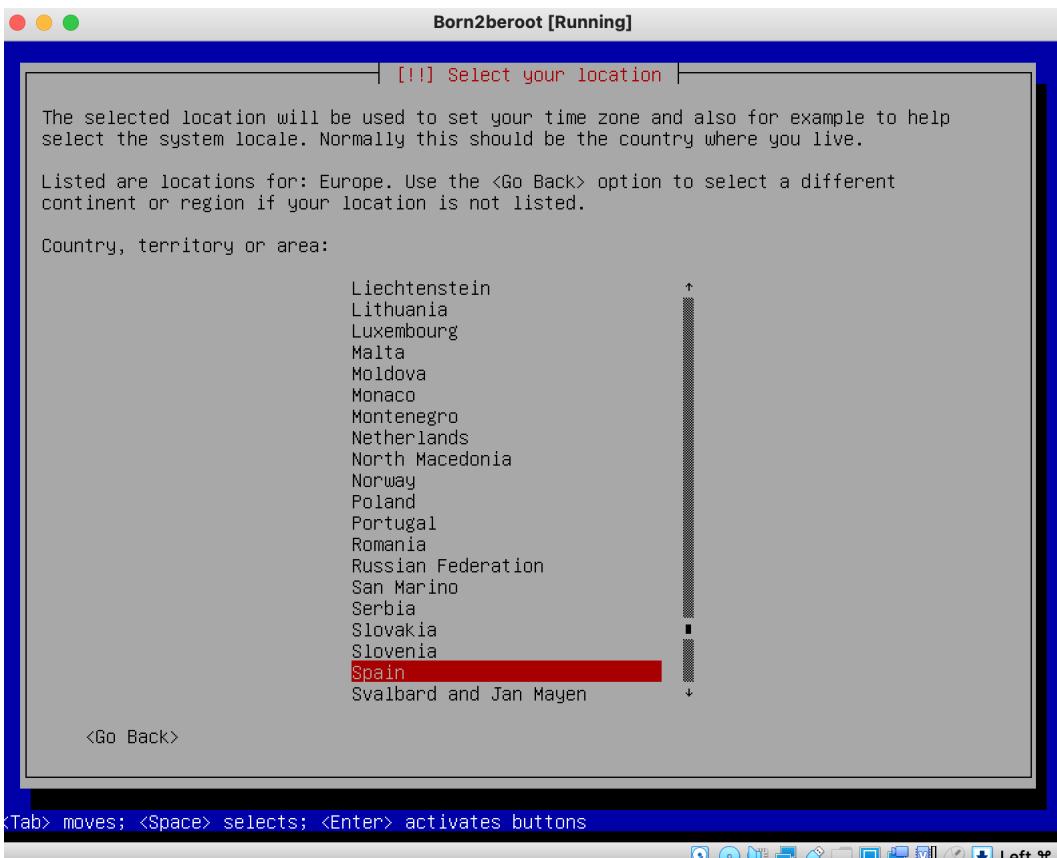
3º Introducimos nuestro País, territorio o zona. En mi caso pondré other .



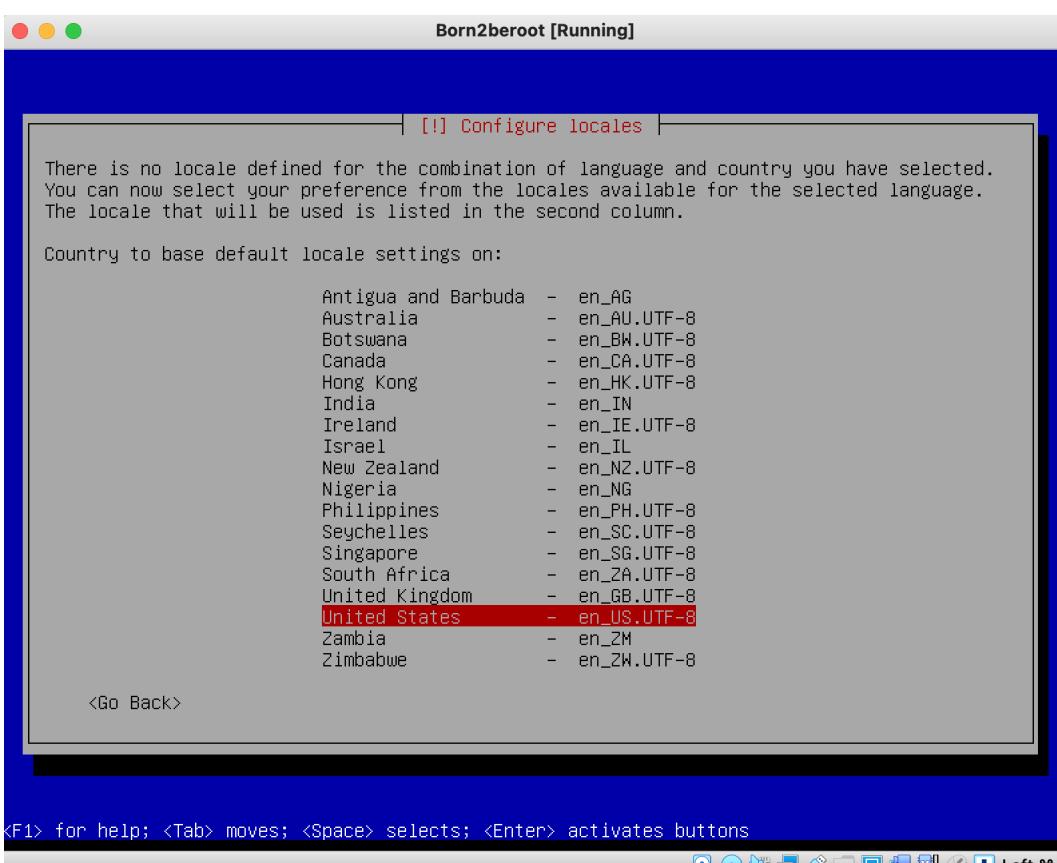
4º Como he seleccionado other debo indicar mi continente o region. En mi caso pongo Europe



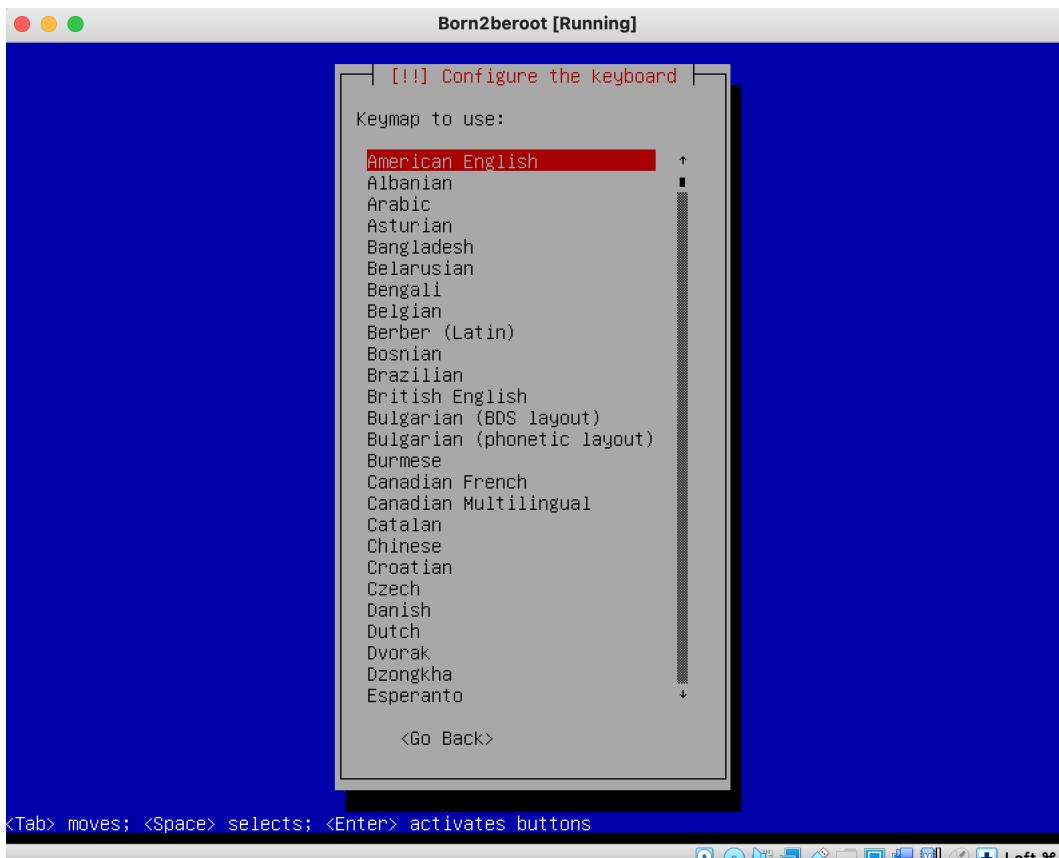
5º Seleccionamos el país. En mi caso Spain



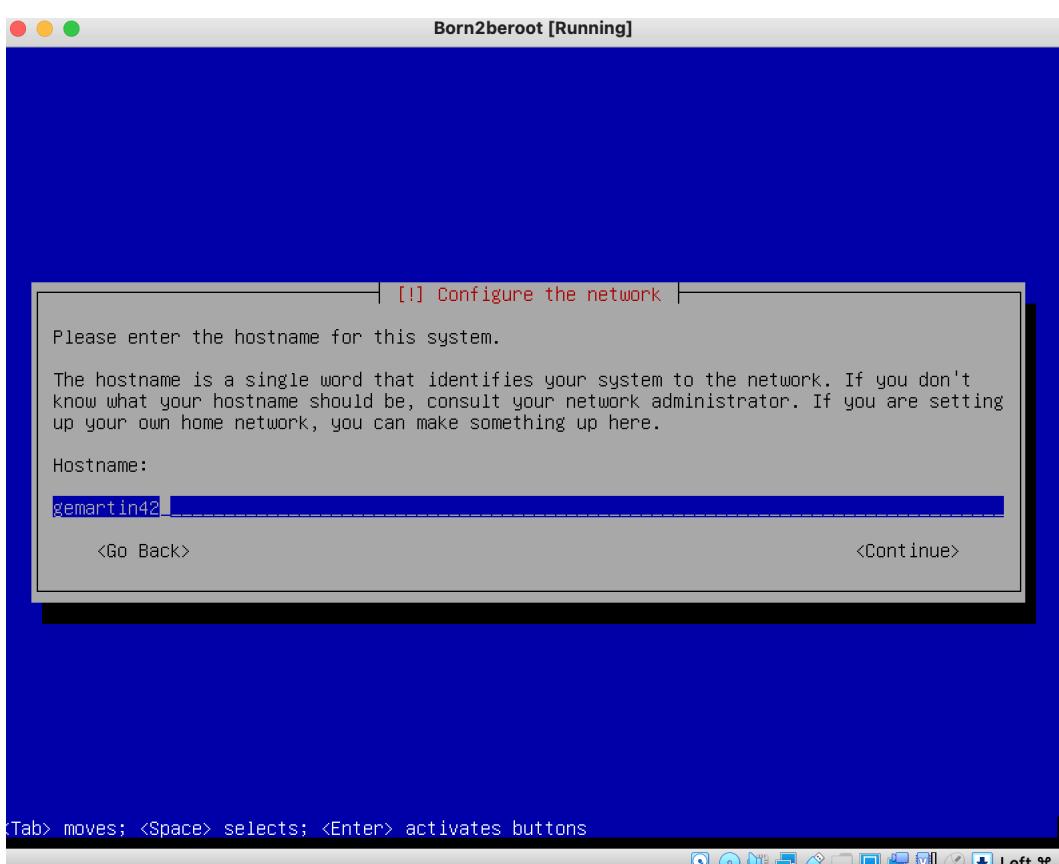
6º Seleccionamos United States.



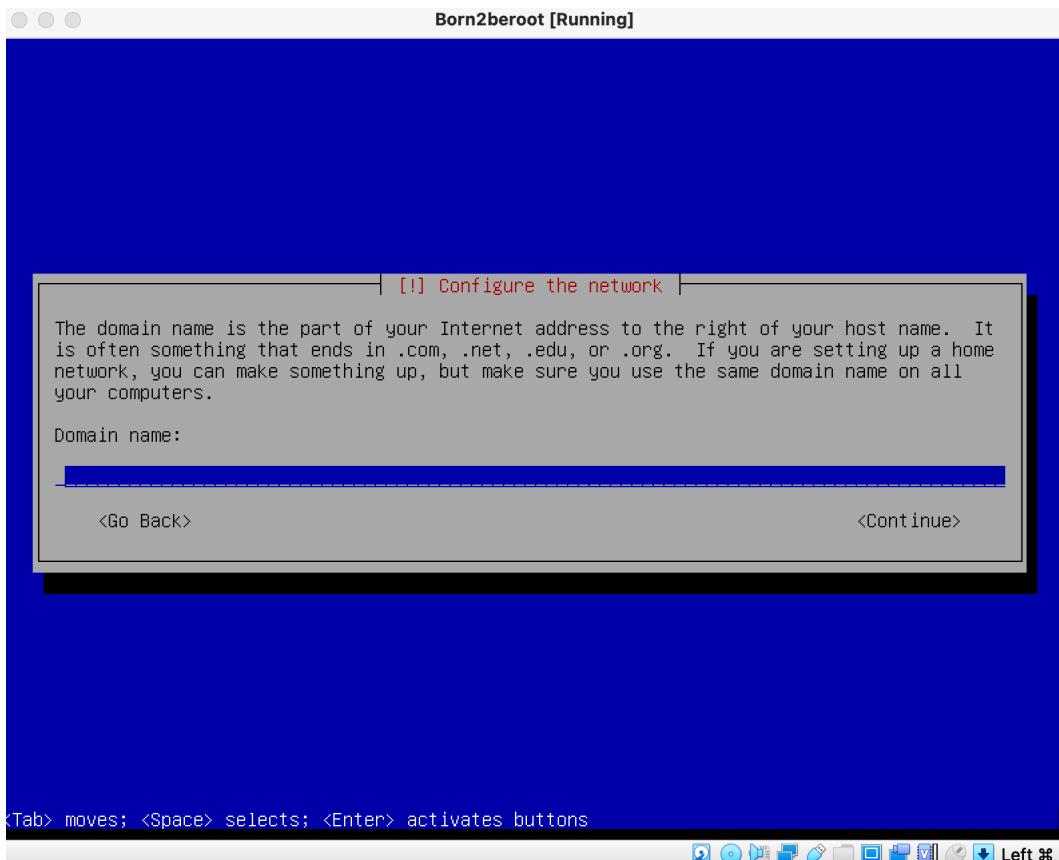
7º Importante seleccionar American English como configuración de teclado ya que si no tendremos las teclas mal enlazadas.



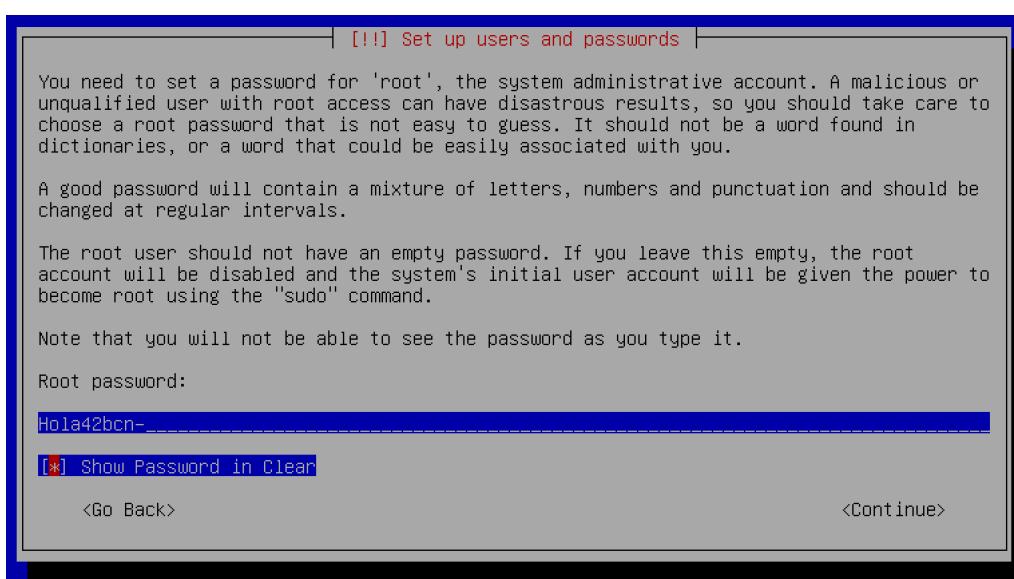
8º En este paso debemos elegir el `Host Name` de la máquina, el cual debe ser tu login seguido de 42.



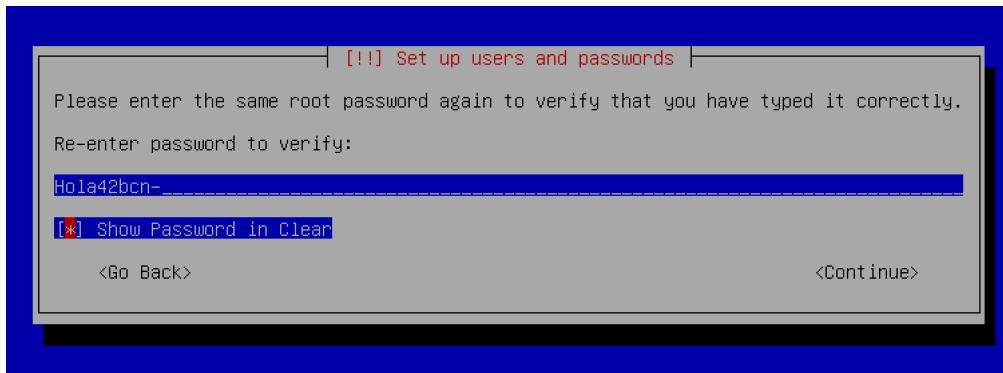
9º Este apartado lo dejaremos vacío ya que el subject no mencionada nada de `Domain name`.



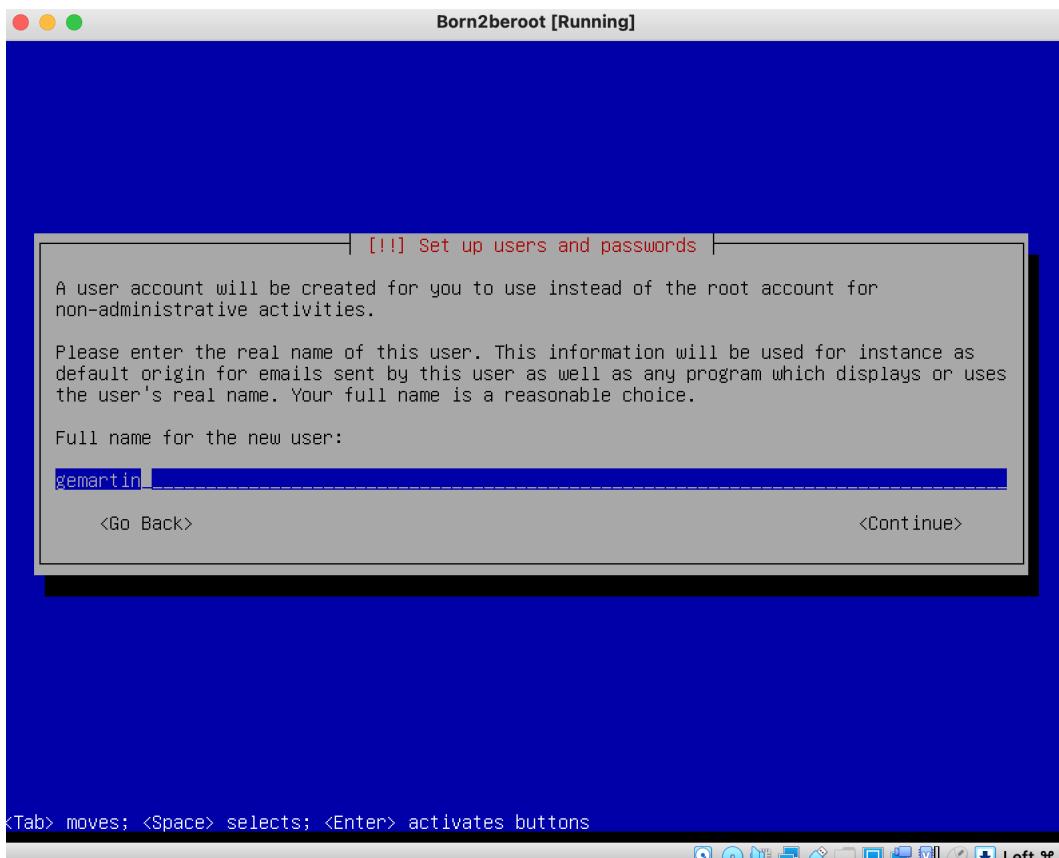
10 ° Debemos introducir una contraseña para la cuenta de administración del sistema. Importante apuntarla o hacer una foto ya que le daremos a la barra espaciadora y se mostrara la clave.



11 ° Repetimos el proceso de nuevo para comprobar que no la hayamos escrito mal.



12. Elegimos el nombre de nuestro nuevo usuario. Como indica el subject hay que crear un usuario adicional que no sea el root con nuestro lo por ese motivo llamaré `gemartin` a mi nuevo usuario.



Volvemos a poner el nombre de usuario.



[!] Set up users and passwords

Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

gemartin

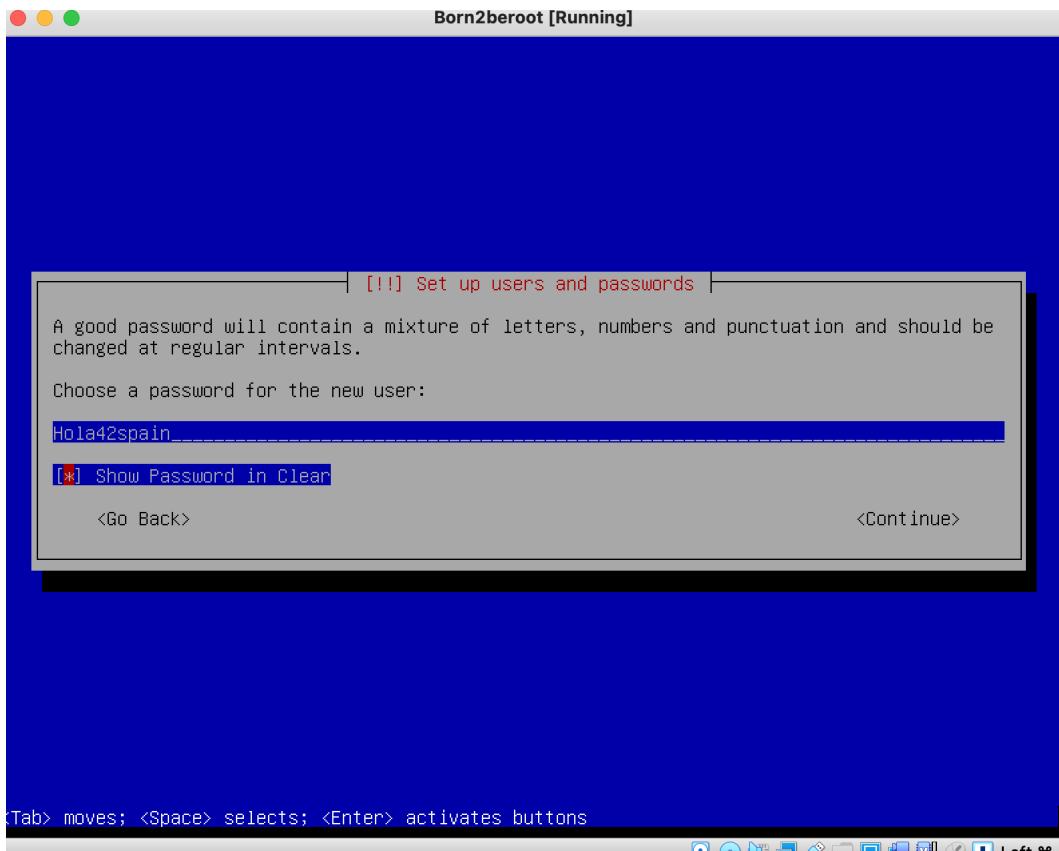
<Go Back>

<Continue>

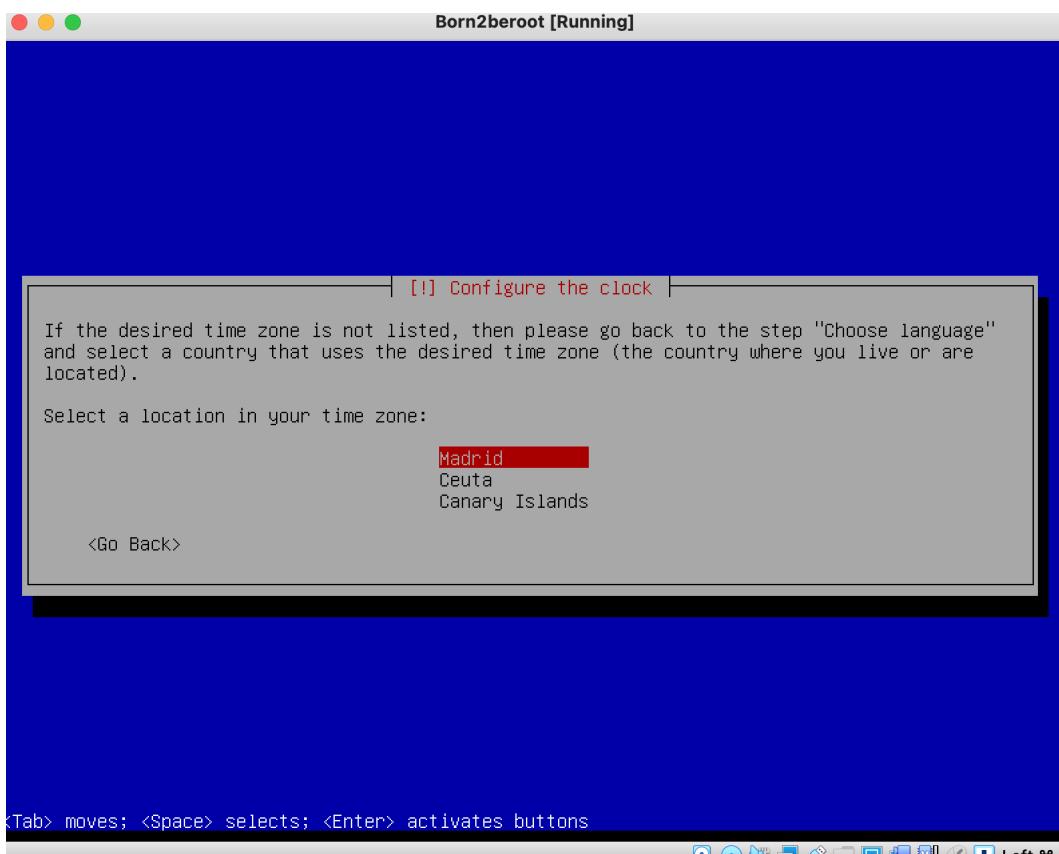
<Tab> moves; <Space> selects; <Enter> activates buttons



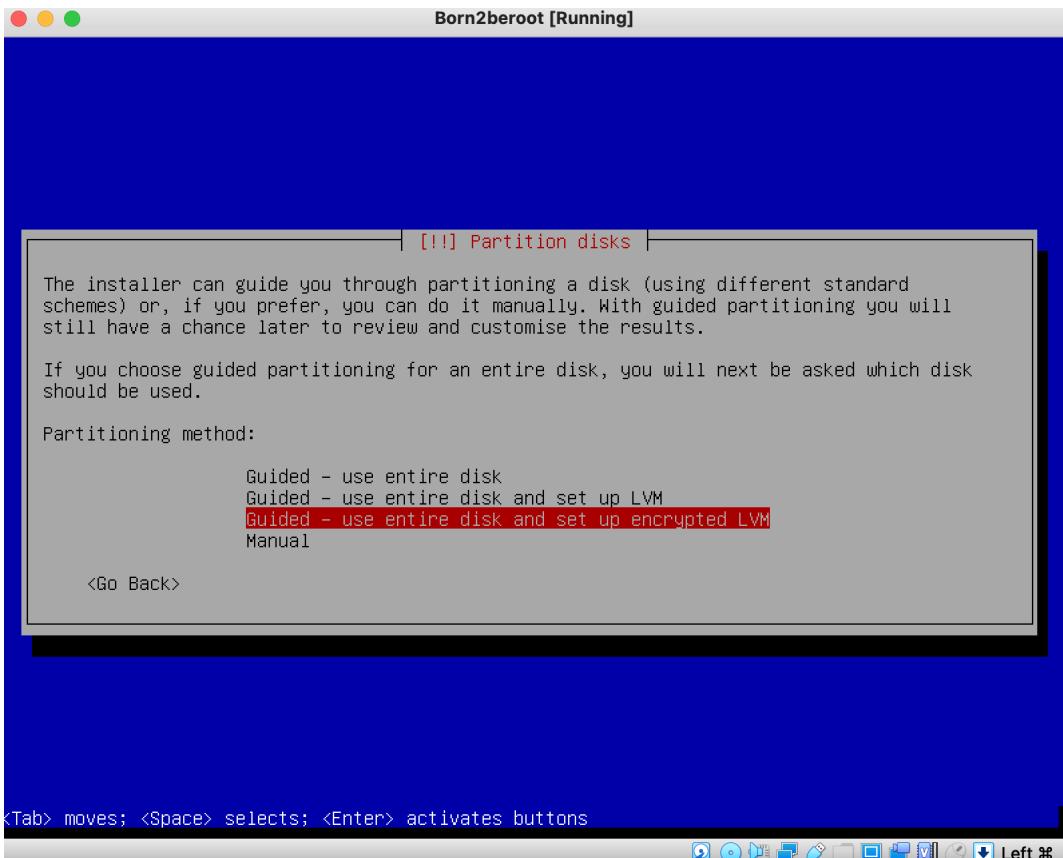
13º Ahora debemos introducir la contraseña de nuestro nuevo usuario. Como la anterior , repetiremos el proceso para comprobar que no la ha escrito mal y tambien es importante que la guardes porque le daremos uso más adelante.



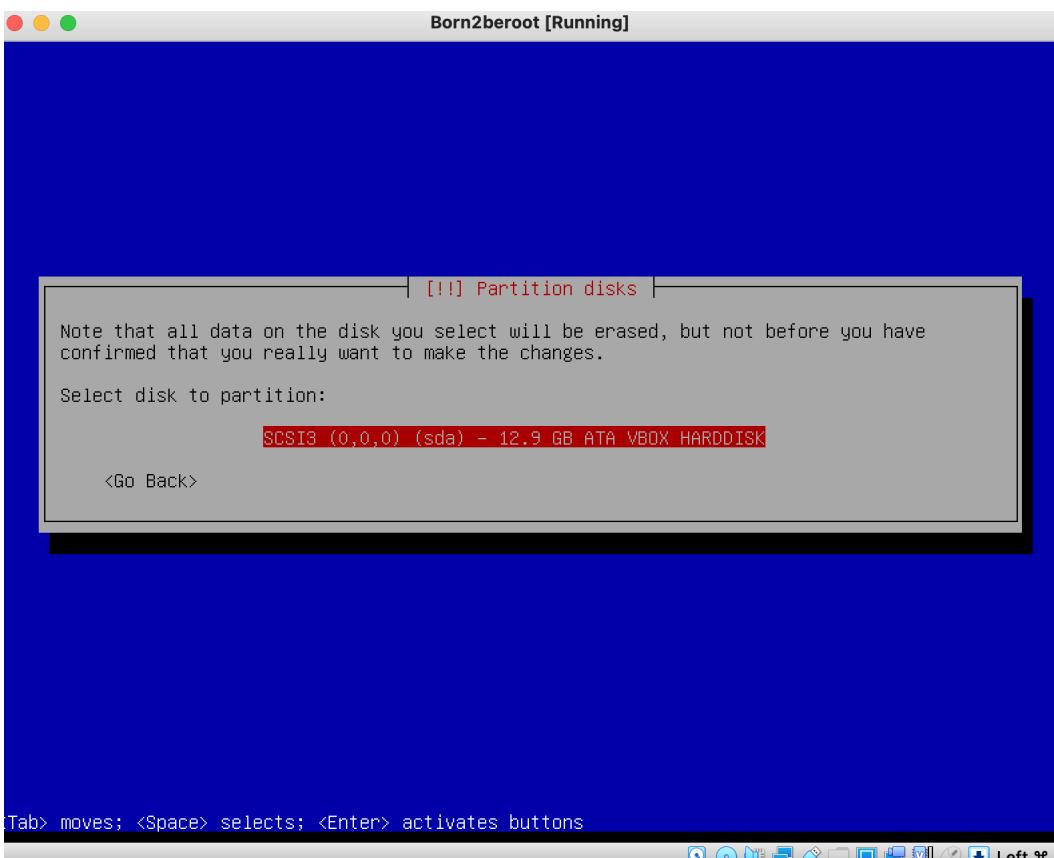
14 ° Seleccionamos la hora de nuestra ubicación.



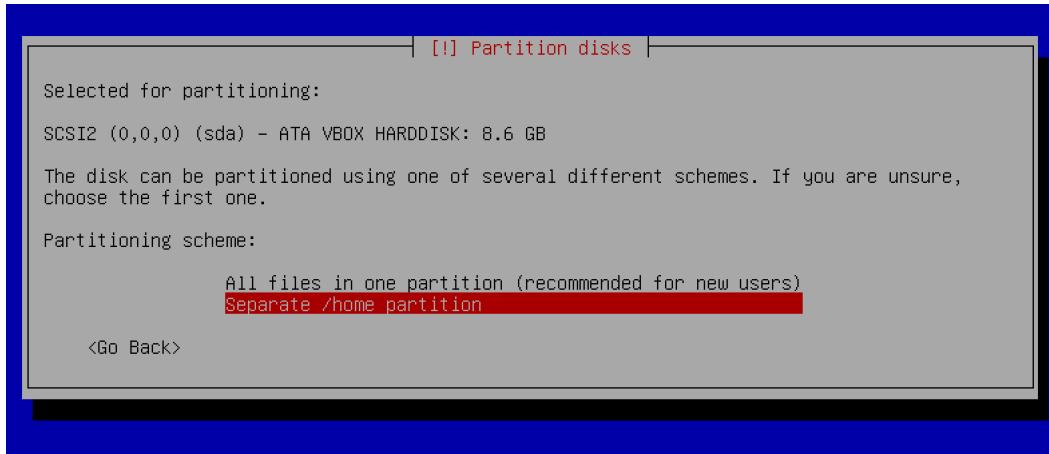
15 ° Esocgeremos la tercera opción Guided - use entire disk and set up encrypted LVM ya que el subject nos dice que deben ser particiones cifradas. △ ! Si quieres hacer el bonus deberás darle a Manual y [hacer click aquí](#) ! △



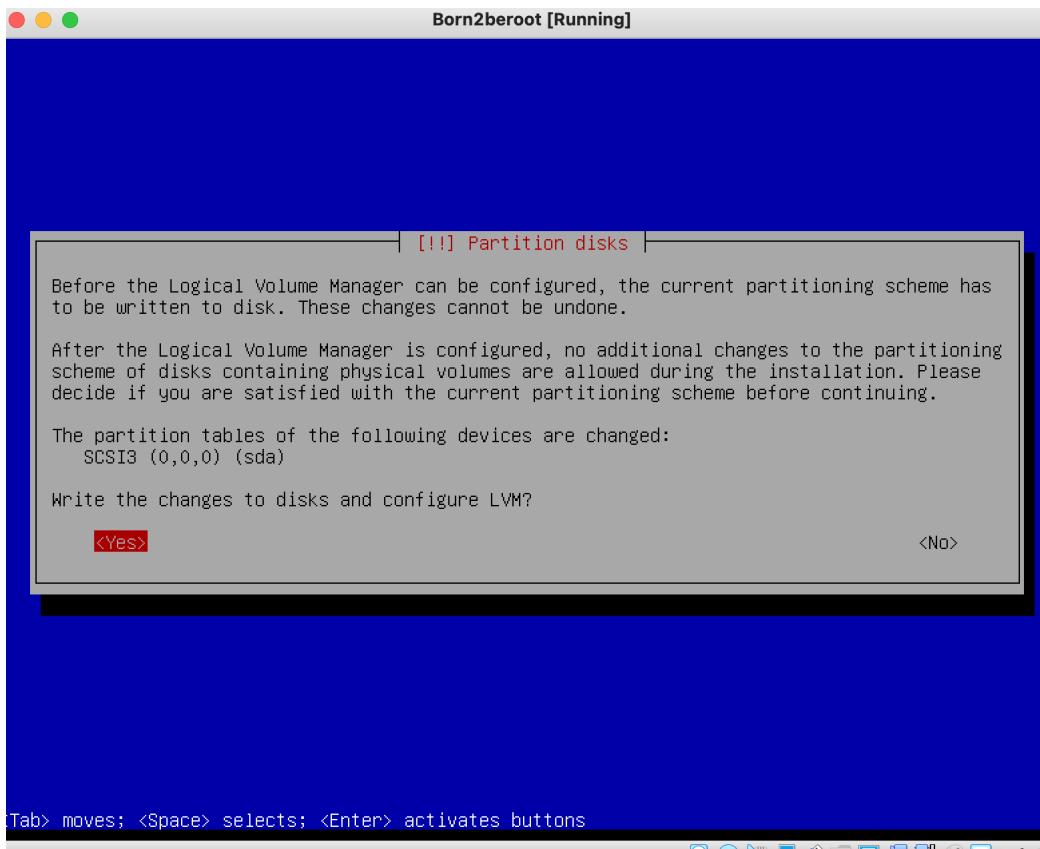
16 ° Seleccionamos el disco en el que queremos hacer el particionado (Solo debe haber un disco).



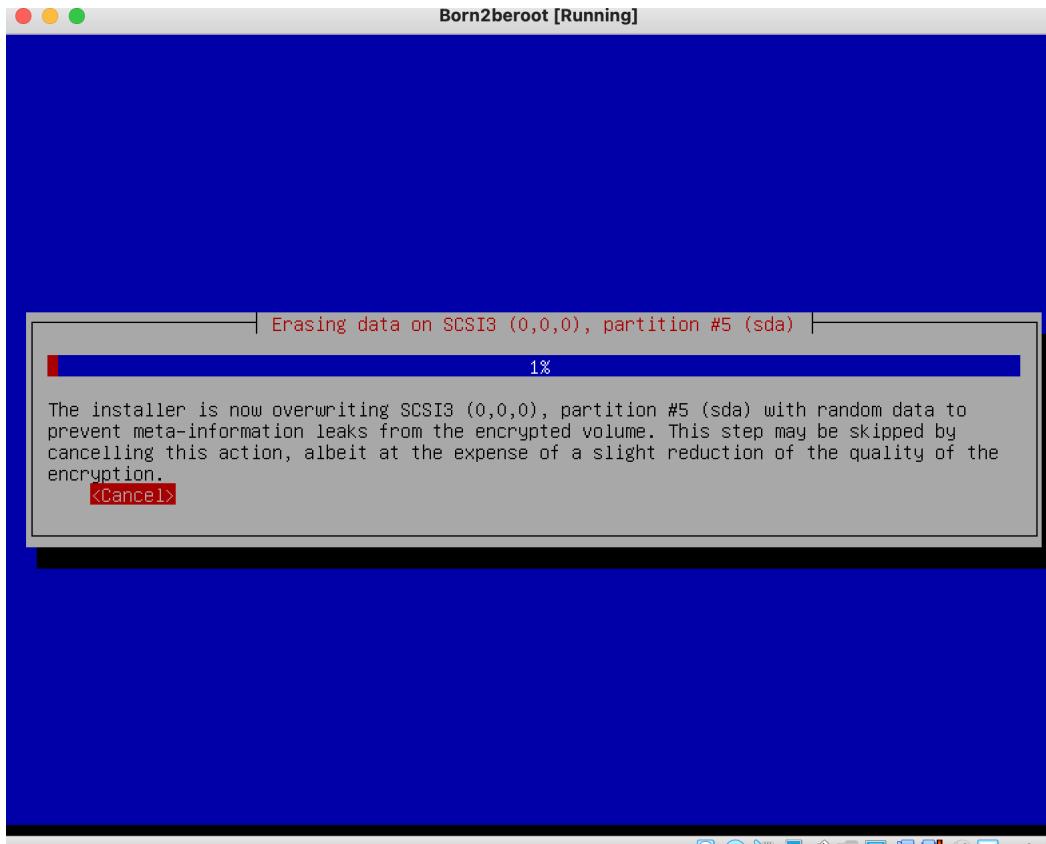
17 ° Una vez hayamos escogido el disco deberemos hacer el particionado tal y como nos piden. Para realizarlo adecuadamente debemos seleccionar la segunda opción `Separate /home partition`.



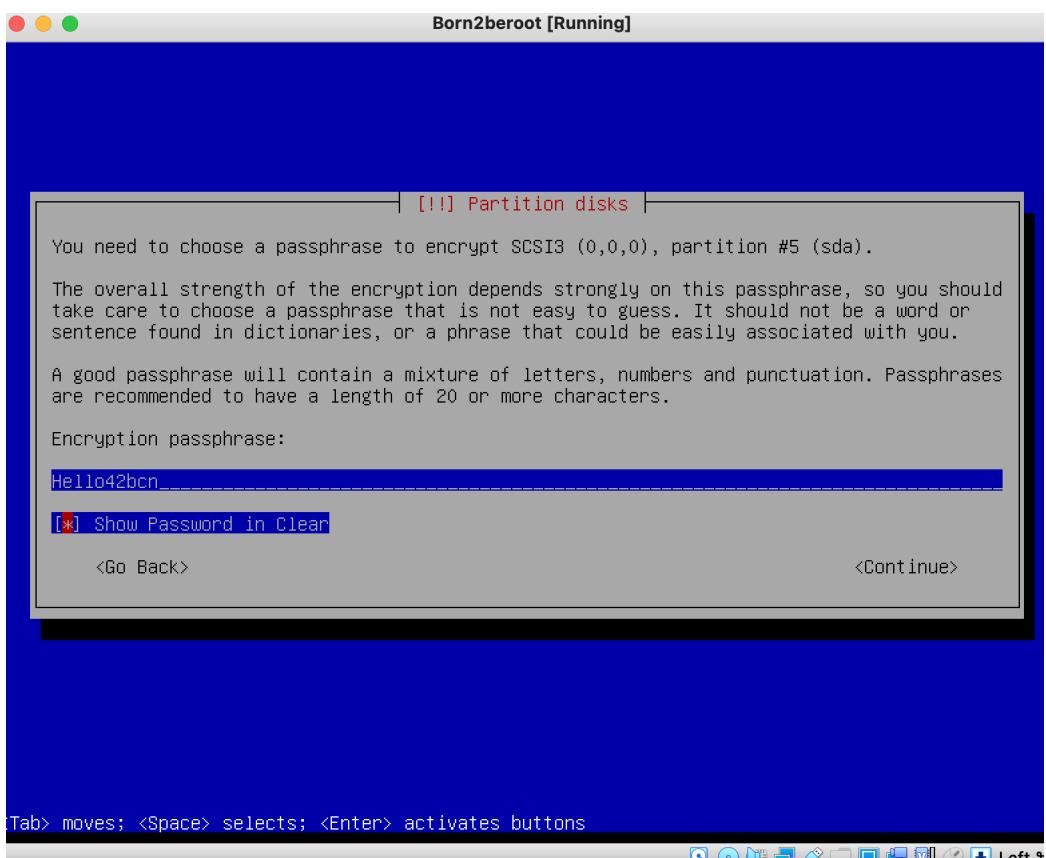
18 • Esocgemos la opción Yes para que asi se escriban los cambios en el disco y podamos configurar el gestor de volúmenes lógicos (LVM).



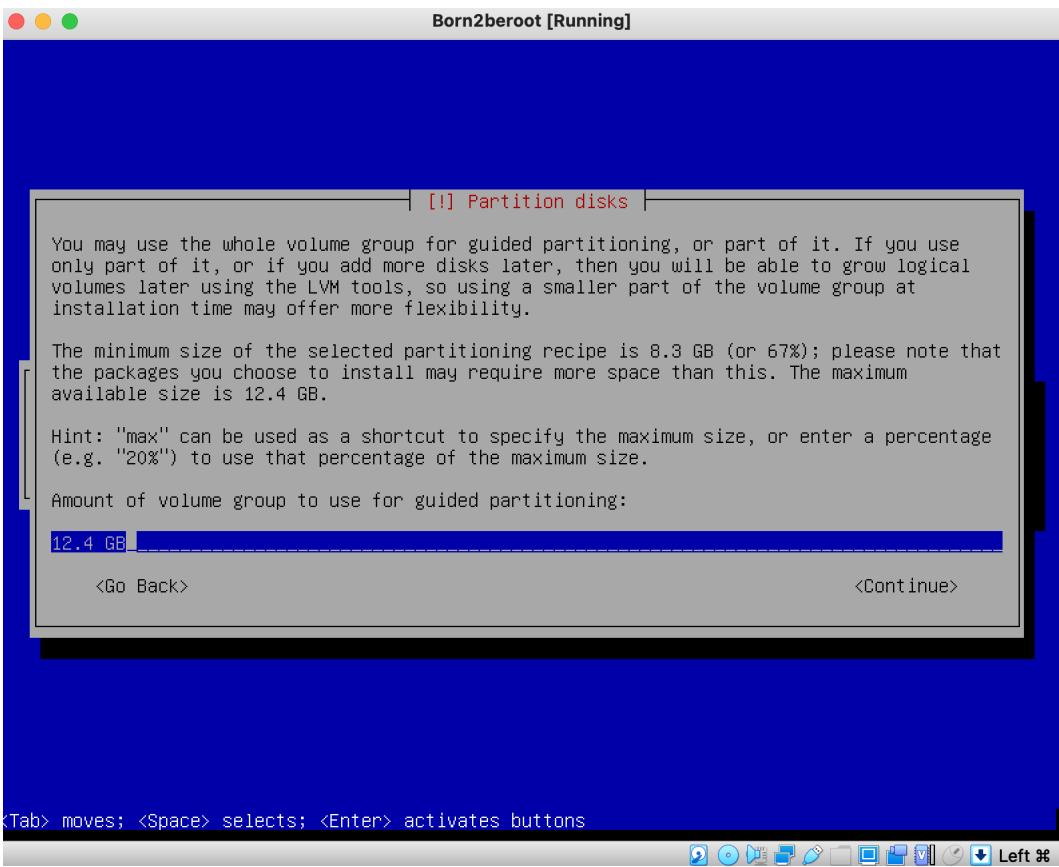
19 • Le damos a Cancel ya que el borrado de datos en el disco no es necesario.



20 ° De nuevo deberemos poner una contraseña, esta vez será la frase de encriptación. Como te he comentado previamente deberás repetir el proceso y la debes anotar ya que será importante en un futuro.



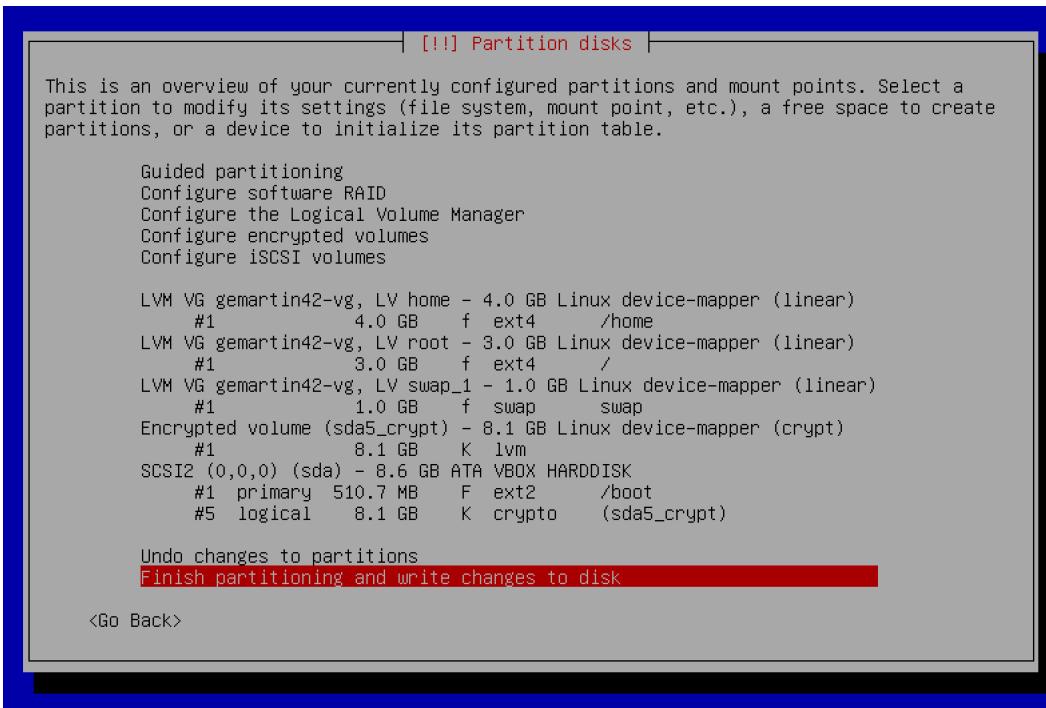
21 ° En este paso debemos introducir la cantidad de volumen que usaremos para la partición guiada. Debemos introducir `max` o el numero de tamaño maximo disponible en mi caso es `12.4 GB`.



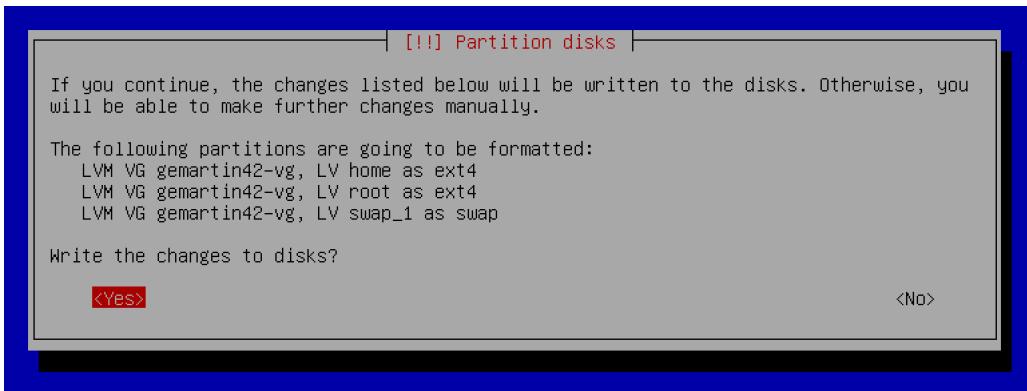
<Tab> moves; <Space> selects; <Enter> activates buttons

Left ⌘

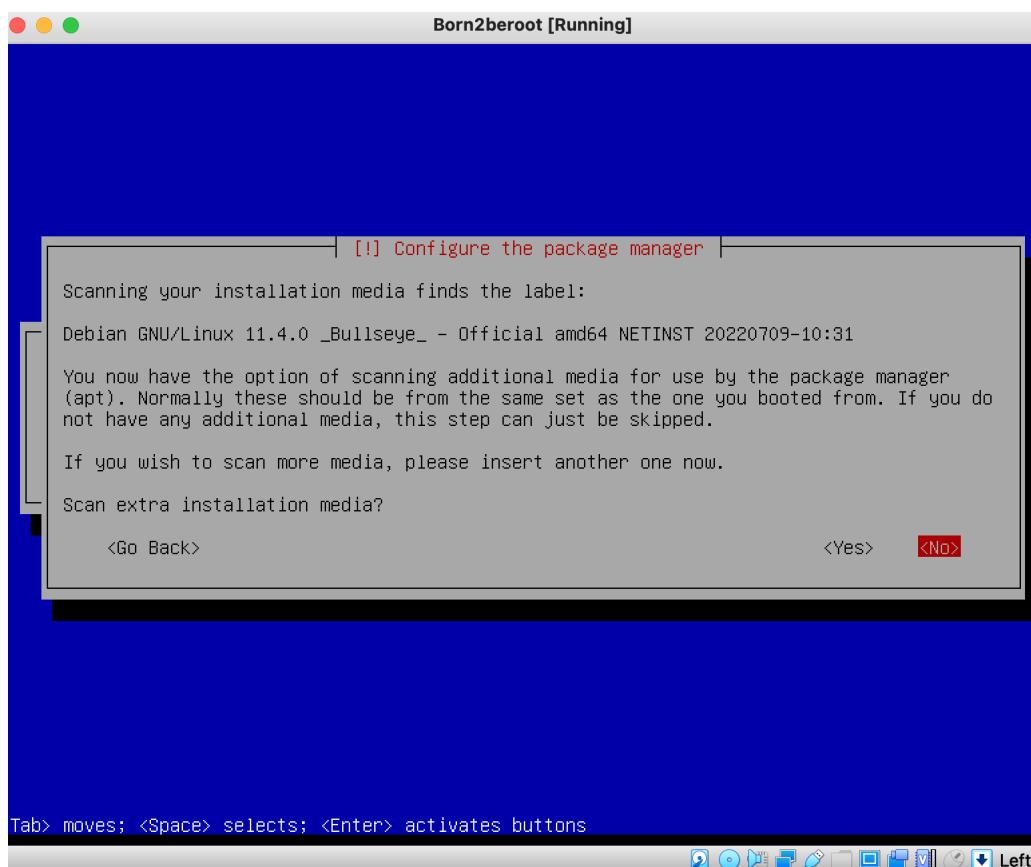
22. Para finalizar la partición y escribir los cambios en el disco le daremos a la opción `Finish partitioning and write changes to disk`.



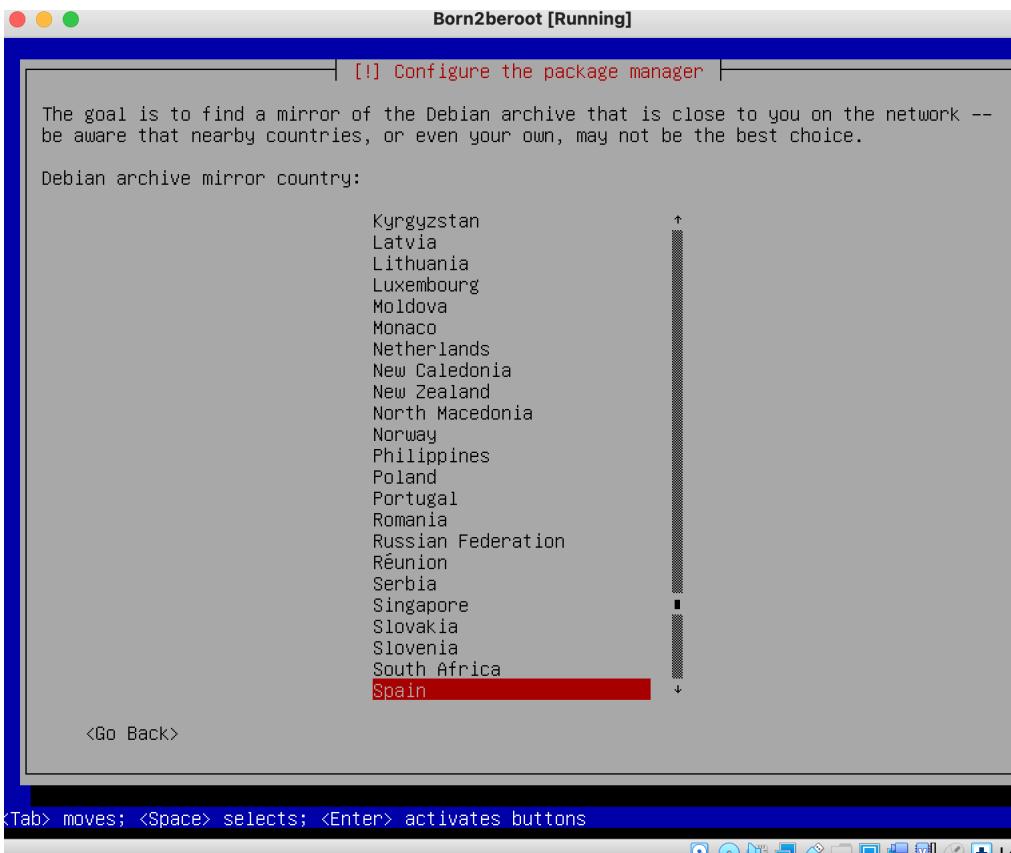
23. Seleccionamos la opción `Yes` para continuar y confirmar que no queremos hacer más cambios en el disco.



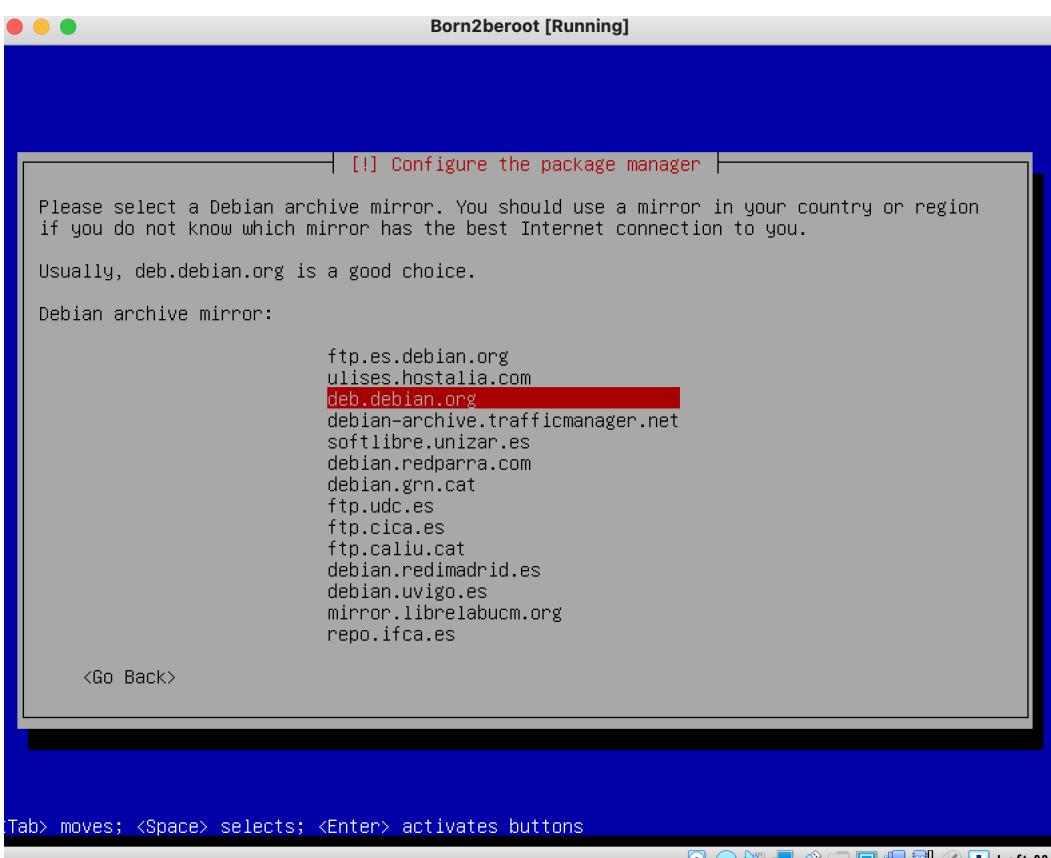
24 ° Seleccionamos la opción **No** ya que no necesitamos paquetes adicionales.



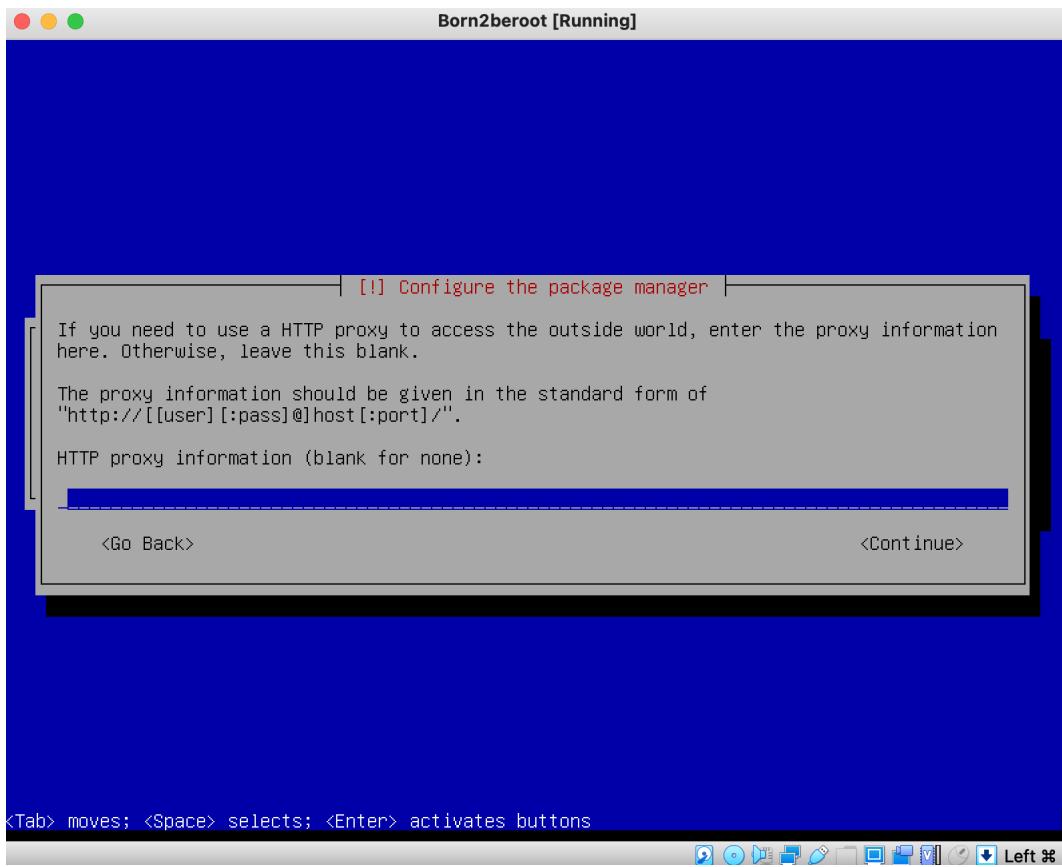
25 ° Escogemos nuestro País.



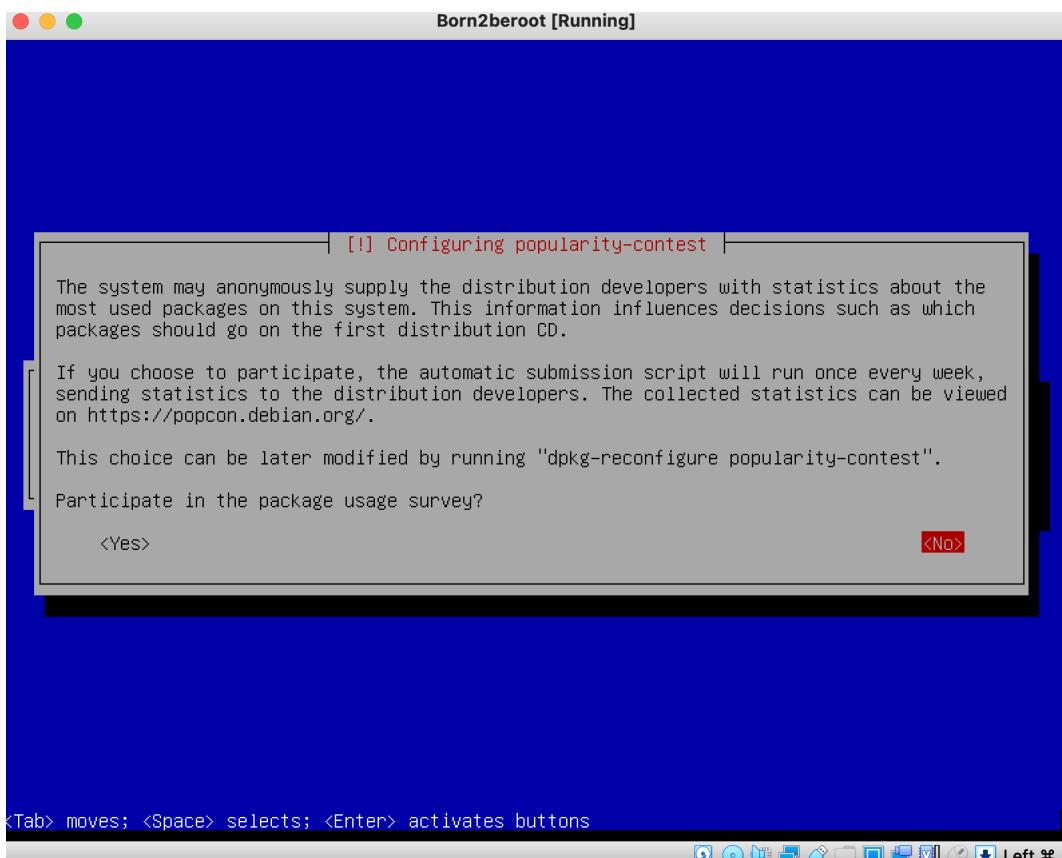
26º Escogemos `deb.debian.org` ya que es lo que recomienda debian.



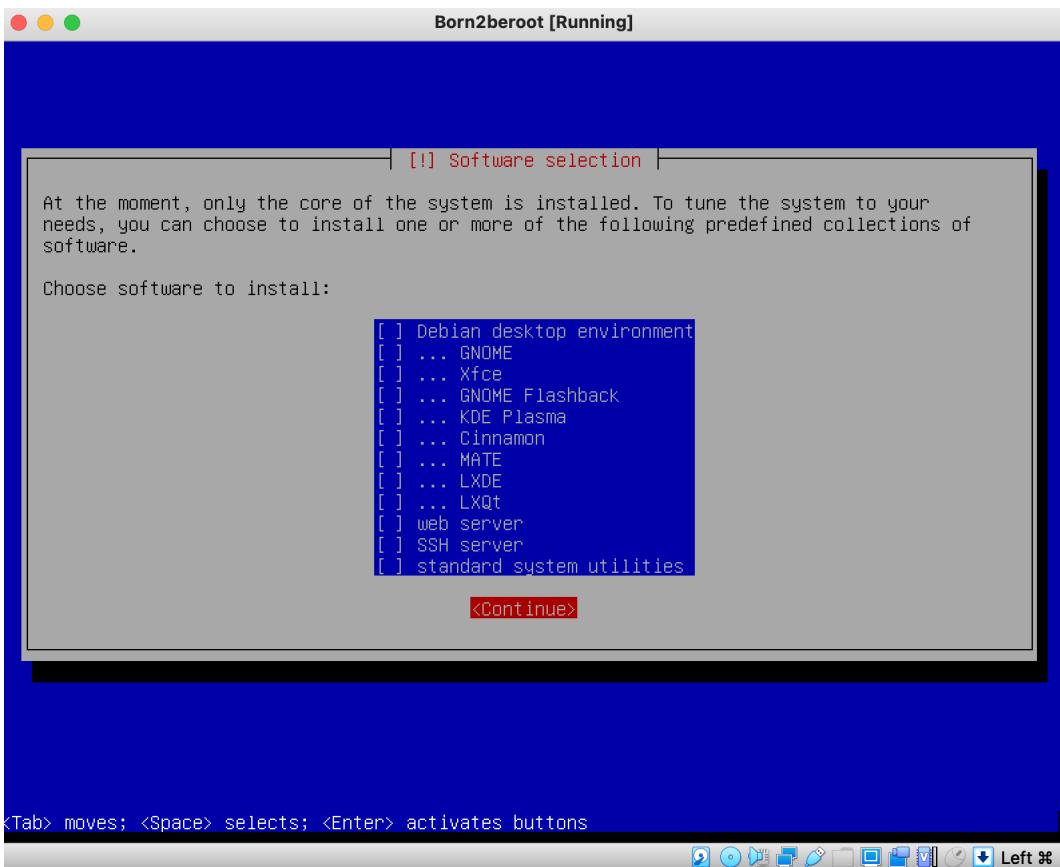
27º Esta opción la dejaremos vacía y le daremos `Continue`.



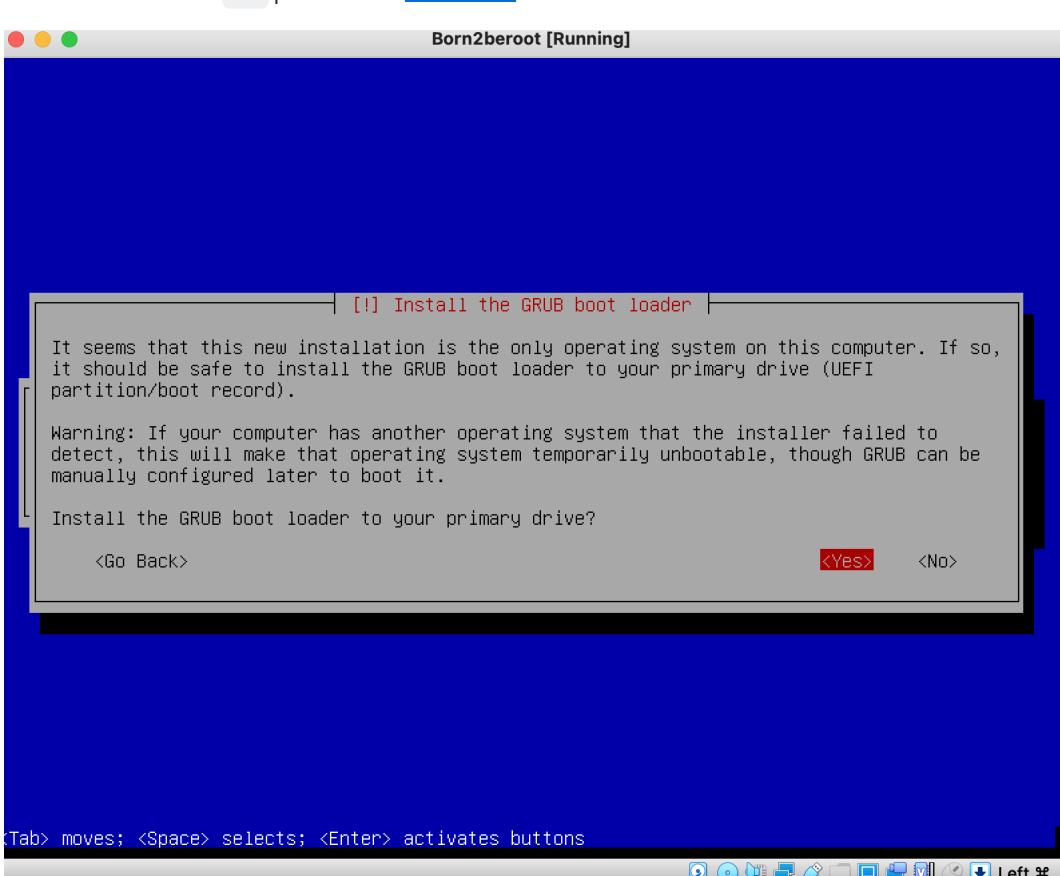
28 ° Seleccionamos la opción `No` ya que no queremos que los developers vean nuestras estadísticas aunque sean anónimas.



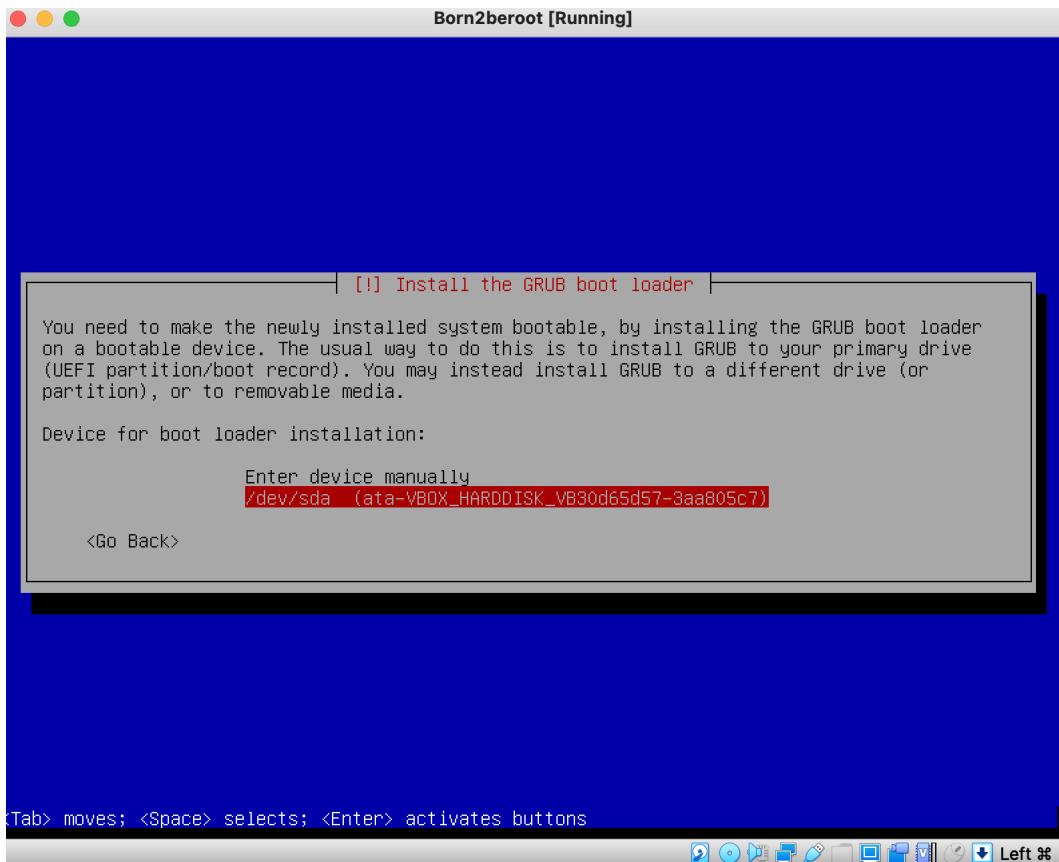
29 ° Quitaremos todas las opciones de software (con la barra espaciadora) y le daremos a `Continue`.



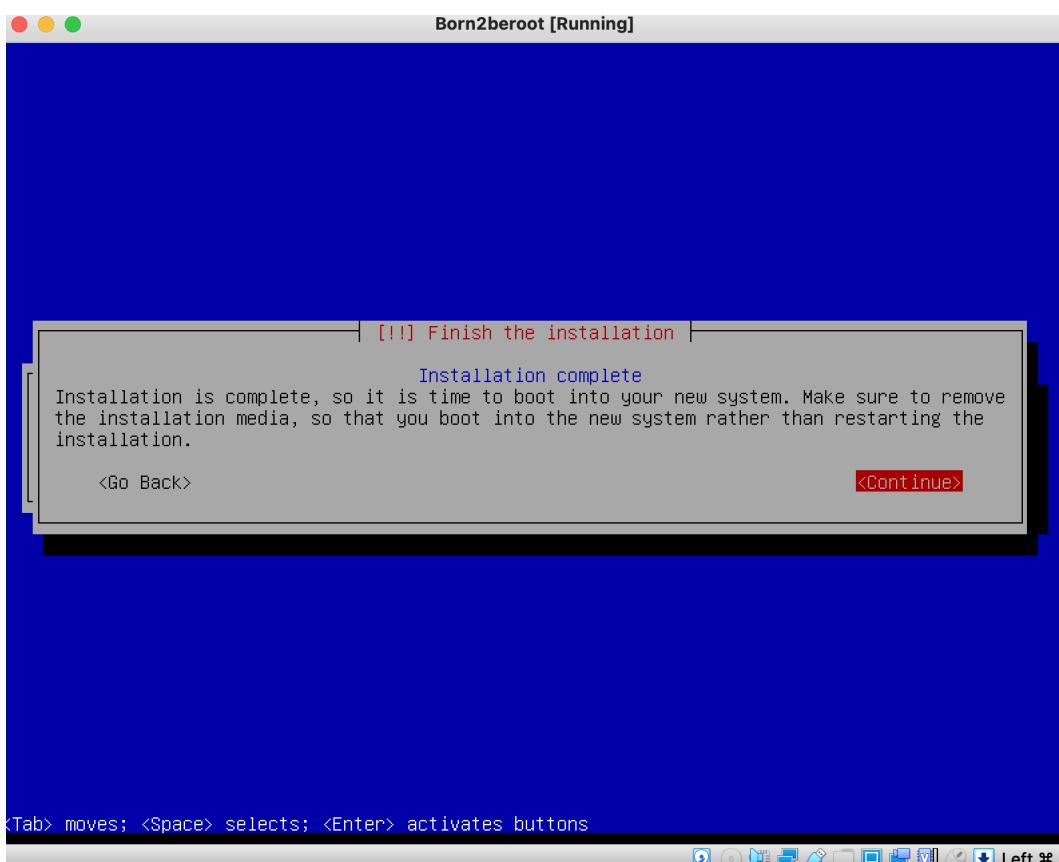
<Tab> moves; <Space> selects; <Enter> activates buttons



<Tab> moves; <Space> selects; <Enter> activates buttons



32º Le daremos a `Continue` para finalizar la instalación.



4 Configuración de la máquina virtual ☀

- Lo primero que debemos hacer es seleccionar `Debian GNU/Linux`.
- Debemos introducir la contraseña de encriptación que utilizamos previamente. En mi caso es `Hello42bcn`.

```
Born2beroot [Running]
Volume group "gemartin42-vg" not found
Cannot process volume group gemartin42-vg
Volume group "gemartin42-vg" not found
Cannot process volume group gemartin42-vg
Please unlock disk sda5_crypt:
```

► Debemos introducir el usuario y contraseña que hemos creado. En mi caso el usuario es `gemartin` y la contraseña `Hola42spain`.

```
Born2beroot [Running]
Debian GNU/Linux 11 gemartin42 tty1
gemartin42 login: gemartin
Password:
Linux gemartin42 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
gemartin@gemartin42:~$ _
```

› Ya tenemos todo listo para empezar a configurar nuestra máquina virtual Debian !

› 4.1 - Instalación de sudo y configuración de usuarios y grupos ↗

1 ° Para la instalación de sudo primero debemos estar en el usuario root, para ello pondremos `su` en el terminal y introduciremos la contraseña mi caso es `Hola42bcn`. Una vez hemos accedido al usuario root debemos poner el comando `apt install sudo` para así instalar los paquetes necesarios.

```
gemartin@gemartin42:~$ su
Password:
root@gemartin42:/home/gemartin# apt install sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,059 kB of archives.
After this operation, 4,699 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 sudo amd64 1.9.5p2-3 [1,059 kB]
Fetched 1,059 kB in 5s (197 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 18614 files and directories currently installed.)
Preparing to unpack .../sudo_1.9.5p2-3_amd64.deb ...
Unpacking sudo (1.9.5p2-3) ...
Setting up sudo (1.9.5p2-3) ...
root@gemartin42:/home/gemartin#
```

2º Debemos reiniciar la máquina para que se apliquen los cambios. Para ello haremos uso del comando `sudo reboot` y esperaremos a que se reinicie.

```
root@gemartin42:~# sudo reboot
```

3º Una vez reiniciado debemos volver a introducir las contraseñas de cifrado y del usuario. Para verificar que hayamos instalado `sudo` correctamente entraremos de nuevo en el usuario root y pondremos el comando `sudo -V`, este comando además de mostrarnos la versión de sudo también mostrará los argumentos pasados para configurar cuando se creó sudo y los plugins que pueden mostrar información más detallada. (Opcionalmente puesto que el output del comando es muy largo si deseamos verlo completamente debemos redireccionar la salida del mismo a un fichero `sudo > file.txt` y luego editar el fichero `nano file.txt`. O poner `| more` después del comando.

```
Born2beroot [Running]
COLORS
Locale to use while parsing sudoers: C
Compress I/O logs using zlib
Directory in which to store input/output logs: /var/log/sudo-io
File in which to store the input/output log: %{seq}
Add an entry to the utmp/utmpx file when allocating a pty
PAM service name to use: sudo
PAM service name to use for login shells: sudo
Attempt to establish PAM credentials for the target user
Create a new PAM session for the command to run in
Perform PAM account validation management
Enable sudoers netgroup support
Check parent directories for writability when editing files with sudoedit
Allow commands to be run even if sudo cannot write to the audit log
Allow commands to be run even if sudo cannot write to the log file
Log entries larger than this value will be split into multiple syslog messages: 960
File mode to use for the I/O log files: 0600
Execute commands by file descriptor instead of by path: digest_only
Type of authentication timestamp record: tty
Ignore case when matching user names
Ignore case when matching group names
Log when a command is allowed by sudoers
Log when a command is denied by sudoers
Sudo log server timeout in seconds: 30
Enable SO_KEEPALIVE socket option on the socket connected to the logserver
Verify that the log server's certificate is valid
Set the pam remote user to the user running sudo
The format of logs to produce: sudo
Enable SELinux RBAC support

Local IP address and netmask pairs:
 10.0.2.15/255.255.255.0
 fe80::a00:27ff:fe93:3558/ffff:ffff:ffff:ffff::

Sudoers I/O plugin version 1.9.5p2
Sudoers audit plugin version 1.9.5p2
root@gemartin42:~#
```

4º Siguiendo en el usuario root crearemos un usuario con nuestro login con el comando `sudo adduser login` como nosotros ya hemos creado el usuario en la instalación nos debe aparecer que el usuario ya existe.

```
Born2beroot [Running]
root@gemartin42:~# sudo adduser gemartin
adduser: The user 'gemartin' already exists.
root@gemartin42:~#
```

5º Ahora deberemos crear un nuevo grupo llamado `user42`. Para crearlo debemos hacer `sudo addgroup user42`.

```
gemartin@gemartin42:~$ sudo addgroup user42
Adding group `user42' (GID 1001) ...
Done.
gemartin@gemartin42:~$
```

💡 **Que es GID ?** Es el identificador de grupo, es una abreviatura de Group ID.

💡 **Se ha creado correctamente el grupo?** Lo cierto es que si ya que no ha habido ningún mensaje de error, aún así podemos comprobar si se ha creado con el comando `getent group nombre_grupo` o también podemos hacer `cat /etc/group` y podremos ver todos los grupos y los usuarios que hay dentro de ellos.

6º Con el comando `sudo adduser user group` incluiremos al usuario en el grupo. Debemos incluir al usuario en los grupos `sudo` y `user42`.

```
gemartin@gemartin42:~$ sudo adduser gemartin user42
Adding user `gemartin' to group `user42' ...
Adding user gemartin to group user42
Done.
gemartin@gemartin42:~$
```

```
gemartin@gemartin42:~$ sudo adduser gemartin sudo
Adding user `gemartin' to group `sudo' ...
Adding user gemartin to group sudo
Done.
gemartin@gemartin42:~$
```

7º Una vez los hayamos introducido para checkear que todo se haya hecho correctamente podemos ejecutar el comando `getent group nombre_grupo` o tambien podemos editar el fichero `/etc/group` `nano /etc/group` y en los grupos `sudo` y `login42` deberá aparecer nuestro usu

```
gemartin@gemartin42:~$ getent group sudo
sudo:x:27:gemartin
gemartin@gemartin42:~$
```

```
sudo:x:27:gemartin
```

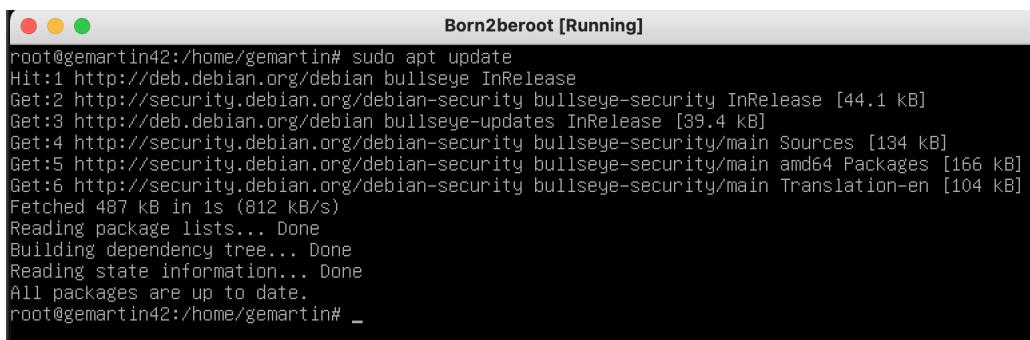
```
gemartin@gemartin42:~$ getent group user42
user42:x:1001:gemartin
gemartin@gemartin42:~$
```

```
user42:x:1001:gemartin
```

4.2 - Instalación y configuración SSH

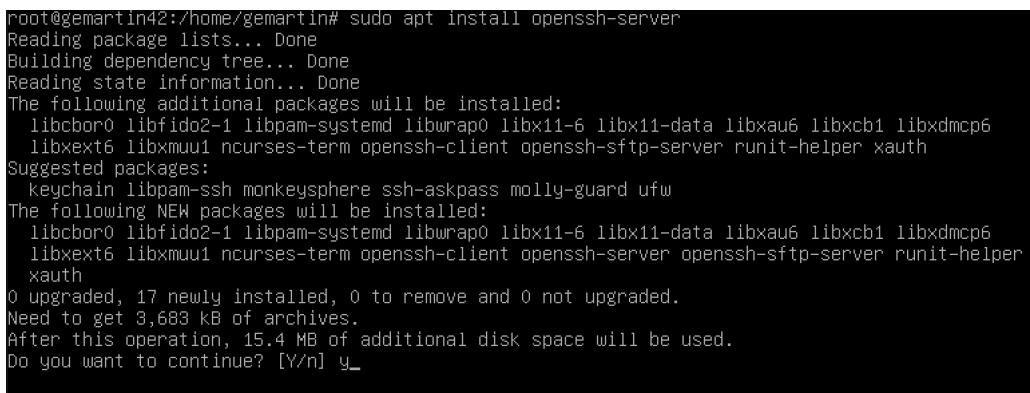
• **Que es SSH ?** Es el nombre de un protocolo y del programa que lo implementa cuya principal función es el acceso remoto a un servidor por de un canal seguro en el que toda la información está cifrada.

1 ° Lo primero que haremos será hacer `sudo apt update` para actualizar los repositorios que definimos en el archivo `/etc/apt/sources.list`



```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Get:2 http://security.debian.org/debian-security bullseye-security InRelease [44.1 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [39.4 kB]
Get:4 http://security.debian.org/debian-security bullseye-security/main Sources [134 kB]
Get:5 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [166 kB]
Get:6 http://security.debian.org/debian-security bullseye-security/main Translation-en [104 kB]
Fetched 487 kB in 1s (812 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@gemartin42:/home/gemartin# _
```

2 ° Acto seguido instalaremos la herramienta principal de conectividad para el inicio de sesión remoto con el protocolo SSH, esta herramienta es OpenSSH. Para instalarla debemos introducir el comando `sudo apt install openssh-server`. En el mensaje de confirmación ponemos `Y`, acto seguido esperaremos a que termine la instalación.



```
root@gemartin42:/home/gemartin# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcbor0 libfido2-1 libpam-systemd liburap0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6
  libxext6 libxmuu1 ncurses-term openssh-client openssh-sftp-server runit-helper xauth
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard ufw
The following NEW packages will be installed:
  libcbor0 libfido2-1 libpam-systemd liburap0 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6
  libxext6 libxmuu1 ncurses-term openssh-client openssh-server openssh-sftp-server runit-helper
  xauth
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 3,683 kB of archives.
After this operation, 15.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y_
```

Para comprobar que se haya instalado correctamente haremos `sudo service ssh status` y nos debe aparecer active.



```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2022-07-13 22:49:28 CEST; 37min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 1250 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1251 (sshd)
      Tasks: 1 (limit: 1128)
        Memory: 1.0M
         CPU: 21ms
        CGroup: /system.slice/ssh.service
                 └─1251 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jul 13 22:49:28 gemartin42 systemd[1]: Starting OpenBSD Secure Shell server...
Jul 13 22:49:28 gemartin42 sshd[1251]: Server listening on 0.0.0.0 port 22.
Jul 13 22:49:28 gemartin42 sshd[1251]: Server listening on :: port 22.
Jul 13 22:49:28 gemartin42 systemd[1]: Started OpenBSD Secure Shell server.
root@gemartin42:/home/gemartin#
```

3 ° Una vez terminada la instalación se han creado algunos ficheros que debemos configurar. Para ello utilizaremos [Nano](#) o si tu lo prefieres otro editor de texto. El primer fichero que editaremos será `/etc/ssh/sshd_config`. Si no estas desde el usuario root no tendrás permisos de escritura para ello haremos `su` y ponemos la contraseña para entrar al usuario root o si no quieres entrar en el usuario root ponemos `sudo` al principio comando `sudo nano /etc/ssh/sshd_config`.

```
● ● ● Born2beroot [Running]
root@gemartin42:/home/gemartin# nano /etc/ssh/sshd_config_
```

4º Los `#` al comienzo de una línea significan que esta comentada, las líneas que vayamos a modificar deberás quitarle el comentario. Una vez estemos editando el fichero deberemos modificar las siguientes líneas:

► #Port 22 -> Port 4242

```
● ● ● Born2beroot [Running]
GNU nano 5.4 /etc/ssh/sshd_config *
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 4242 ←
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m

[G] Help      [W] Write Out  [F] Where Is  [C] Cut      [E] Execute  [U] Location  [U-U] Undo
[Q] Exit      [R] Read File  [R] Replace  [P] Paste    [J] Justify  [G] Go To Line  [R-R] Redo
[Left ⌘]
```

► #PermitRootLogin prohibit-password -> PermitRootLogin no

```
GNU nano 5.4
Born2beroot [Running]
/etc/ssh/sshd_config *

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no ←
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication

^G Help      ^O Write Out  ^W Where Is  ^K Cut        ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^Y Replace   ^U Paste     ^J Justify    ^L Go To Line M-E Redo
                                         ^D Delete    ^I Insert    ^H Home     ^F End      Left %
```

Una vez hayamos modificado esas líneas debemos guardar los cambios realizados sobre el fichero y dejar de editarlo.

5º Ahora debemos editar el fichero /etc/ssh/sshd_config .

```
Born2beroot [Running]
root@gemartin42:/home/gemartin# nano /etc/ssh/sshd_config_
```

Editaremos la siguiente línea:

► #Port 22 -> Port 4242

Born2beroot [Running] /etc/ssh/ssh_config *

```

# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
Port 4242
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1G 1h
# UserKnownHostsFile ~/.ssh/known_hosts.d/%k
SendEnv LANG LC_*
HashKnownHosts yes
GSSAPIAuthentication yes

^G Help      ^Q Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location  M-U Undo
^X Exit      ^R Read File  ^Y Replace   ^U Paste    ^J Justify  ^A Go To Line M-E Redo

```

6º Por último debemos reiniciar el servicio ssh para que así se actualicen las modificaciones que acabamos de realizar. Para ello debemos escribir el comando `sudo service ssh restart` y una vez reseteado miraremos el estado actual con `sudo service ssh status` y para confirmar que se han realizado los cambios en la escucha del servidor debe aparecer el Puerto 4242.

Born2beroot [Running]

```

root@gemartin42:/home/gemartin# sudo service ssh restart
root@gemartin42:/home/gemartin# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2022-07-13 23:30:50 CEST; 16s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 1307 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1310 (sshd)
   Tasks: 1 (limit: 1128)
  Memory: 1.0M
     CPU: 28ms
    CGroup: /system.slice/ssh.service
            └─1310 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jul 13 23:30:50 gemartin42 systemd[1]: Starting OpenBSD Secure Shell server...
Jul 13 23:30:50 gemartin42 sshd[1310]: Server listening on 0.0.0.0 port 4242.
Jul 13 23:30:50 gemartin42 sshd[1310]: Server listening on :: port 4242.
Jul 13 23:30:50 gemartin42 systemd[1]: Started OpenBSD Secure Shell server.
root@gemartin42:/home/gemartin# 

```

4-3 Instalació y configuración de UFW 🔥

• **Que es UFW?** Es un [firewall](#) el cual utiliza la línea de comandos para configurar las [iptables](#) usando un pequeño número de comandos simples.

1º Lo primero que debemos hacer es instalar UFW, para ello haremos uso del comando `sudo apt install ufw` acto seguido escribiremos una | para confirmar que deseamos instalarlo y esperaremos a que termine.

```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates iptables libip6tc2 libmpdec3 libnetfilter-contrack3 libnfnetwork0
  libpython3-stdlib libpython3.9-minimal libpython3.9-stdlib libssqlite3-0 media-types openssl
  python3 python3-minimal python3.9 python3.9-minimal
Suggested packages:
  firewalld python3-doc python3-tk python3-venv python3.9-venv python3.9-doc binutils
  binfmt-support
The following NEW packages will be installed:
  ca-certificates iptables libip6tc2 libmpdec3 libnetfilter-contrack3 libnfnetwork0
  libpython3-stdlib libpython3.9-minimal libpython3.9-stdlib libssqlite3-0 media-types openssl
  python3 python3-minimal python3.9 python3.9-minimal ufw
0 upgraded, 17 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,568 kB of archives.
After this operation, 27.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y_
```

```
Born2beroot [Running]
Setting up iptables (1.8.7-1) ...
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables (ip6tables) in auto mode
update-alternatives: using /usr/sbin/iptables-nft to provide /usr/sbin/iptables (iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-nft to provide /usr/sbin/ip6tables (ip6tables) in auto mode
update-alternatives: using /usr/sbin/arpTables-nft to provide /usr/sbin/arpTables (arpTables) in auto mode
update-alternatives: using /usr/sbin/eBTables-nft to provide /usr/sbin/eBTables (eBTables) in auto mode
Setting up python3 (3.9.2-3) ...
running python rtupdate hooks for python3.9...
running python post-rtupdate hooks for python3.9...
Setting up ufw (0.36-7.1) ...

Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
Created symlink /etc/systemd/system/multi-user.target.wants/ufw.service → /lib/systemd/system/ufw.service.
Processing triggers for rsyslog (8.2102.0-2+deb11u1) ...
Processing triggers for libc-bin (2.31-13+deb11u3) ...
Processing triggers for ca-certificates (20210119) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@gemartin42:/home/gemartin# _
```

2º Una vez instalado debemos habilitarlo , para ello debemos poner el siguiente comando `sudo ufw enable` y acto seguido nos debe indicar que el firewall esta activo.

```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo ufw enable
Firewall is active and enabled on system startup
root@gemartin42:/home/gemartin# _
```

3º Ahora lo que debemos hacer es que nuestro firewall permita las conexiones que se lleven a cabo mediante el puerto 4242. Lo haremos con el siguiente comando `sudo ufw allow 4242`.

```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo ufw allow 4242
Rule added
Rule added (v6)
root@gemartin42:/home/gemartin# _
```

4º Por último comprobaremos que esta todo correctamente configurado mirando el estado de nuestro cortafuegos , en donde ya debe aparecer como permitidas las conexiones mediante el puerto 4242. Para ver el estado daremos uso del comando `sudo ufw status`.

```
Born2beroot [Running]
root@gemartin42:/home/gemartin# sudo ufw status
Status: active

To                         Action      From
--                         ----       ---
4242                       ALLOW      Anywhere
4242 (v6)                   ALLOW      Anywhere (v6)

root@gemartin42:/home/gemartin# _
```

4-4 Configurar contraseña fuerte para sudo

1º Crearemos un fichero en la ruta /etc/sudoers.d/ a mi fichero yo le he decidido llamar sudo_config ya que en ese fichero se almacenará la configuración de la contraseña. El comando exacto para crear el fichero es `touch /etc/sudoers.d/sudo_config`.

```
Born2beroot [Running]
root@gemartin42:~# touch /etc/sudoers.d/sudo_config
root@gemartin42:~# _
```

2º Debemos crear el directorio sudo en la ruta /var/log porque cada comando que ejecutemos con sudo , tanto el input como el output debe q almacenado en ese directorio. Para crearlo utilizaremos el comando `mkdir /var/log/sudo`.

```
Born2beroot [Running]
root@gemartin42:~# mkdir /var/log/sudo
root@gemartin42:~# _
```

3º Debemos editar el fichero creado en el paso 1. Como he comentado anteriormente puedes utilizar el editor que mas te guste , pero yo dare nano. Comando para editar el fichero: `nano /etc/sudoers.d/sudo_config`.

```
Born2beroot [Running]
root@gemartin42:~# nano /etc/sudoers.d/sudo_config
```

4º Una vez estamos editando el fichero deberemos introducir los siguientes comandos para cumplir todos los requisitos que pide el subject.

```
Defaults    passwd_tries=3
Defaults    badpass_message="Mensaje de error personalizado"
Defaults    logfile="/var/log/sudo/sudo_config"
Defaults    log_input, log_output
Defaults    iolog_dir="/var/log/sudo"
Defaults    requiretty
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
```

► Como debería verse el fichero.

```
Born2beroot [Running]
GNU nano 5.4                               /etc/sudoers.d/sudo_config *
Defaults    passwd_tries=3
Defaults    badpass_message="Mensaje de error personalizado"
Defaults    logfile="/var/log/sudo/sudo_config"
Defaults    log_input, log_output
Defaults    iolog_dir="/var/log/sudo"
Defaults    requiretty
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap
```

⊗ Que hace cada comando ?

```
Born2beroot [Running]
GNU nano 5.4          /etc/sudoers.d/sudo_config
Defaults    passwd_tries=3
Defaults    badpass_message="Mensaje de error personalizado"
Defaults    logfile="/var/log/sudo/sudo_config"
Defaults    log_input, log_output
Defaults    iolog_dir="/var/log/sudo"
Defaults    requiretty
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

Numero de intentos en caso de introducir una contraseña erronea
Mensaje que se mostrara por pantalla en caso de que la contraseña introducida sea incorrecto
Archivo en el que quedaran registrados todos los comandos sudo
Para que cada comando ejecutado con sudo, tanto input como output quede archivado en el directorio especificado
Para activar el modo TTY
Para restringir los directorios utilizables por sudo
```

4-5 Configuración de política de contraseñas fuerte

1 ° El primer paso será editar el fichero login.defs.

```
Born2beroot [Running]
root@gemartin42:~# nano /etc/login.defs
```

2 ° Una vez estemos editando el fichero modificaremos los siguientes parametros:

- PASS_MAX_DAYS 99999 -> PASS_MAX_DAYS 30
- PASS_MIN_DAYS 0 -> PASS_MIN_DAYS 2

```
Born2beroot [Running]
GNU nano 5.4          /etc/login.defs *
# used as group permissions, e. g. 022 will become 002.
#
# Prefix these values with "0" to get octal, "0x" to get hexadecimal.
#
ERASECHAR      0177
KILLCHAR        025
UMASK           022

#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   30
PASS_MIN_DAYS   2
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN          1000
UID_MAX         60000
# System accounts
#SYS_UID_MIN     100
#SYS_UID_MAX     999

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN          1000
GID_MAX         60000

^G Help          ^O Write Out  ^W Where Is  ^K Cut          ^T Execute  ^C Location  M-U Undo
^X Exit          ^R Read File  ^N Replace  ^U Paste  ^J Justify  ^_ Go To Line M-E Redo
Left %
```

PASS_MAX_DAYS: Es el tiempo de expiración de la contraseña. El numero corresponde a días.

PASS_MIN_DAYS: El número mínimo de días permitido antes de modificar una contraseña.

PASS_WARN_AGE: El usuario recibira un mensaje de aviso indicando que faltan los dias especificados para que expire su contraseña.

3º Para poder seguir con la configuración debemos instalar los siguientes paquetes con este comando `sudo apt install libpam-pwquality`, seguido pondremos `Y` para confirmar la instalación y esperaremos a que termine.

```
● ● ● Born2beroot [Running]
root@gemartin42:~# sudo apt install libpam-pwquality
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cracklib-runtime file libcrack2 libmagic-mgc libmagic1 libpwquality-common libpwquality1
The following NEW packages will be installed:
  cracklib-runtime file libcrack2 libmagic-mgc libmagic1 libpam-pwquality libpwquality-common
  libpwquality1
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 757 KB of archives.
After this operation, 8,480 KB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

4º Lo siguiente que debemos hacer es volver a editar un fichero y modificar algunas líneas. Haremos `nano /etc/pam.d/common-password`.

```
● ● ● Born2beroot [Running]
root@gemartin42:~# nano /etc/pam.d/common-password _
```

5º Despues de `retry=3` debemos añadir los siguientes comandos:

```
minlen=10
ucredit=-1
dcredit=-1
lcredit=-1
maxrepeat=3
reject_username
difok=7
enforce_for_root
```

► Así debe ser la línea ↴

```
retry=3 minlen=10 ucredit=-1 dcredit=-1 lcredit=-1 maxrepeat=3 reject_username difok=7
|enforce_for_root
```

► Así se debe ver en el fichero ↴

```
Born2beroot [Running]
GNU nano 5.4
/etc/pam.d/common-password

# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.

# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# 11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# for compatibility. The "obscure" option replaces the old
# `OBSCURE_CHECKS_ENAB' option in login.defs. See the pam_unix manpage
# for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password      requisite          pam_pwquality.so retry=3 minlen=10 ucred=-1 dcredit>
password      [success=1 default=ignore]    pam_unix.so obscure use_authok try_first_pass yesc>
# here's the fallback if no module succeeds
password      requisite          pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password      required           pam_permit.so
# and here are more per-package modules (the "Additional" block)

[ Read 34 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^Y Replace   ^U Paste     ^J Justify   ^L Go To Line M-E Redo
                                         Left %
```

💡 Que hace cada comando ?

minlen=10 ➤ La cantidad mínima de caracteres que debe contener la contraseña.

ucred=-1 ➤ Como mínimo debe contener una letra mayúscula. Ponemos el - ya que debe contener como mínimo un carácter, si ponemos + no referimos a como máximo esos caracteres.

dcredit=-1 ➤ Como mínimo debe contener un dígito.

lcredit=-1 ➤ Como mínimo debe contener una letra minúscula.

maxrepeat=3 ➤ No puede tener más de 3 veces seguidas el mismo carácter.

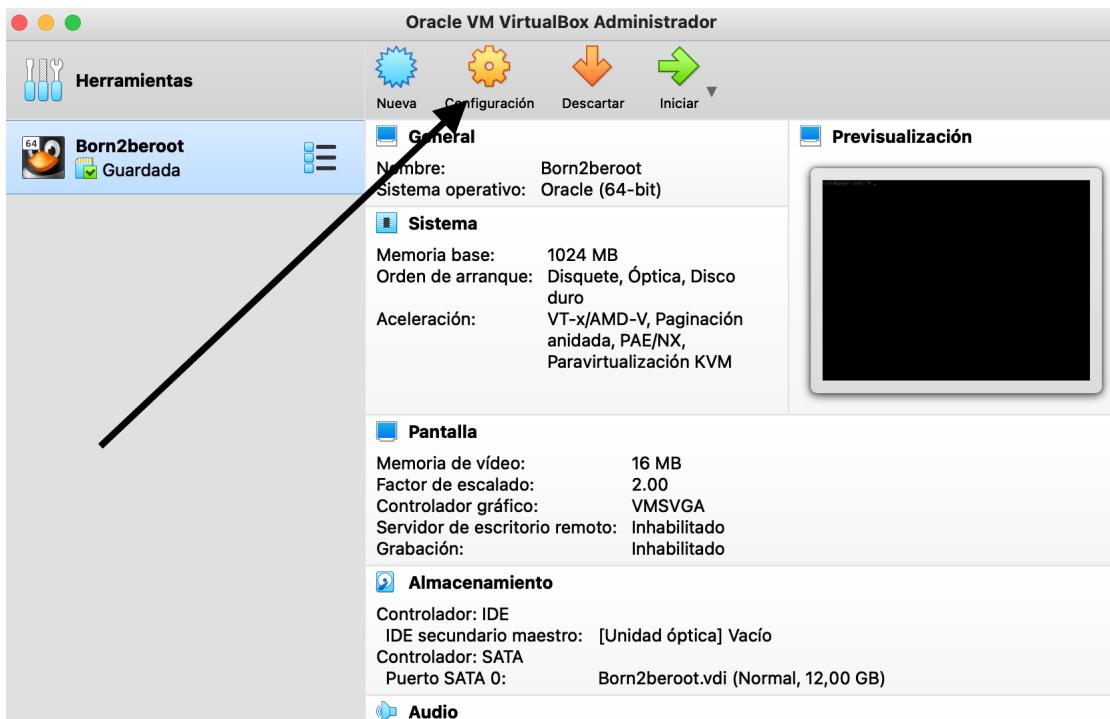
reject_username ➤ No puede contener el nombre del usuario.

difok=7 ➤ Debe tener al menos 7 caracteres que no sean parte de la antigua contraseña.

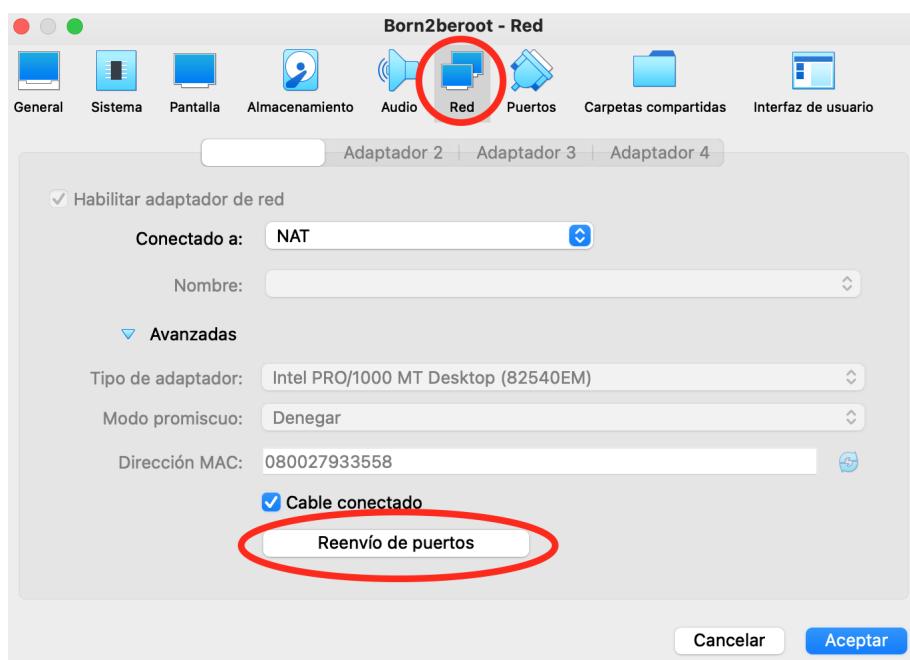
enforce_for_root ➤ Implementaremos esta política para el usuario root.

’4-6 Conectarse via SSH 🛡

1º Para conectarnos por SSH debemos cerrar la máquina, abrir VirtualBox y darle a configuración.



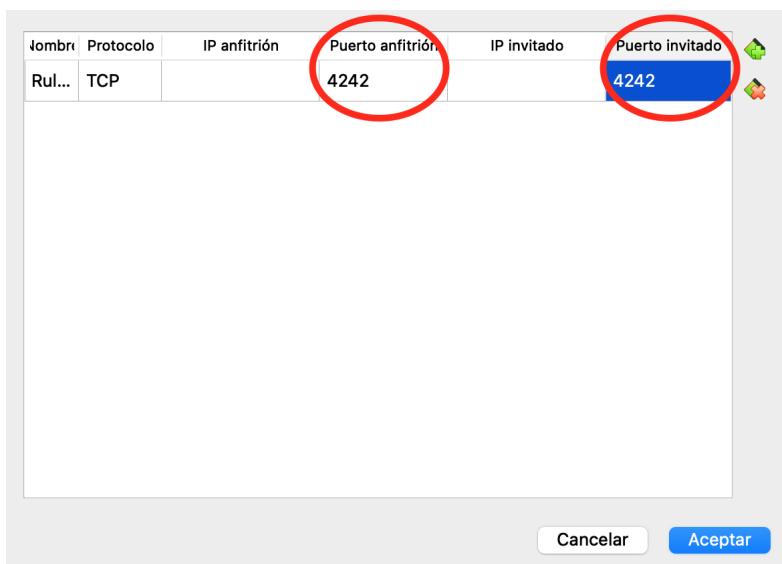
2º Una vez en configuración debemos pinchar sobre el apartado de **Red**, pincharemos sobre **Avanzadas** para que así nos muestre más opciones y le daremos a **Reenvío de puertos**.



3º Pincharemos sobre el siguiente emotícono para agregar una regla de reenvío.



4º Por último agregaremos el puerto 4242 al anfitrión y al invitado. Las IP's no son necesarias. Pincharemos sobre el botón de aceptar para que se apliquen los cambios.



► Para poder conectarnos a la máquina virtual desde la real debemos abrir un terminal en la máquina real y escribir `ssh gemartin@localhost -p 4242` nos pedirá la clave del usuario y una vez la introduzcamos ya nos saldrá el login en verde y eso significa que estaremos conectados.

```
gemartin@car15s4 ~ % ssh gemartin@localhost -p 4242
```

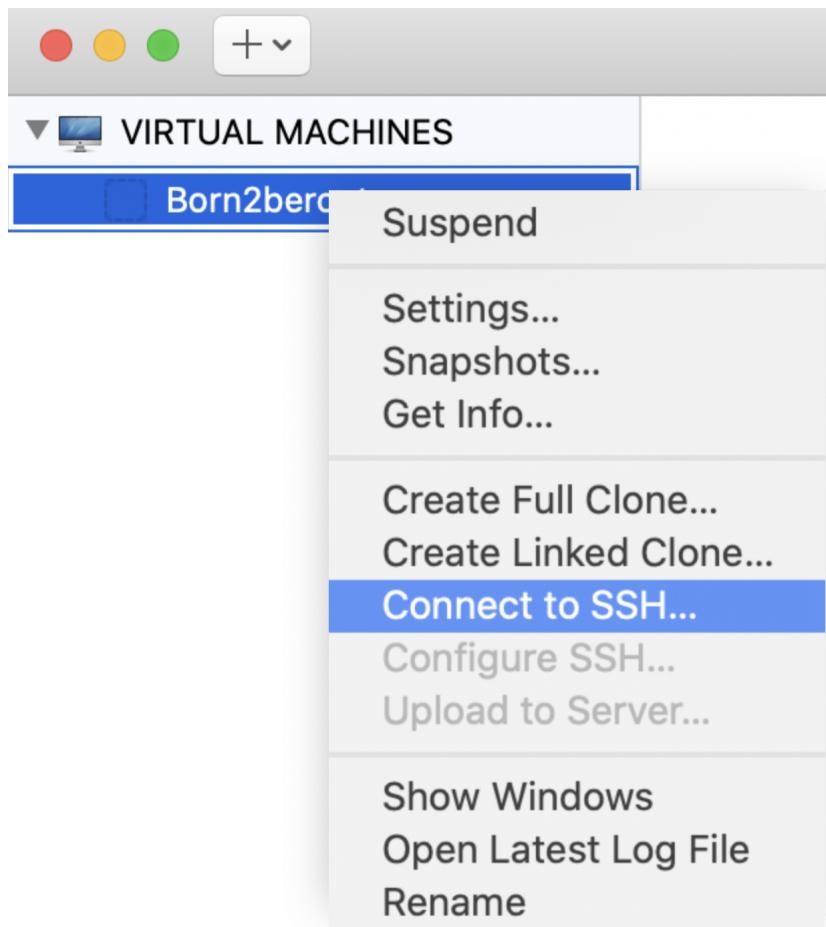
```
gemartin@car15s4 ~ % ssh gemartin@localhost -p 4242
gemartin@localhost's password:
Linux gemartin42 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

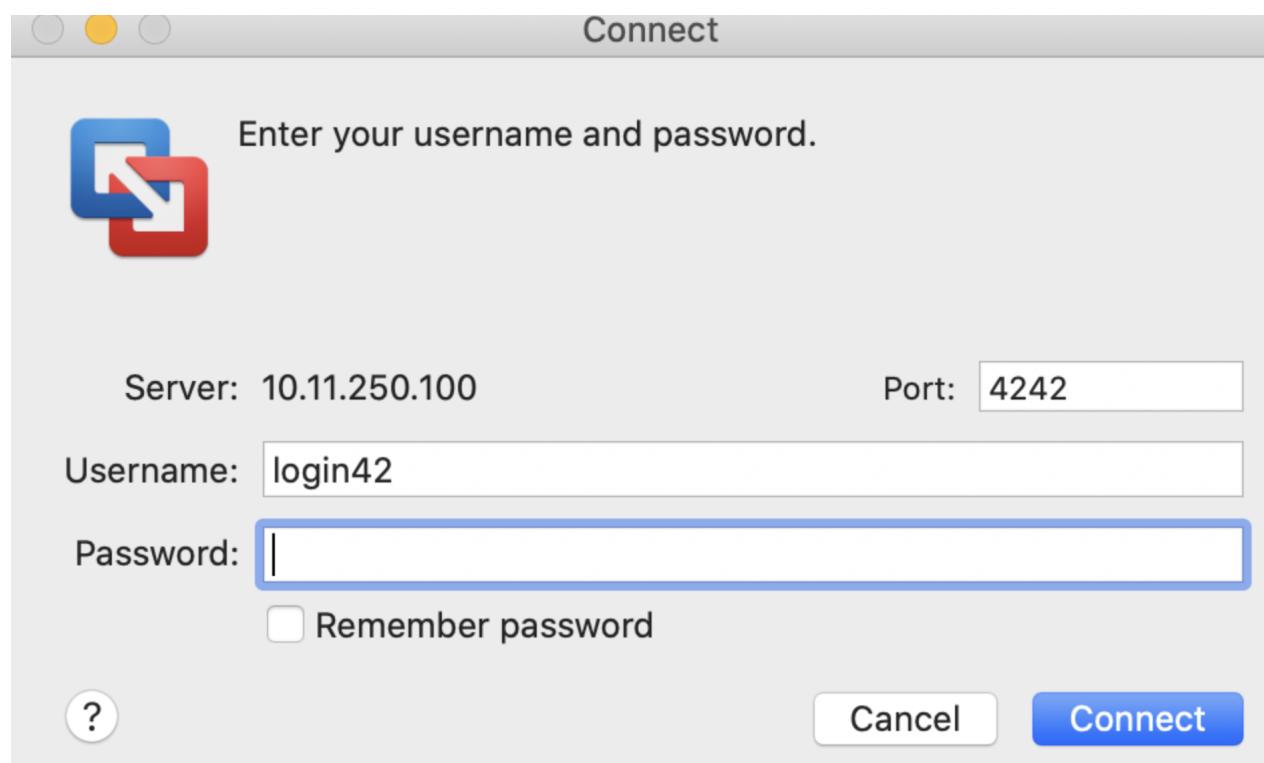
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct 27 00:28:00 2022
gemartin@gemartin42:~$
```

⚠ Siquieres conectarte por SSH con VMware ⚡

1º Daremos click derecho sobre nuestra máquina y escogeremos la opción `Connect to SSH`.



2º Una vez se nos abra la siguiente pestaña debemos llenar todos los campos. En port debemos poner 4242 para indicar que queremos conectarnos por ese puerto. Los siguientes campos son el username de tu maquina, en mi caso gemartin y la contraseña del usuario , en mi caso Hola42spain .



También podemos conectarnos mediante el terminal pero debemos sustituir localhost por la IP de la máquina virtual quedaría algo así: ssh gemartin@10.11.250.100 -p 4242 . Revisa la IP de tu máquina virtual y sustituéyela por la que pongo de ejemplo.

5- Script

Esta es una parte muy importante del proyecto. Debes prestar atención en todo, muy importante no copiar y pegar directamente el fichero sin que hace cada cosa. En la evaluación debes explicar cada comando si el evaluador lo pide.

• **Que es un script ?** Es una secuencia de comandos guardada en un fichero que cuando se ejecuta hará la función de cada comando.

’5-1 Architecture

Para poder ver la arquitectura del SO y su versión de kernel utilizaremos el comando `uname -a` (" -a" == "--all") que básicamente printará toda información excepto si el tipo de procesador es desconocido o la plataforma de hardware.

```
root@gemartin42:/home/gemartin# uname -a
Linux gemartin42 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64 GNU/Linux
```

’5-2 Núcleos físicos

Para poder mostrar el número de núcleos físicos haremos uso del fichero `/proc/cpuinfo` el cual proporciona información acerca del procesador: tipo, marca, modelo, rendimiento, etc. Usaremos el comando `grep "physical id" /proc/cpuinfo | wc -l` con el comando grep buscaremos dentro del fichero "physical id" y con wc -l contaremos las líneas del resultado de grep. Esto lo hacemos ya que la manera de cuantificar los núcleos no es muy común. Si hay un procesador marcará 0 y si tiene más de un procesador, mostrará toda la información del procesador por separado contando los procesadores usando la notación cero. De esta manera simplemente contaremos las líneas que hay ya que es más cómodo cuantificarlo así.

```
root@gemartin42:/home/gemartin# grep "physical id" /proc/cpuinfo | wc -l
1
root@gemartin42:/home/gemartin# _
```

’5-3 Núcleos virtuales

Para poder mostrar el número de núcleos virtuales es muy parecido al anterior. Haremos uso de nuevo del fichero `/proc/cpuinfo`, pero, en este caso utilizaremos el comando `grep processor /proc/cpuinfo | wc -l`. El uso es prácticamente el mismo al anterior solo que en vez de contar las líneas de "physical id" lo haremos de processor. Lo hacemos así por el mismo motivo de antes, la manera de cuantificar marca 0 si hay un procesador.

```
root@gemartin42:/home/gemartin# grep processor /proc/cpuinfo | wc -l
1
```

’5-4 Memoria RAM

Para mostrar la memoria ram haremos uso del comando `free` para así ver al momento información sobre la ram, la parte usada, libre, reservada para otros recursos, etc. Para más info sobre el comando pondremos `free --help`. Nosotros daremos uso de `free --mega` ya que en el subject apunta esa unidad de medida (Megabyte). Es importante poner `--mega` y no `-m`. Con `-m` nos referiremos a la unidad de medida Mebibyte y no es la que especifica el subject.

```
root@gemartin42:/home/gemartin# free --mega
              total        used        free      shared  buff/cache   available
Mem:       1023         87        562          0         374        788
Swap:      1023          0       1023
root@gemartin42:/home/gemartin#
```

Una vez hemos ejecutado este comando debemos filtrar nuestra búsqueda ya que no necesitamos toda la información que nos aporta, lo primero que debemos mostrar es la memoria usada, para ello haremos uso del comando `awk` que lo que hace este comando es para procesar datos basados en archivos de texto, es decir, podemos utilizar los datos que nos interesen de X fichero. Por último lo que haremos será comparar si la primera palabra de una fila es igual a "Mem:" printaremos la tercera palabra de esa fila que será la memoria usada. Todo el comando junto sería `free -m | awk '$1 == "Mem:" {print $3}'`. En el script el valor de retorno de este comando se lo asignaremos a una variable que concatenaremos con otras variables para que todo quede igual como especifica el subject.

```
root@gemartin42:/home/gemartin# free --mega | awk '$1 == "Mem:" {print $3}'
87
```

Para obtener la memoria total el comando es prácticamente igual al anterior lo único que deberemos cambiar es que en vez de printar la tercera palabra de la fila queremos la segunda `free --mega | awk '$1 == "Mem:" {print $2}'`.

```
root@gemartin42:/home/gemartin# free --mega | awk '$1 == "Mem:" {print $2}'
1023
```

Por última parte debemos calcular el % de memoria usada. El comando de nuevo es parecido a los anteriores la única modificación que haremos es parte del printeo. Como la operación para conseguir el tanto poriento no es exacta nos puede dar muchos decimales y en el subject solo aparece así que nosotros haremos lo mismo, por eso utilizamos `%.2f` para que así solo se muestren 2 decimales. Otra cosa que quizás no sepas es en el printeo para que se muestre un % hay que poner %% . Todo el comando `free --mega | awk '$1 == "Mem:" {printf("(%.2f%%)\n", $3/$2*100)}'`.

```
root@gemartin42:/home/gemartin# free --mega | awk '$1 == "Mem:" {printf("(%.2f%%)\n", $3/$2*100)}'
(8.50%)
```

' 5-5 Memoria del disco

Para poder ver la memoria del disco ocupada y disponible utilizaremos el comando `df` que significa "disk filesystem", se utiliza para obtener un resumen completo del uso del espacio en disco. Como en el subject indica la memoria utilizada se muestra en MB así que entonces utilizaremos el flag `-m`. Acto seguido haremos un grep para que solo nos muestre las líneas que contengan `/dev/` y seguidamente volveremos a hacer otro grep con el flag `-v` para excluir las líneas que contengan `/boot`. Por último utilizaremos el comando awk y sumaremos el valor de la tercera palabra de cada línea para una vez sumadas todas las líneas printar el resultado final de la suma. El comando entero es el siguiente: `df -m | grep "/dev/" | grep -v "/boot" | awk '{memory_use += $3} END {print memory_use}'`.

```
root@gemartin42:/home/gemartin# df -m | grep "/dev/" | grep -v "/boot" | awk '{memory_use += $3} END {print memory_use}'  
965  
root@gemartin42:/home/gemartin# _
```

Para obtener el espacio total utilizaremos un comando muy parecido. Las únicas diferencias serán que los valores que sumaremos serán los `$2` de `$3` y la otra diferencia es que en el subject aparece el tamaño total en Gb así como el resultado de la suma nos da el número en Mb debe transformarlo a Gb, para ello debemos dividir el número entre 1024 y quitar los decimales.

```
root@gemartin42:/home/gemartin# df -m | grep "/dev/" | grep -v "/boot" | awk '{memory_result += $2} END {printf ("%.\nGb"), memory_result/1024}'  
26Gb
```

Por último debemos mostrar un porcentaje de la memoria usada. Para ello, de nuevo, utilizaremos un comando muy parecido a los dos anteriores. La única diferencia es que combinaremos los dos comandos anteriores para tener dos variables, una que representa la memoria usada y la total. Hecho esto haremos una operación para conseguir el tanto por ciento `use/total*100` y el resultado de esta operación lo printaremos. El resultado aparece en el subject, entre paréntesis y con el símbolo % al final. El comando final es este: `df -m | grep "/dev/" | grep -v "/boot" | awk '{use += $3} {total += $2} END {printf("(%d%%)\n"), use/total*100}'`.

```
root@gemartin42:/home/gemartin# df -m | grep "/dev/" | grep -v "/boot" | awk '{use += $3} {total += $2} END {printf("(%d%%)\n"), use/total*100}'  
(8%)  
root@gemartin42:/home/gemartin#
```

' 5-6 Porcentaje uso de CPU

Para poder ver el porcentaje de uso de CPU haremos uso del comando `vmstat`. Este muestra estadísticas del sistema, permitiendo obtener un detalle general de los procesos, uso de memoria, actividad de CPU, estado del sistema, etc. Podríamos poner si ninguna opción pero en mi caso pondré un intervalo de segundos de 1 a 4. También daremos uso del comando `tail -1` que este lo que nos va a permitir es que solo produzca output la última línea, entonces de las 4 generadas solo se printará la última. Por último solo printaremos la palabra 15 que es el uso de memoria disponible. El comando entero es el siguiente: `vmstat 1 4 | tail -1 | awk '{print %15}'`. El resultado de este comando solo es una parte del resultado final ya que todavía hay que hacer alguna operación en el script para que quede bien. Lo que habría que hacer es a 100 restarle la cantidad que nos ha devuelto nuestro comando, el resultado de esa operación lo printaremos con un decimal y un % al final y ya estaría hecha la operación.

```
root@gemartin42:/home/gemartin# vmstat 1 3| tail -1 | awk '{print $15}'  
100
```

' 5-7 Último reinicio

Para ver la fecha y hora de nuestro último reinicio haremos uso del comando `who` con el flag `-b` ya que con ese flag nos mostrará por pantalla el tiempo del último arranque del sistema. Como ya nos ha pasado anteriormente nos muestra más información de la que deseamos así que filtraremos y solo mostraremos lo que nos interesa, para ello haremos uso del comando awk y compararemos si la primera palabra de una línea es "system" y si es así printará por pantalla la tercera palabra de esa línea, un espacio y la cuarta palabra. El comando entero sería el siguiente: `who -b | awk '$1 == "system" {print $3 " " $4}'`.

```
root@gemartin42:/home/gemartin# who -b | awk '$1 == "system" {print $3 " " $4}'  
2022-07-13 22:12
```

' 5-8 Uso LVM

Para checar si LVM está activo o no haremos uso del comando `lsblk`, este nos muestra información de todos los dispositivos de bloque (discos duros, SSD, memorias, etc) entre toda la información que proporciona podemos ver lvm en el tipo de gestor. Para este comando haremos un if o printaremos Yes o No. Basicamente la condición que buscamos será contar el número de líneas en las que aparece "lvm" y si hay más de 0 printaremos Yes, si hay 0 se printará No. Todo el comando sería: `if [$(lsblk | grep "lvm" | wc -l) -gt 0]; then echo yes; else echo no;`

```
root@gemartin42:/home/gemartin# if [ $(lsblk | grep lvm | wc -l) -gt 0 ]; then echo yes; else echo no;  
; fi  
yes
```

'5-9 Conexiones TCP

Para mirar el numero de conexiones TCP establecidas. Utilizaremos el comando `ss` sustituyendo al ya obsoleto netstat. Filtraremos con el flag para que solo se muestren las conexiones TCP. Por ultimo haremos un grep para ver las que estan establecidas ya que tambien hay solo de esc cerraremos con wc -l para que cuente el numero de lineas. El comando queda tal que asi: `ss -ta | grep ESTAB | wc -l`.

```
root@gemartin42:/home/gemartin# ss -ta | grep ESTAB | wc -l
1
root@gemartin42:/home/gemartin#
```

'5-10 Número de usuarios

Daremos uso del comando `users` que nos mostrará el nombre de los usuarios que hay, sabiendo esto, pondremos wc -w para que cuente la cantidad de palabras que hay en la salida del comando. El comando entero queda así `users | wc -w`.

```
root@gemartin42:/home/gemartin# users | wc -w
2
```

'5-11 Dirección IP y MAC

Para obtener la dirección del host haremos uso del comando `hostname -I` y para obtener la MAC haremos uso del comando `ip link` que se usa para mostrar o modificar las interfaces de red. Como aparecen más de una interfaz, IP's etc. Utilizaremos el comando grep para buscar lo que deseamos y asi poder printar por pantalla solo lo que nos piden. Para ello pondremos `ip link | grep "link/ether" | awk '{print $2}'` y de manera solo printaremos la MAC.

```
root@gemartin42:/home/gemartin# ip link | grep "link/ether" | awk '{print $2}'
08:00:27:93:35:58
```

'5-12 Número de comandos ejecutados con sudo

Para poder obtener el numero de comandos que son ejecutados con sudo haremos uso del comando `journalctl` que este es una herramienta que encarga de recopilar y administrar los registros del sistema. Acto seguido pondremos `_COMM=sudo` para así filtrar las entradas especificando su ruta. En nuestro ponemos `_COMM` ya que hace referencia a un script ejecutable. Una vez tengamos filtrada la búsqueda y solo aparezcan los registros de sudo todavía deberemos filtrar un poco más ya que cuando incias o cierras sesion de root tambien aparece en el registro, entonces para terminar de filtrar pondremos un `grep COMMAND` y asi solo aparecerán las líneas de comandos. Por ultimo pondremos `wc -l` para que así nos salgan el número de las líneas. El comando entero es el siguiente: `journalctl _COMM=sudo | grep COMMAND | wc -l`. Para comprobar que funcione correctamente podemos correr el comando en el terminal, poner un comando que incluya sudo y volver a correr el comando y deberá incrementar el número de ejecuciones de sudo.

```
root@gemartin42:/home/gemartin# journalctl _COMM=sudo | grep COMMAND | wc -l
36
root@gemartin42:/home/gemartin# sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@gemartin42:/home/gemartin# journalctl _COMM=sudo | grep COMMAND | wc -l
37
root@gemartin42:/home/gemartin# _
```

'5-13 Resultado total del script

⚠ Recuerda no hacer copia y pega si no sabes el funcionamiento de cada comando ⚠

```
#!/bin/bash

# ARCH
arch=$(uname -a)

# CPU PHYSICAL
cpuf=$(grep "physical id" /proc/cpuinfo | wc -l)

# CPU VIRTUAL
cpuv=$(grep "processor" /proc/cpuinfo | wc -l)

# RAM
ram_total=$(free --mega | awk '$1 == "Mem:" {print $2}')
ram_use=$(free --mega | awk '$1 == "Mem:" {print $3}')
ram_percent=$(free --mega | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')

# DISK
disk_total=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_t += $2} END {printf ("% .1f Gb\n"), disk_t/1024}')
```

```

disk_use=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} END {print disk_u}')
disk_percent=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} {disk_t+= $2} END {printf("%d"), disk_u/disk_t*100}')

# CPU LOAD
cpul=$(vmstat 1 2 | tail -1 | awk '{printf $15}')
cpu_op=$((expr 100 - $cpul))
cpu_fin=$(printf "%..1f" $cpu_op)

# LAST BOOT
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

# LVM USE
lvmu=$(if [ $(lsblk | grep "lvm" | wc -l) -gt 0 ]; then echo yes; else echo no; fi)

# TCP CONNEXIONS
tcpc=$(ss -ta | grep ESTAB | wc -l)

# USER LOG
ulog=$(users | wc -w)

# NETWORK
ip=$(hostname -I)
mac=$(ip link | grep "link/ether" | awk '{print $2}')

# SUDO
cmnd=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

wall "  Architecture: $arch
      CPU physical: $cpuf
      VCPU: $cpuv
      Memory Usage: ${ram_use}/${ram_total}MB ($ram_percent%)
      Disk Usage: ${disk_use}/${disk_total} ($disk_percent%)
      CPU load: $cpu_fin%
      Last boot: $lb
      LVM use: $lvmu
      Connections TCP: $tcpc ESTABLISHED
      User log: $ulog
      Network: IP $ip ($mac)
      Sudo: $cmnd cmd"

```

Script visto desde nano ↵

```

● ● ● usuario — gemartin@gemartin42: ~ — ssh gemartin@localhost -p 4242 — 129x69
GNU nano 5.4                               monitoring.sh *

#!/bin/bash

# ARCH
arch=$(uname -a)

# CPU PHYSICAL
cpuf=$(grep "physical id" /proc/cpuinfo | wc -l)

# CPU VIRTUAL
cpuv=$(grep "processor" /proc/cpuinfo | wc -l)

# RAM
ram_total=$(free --mega | awk '$1 == "Mem:" {print $2}')
ram_use=$(free --mega | awk '$1 == "Mem:" {print $3}')
ram_percent=$(free --mega | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')

# DISK
disk_total=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_t += $2} END {printf ("% .1fGb\n"), disk_t/1024}')
disk_use=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} END {print disk_u}')
disk_percent=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} {disk_t+= $2} END {printf("%d"), disk_u/disk_t*100}')

# CPU LOAD
cpul=$(vmstat 1 2 | tail -1 | awk '{printf $15}')
cpu_op=$((100 - $cpul))
cpu_fin=$(printf "% .1f" $cpu_op)

# LAST BOOT
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

# LVM USE
lvmu=$(if [ $(lsblk | grep "lvm" | wc -l) -gt 0 ]; then echo yes; else echo no; fi)

# TCP CONNEXIONS
tcpc=$(ss -ta | grep ESTAB | wc -l)

# USER LOG
ulog=$(users | wc -w)

# NETWORK
ip=$(hostname -I)
mac=$(ip link | grep "link/ether" | awk '{print $2}')

# SUDO
cmnd=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

wall " Architecture: $arch
      CPU physical: $cpuf
      vCPU: $cpuv
      Memory Usage: $ram_use/${ram_total}MB ($ram_percent%)
      Disk Usage: $disk_use/${disk_total} ($disk_percent%)
      CPU load: $cpu_fin%
      Last boot: $lb
      LVM use: $lvmu
      Connections TCP: $tcpc ESTABLISHED
      User log: $ulog
      Network: IP $ip ($mac)
      Sudo: $cmnd cmd"

```

Resultado tras la ejecución del script ✓

```

root@gemartin42:/home/gemartin# sh monitoring.sh
Broadcast message from gemartin@gemartin42 (tty1) (Thu Jul 14 18:01:46 2022):
Architecture: Linux gemartin42 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64
GNU/Linux
      CPU physical: 1
      vCPU: 1
      Memory Usage: 86/1023MB (8.41%)
      Disk Usage: 965/10.7Gb (8%)
      CPU load: 0.0%
      Last boot: 2022-07-13 22:12
      LVM use: yes
      Connections TCP: 1 ESTABLISHED
      User log: 2
      Network: IP 10.0.2.15 (08:00:27:93:35:58)
      Sudo: 39 cmd

```

6- Crontab

Que es crontab? Es un administrador de procesos en segundo plano. Los procesos indicados serán ejecutados en el momento que especifique el fichero crontab.

Para tener correctamente crontab configurado debemos editar el fichero crontab con el siguiente comando `sudo crontab -u root -e`.

En el fichero debemos añadir el siguiente comando para que el script se ejecute cada 10 minutos `*/10 * * * * sh /ruta del script`.

```
Born2beroot [Running]
GNU nano 5.4
/tmp/crontab.pbIbYN/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/10 * * * * sh /home/gemartin/monitoring.sh
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^Y Replace ^U Paste ^J Justify ^L Go To Line M-E Redo

Funcionamiento de cada parametro de crontab:

m ► Corresponde al minuto en que se va a ejecutar el script, el valor va de 0 a 59.

h ► La hora exacta, se maneja el formato de 24 horas, los valores van de 0 a 23, siendo 0 las 12:00 de la medianoche. dom ► hace referencia al mes, por ejemplo se puede especificar 15 si se quiere ejecutar cada dia 15.

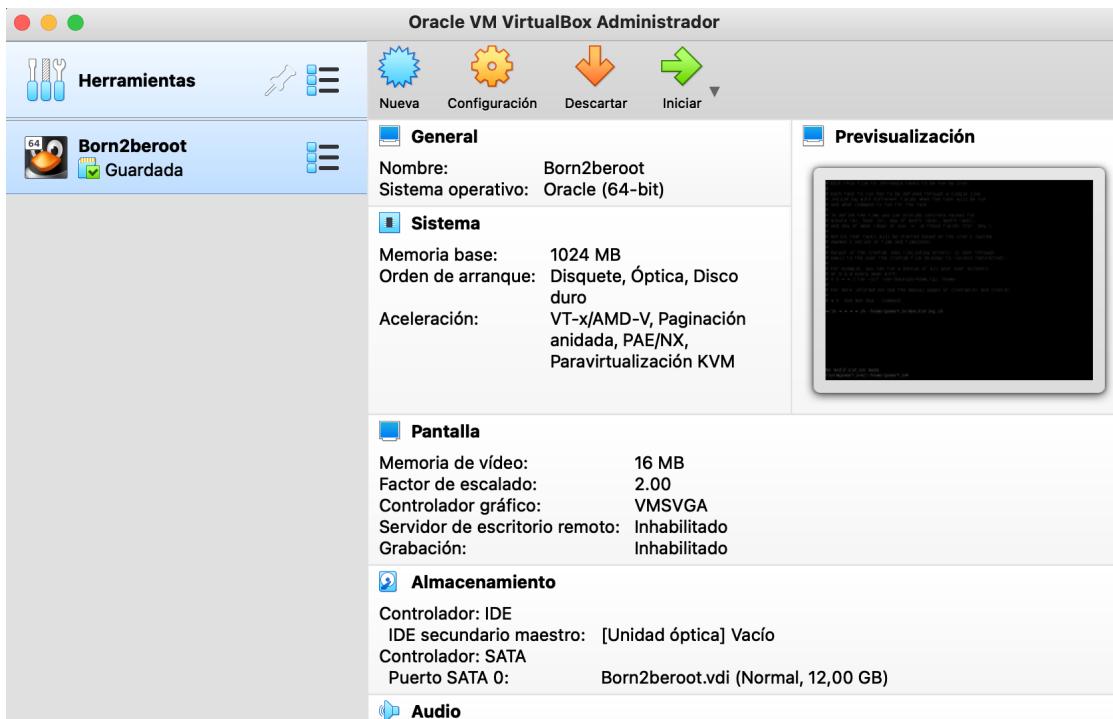
dow ► Significa el día de la semana, puede ser numérico (0 a 7, donde 0 y 7 son domingo) o las 3 primeras letras del día en inglés: mon, tue, wed, fri, sat, sun.

user ► Define el usuario que va a ejecutar el comando, puede ser root, u otro usuario diferente siempre y cuando tenga permisos de ejecución script.

command ► Refiere al comando o a la ruta absoluta del script a ejecutar.

7- Signature.txt

Para obtener la firma lo primero que debemos hacer es apagar la máquina virtual ya que una vez la enciendas o modifiques algo la firma cambia.



El siguiente paso será ubicarnos en la ruta donde tengamos el .vdi de nuestra maquina virtual.

```
[gemartin@cbr12s2 gemartin % ls
Born2beroot.vbox      Born2beroot.vbox-prev  Born2beroot.vdi
[gemartin@cbr12s2 gemartin % pwd
/gooinfre/Perso/gemartin
gemartin@cbr12s2 gemartin % ]
```

Por último haremos `shasum nombremaquina.vdi` y esto nos dara la firma. El resultado de esta firma es lo que tendremos añadir a nuestro fichero signature.txt para posteriormente subir el fichero al repositorio de la intra. Muy importante no volver a abrir la maquina ya que se modificara la firma. Para las correcciones recuerda clonar la maquina ya que asi podras encenderla sin miedo a que cambie la firma.

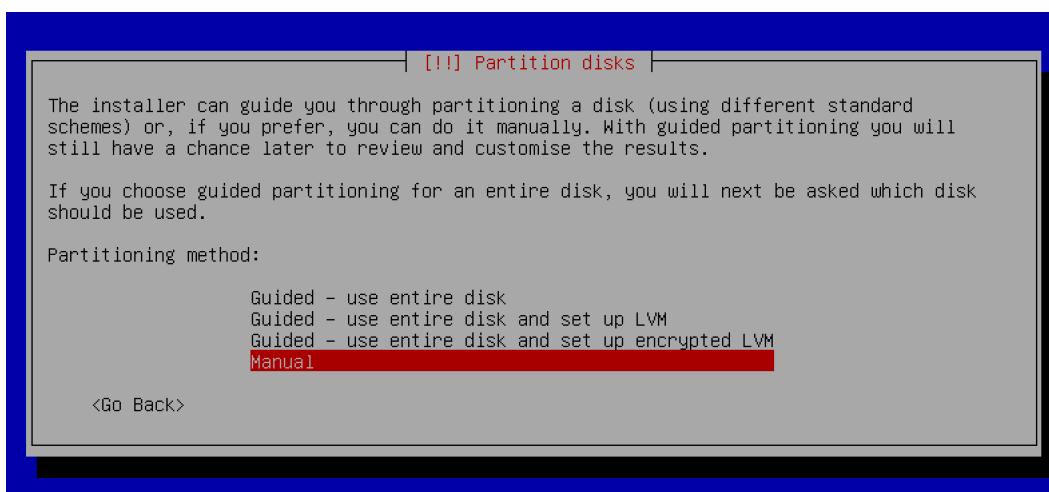
• **Que es shasum ?** Es un comando que permite identificar la integridad de un fichero mediante la suma de comprobación del hash SHA-1 de archivo.

```
[gemartin@cbr12s2 gemartin % shasum Born2beroot.vdi
28169292244a4498cff84716cdc5123c15f08c5b  Born2beroot.vdi
gemartin@cbr12s2 gemartin % ]
```

’8- Bonus ★

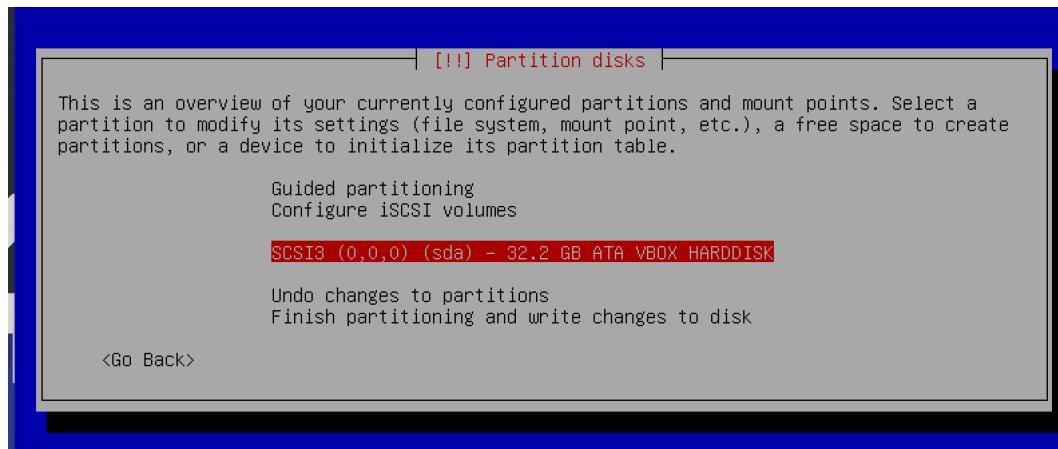
’8.1- Particionado manual del disco

1º En el momento de escoger el particionado de disco seleccionaremos manual. De esta manera podremos editar las particiones una a una.



2º En este apartado nos muestra una descripción general de nuestras particiones y puntos de montaje. Actualmente no tenemos particiones. Para crear una nueva tabla de particiones debemos escoger el dispositivo donde queremos crearlas. En nuestro caso escogeremos el único

disponible.



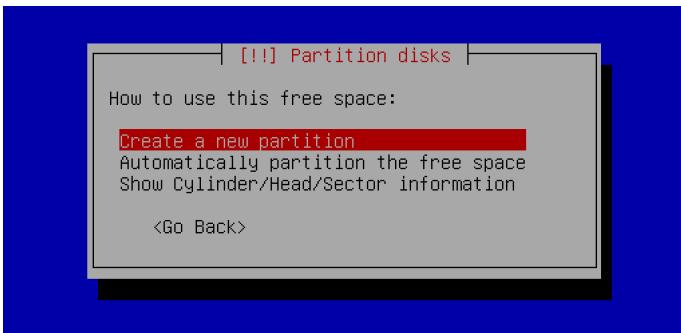
3º Aceptamos el mensaje de confirmación. Básicamente nos avisa que si ya hay particiones en el dispositivo serán eliminadas y que si estamos seguros de crear una nueva tabla de particiones vacía..



4º Una vez hemos completado el paso anterior podemos ver como nos aparece nuestra tabla de particiones vacía. Ahora debemos configurarla, ello debemos seleccionarla.



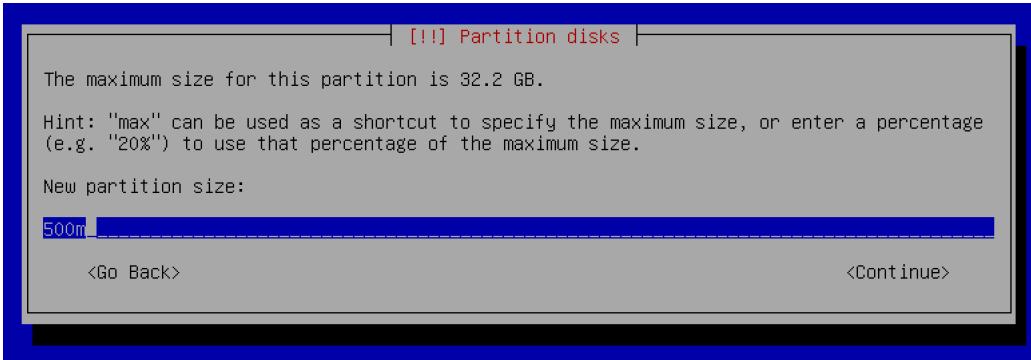
5º Crearemos una nueva partición.



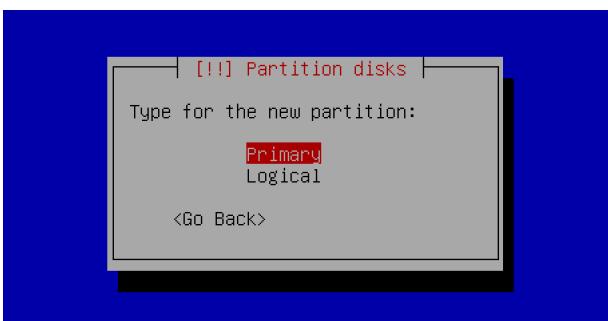
Empezaremos creando esta:

```
# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda        8:0    0 30.8G  0 disk
└─sda1     8:1    0 500M  0 part  /boot
  └─sda2     8:2    0   1K  0 part
  └─sda5     8:5    0 30.3G  0 part
    └─sda5_crypt 254:0  0 30.3G  0 crypt
      ├─LVMGroup-root 254:1  0 10G  0 lvm   /
      ├─LVMGroup-swap 254:2  0 2.3G  0 lvm  [SWAP]
      ├─LVMGroup-home 254:3  0 5G   0 lvm  /home
      ├─LVMGroup-var  254:4  0 3G   0 lvm  /var
      ├─LVMGroup-srv  254:5  0 3G   0 lvm  /srv
      ├─LVMGroup-tmp  254:6  0 3G   0 lvm  /tmp
      └─LVMGroup-var--log 254:7  0 4G   0 lvm  /var/log
sr0       11:0   1 1024M 0 rom
```

6º Como bien indica el subject el tamaño de la partición debe ser de 500 megabytes.



7º Escogemos el tipo de la partición. Escogemos primaria ya que será la partición donde se encontrará instalado el Sistema Operativo.



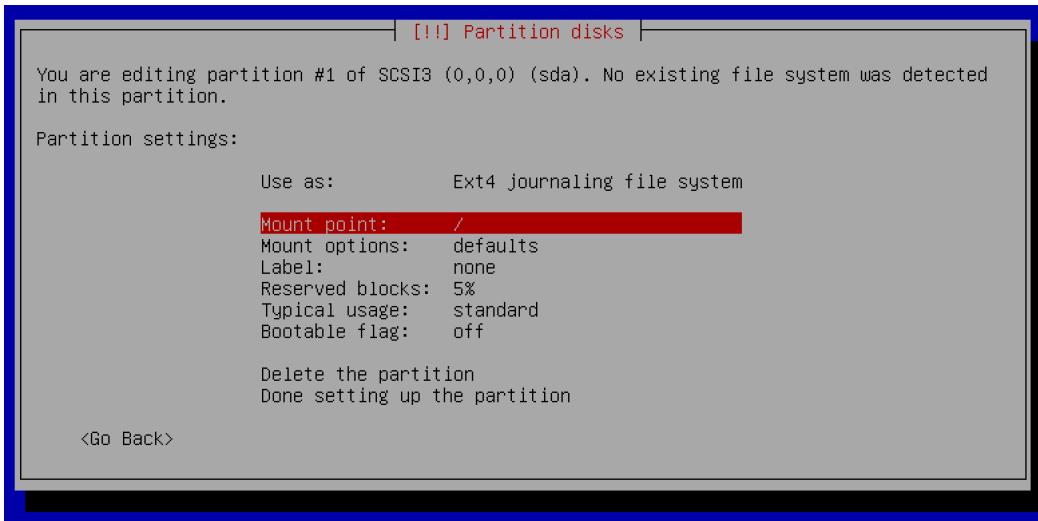
Descripción breve de todos los tipos de particiones:

- **Primaria:** La única partición en la que puede estar instalada un SO. Solo pueden haber 4 particiones primarias por disco duro o 3 primarias y extendida.
- **Secundario/Extendida:** Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición este tipo por disco, y solo sirve para contener particiones lógicas.
- **Lógica:** Ocupa una porción de la partición extendida/primaria o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (en nuestro caso usaremos ext4) y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o sus sistemas de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida, sin embargo Linux el SO con el que trabajamos actualmente lo reduce a 15, más que suficientes para realizar este proyecto.

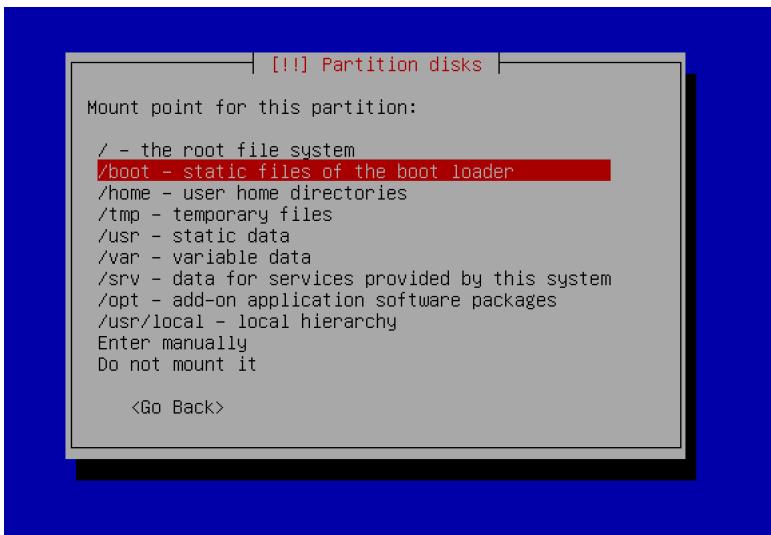
8º Seleccionaremos 'beginning' ya que queremos que la nueva partición se cree al principio del espacio disponible.



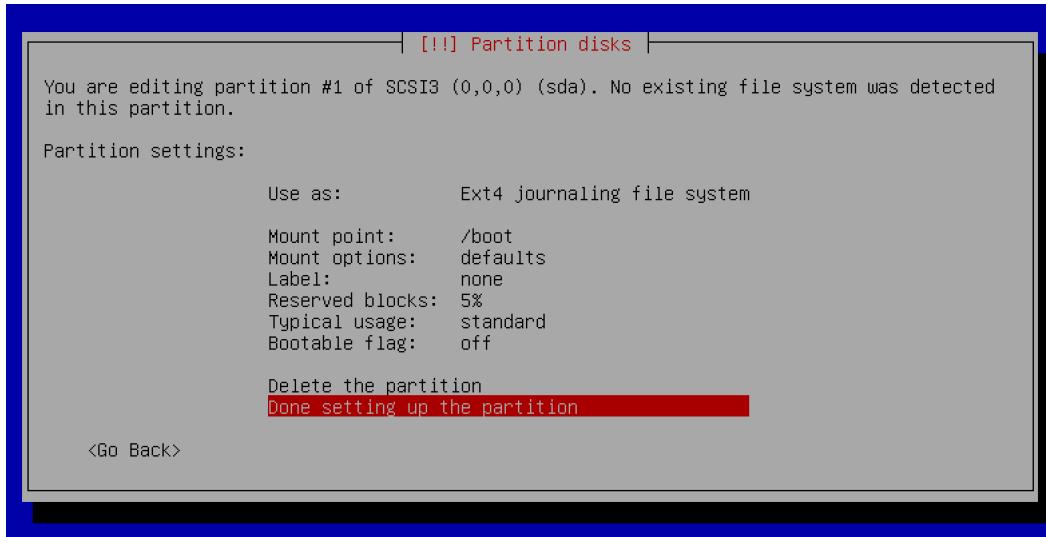
9 ° En la siguiente captura nos muestra los detalles de la partición. Modificaremos el punto de montaje al que especifica el subject.



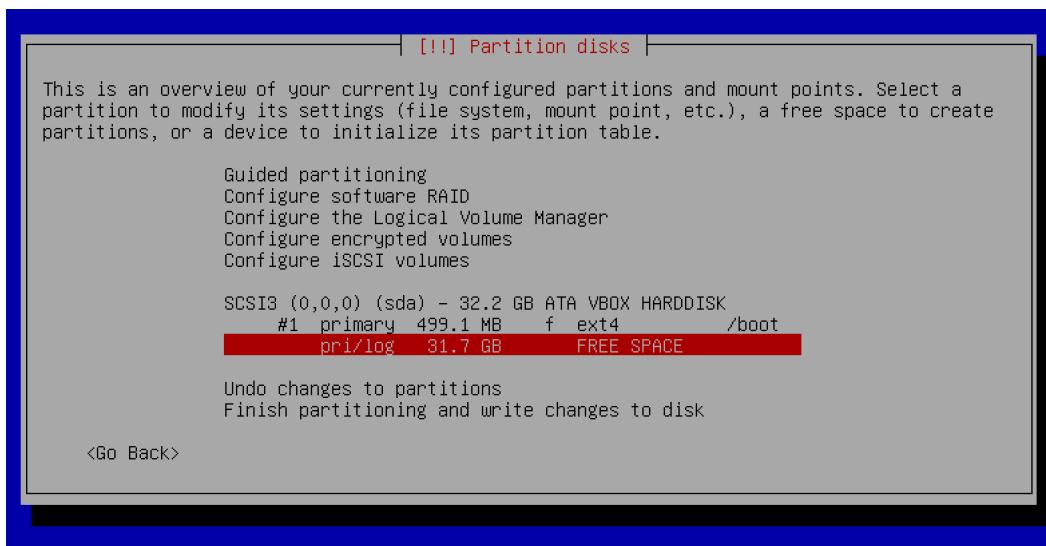
10 ° Escogemos boot como el punto de montaje de nuestra partición.



11 ° Terminamos de configurar la partición actual.

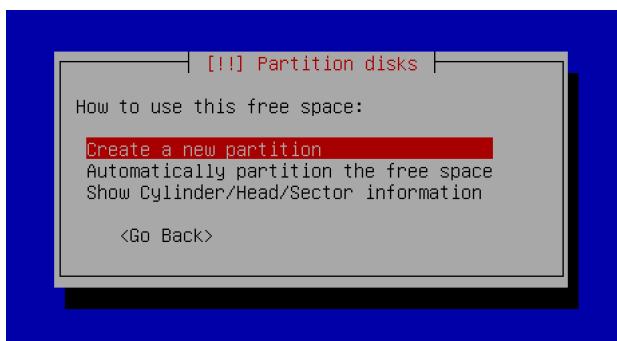


12. Una vez hemos completado el paso anterior ya nos debe aparecer la partición. Ahora debemos crear una partición lógica con todo el espacio disponible del disco, que no tenga punto de montaje y que este encriptada. Para ello seleccionamos el espacio libre donde queremos crearla.

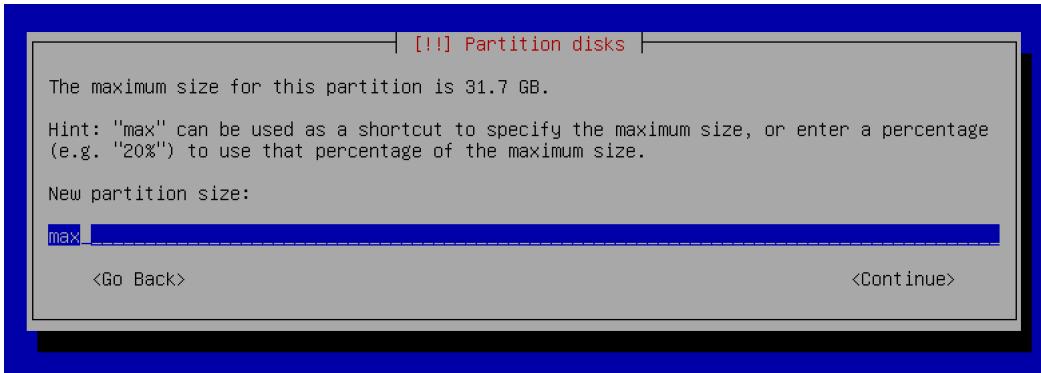


```
# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda       8:0     0 30.8G  0 disk
|__sda1    8:1     0 500M  0 part  /boot
|__sda2    8:2     0   1K  0 part
|__sda5    8:5     0 30.3G  0 part
|   __sda5_crypt 254:0   0 30.3G  0 crypt
|   |__LVMGroup-root 254:1   0 10G  0 lvm   /
|   |__LVMGroup-swap 254:2   0 2.3G  0 lvm  [SWAP]
|   |__LVMGroup-home 254:3   0 5G   0 lvm  /home
|   |__LVMGroup-var  254:4   0 3G   0 lvm  /var
|   |__LVMGroup-srv  254:5   0 3G   0 lvm  /srv
|   |__LVMGroup-tmp  254:6   0 3G   0 lvm  /tmp
|   |__LVMGroup-var--log 254:7   0 4G   0 lvm  /var/log
sr0      11:0     1 1024M 0 rom
```

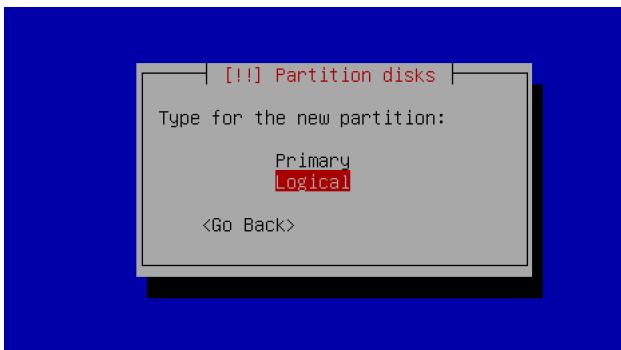
13. Creamos nueva partición.



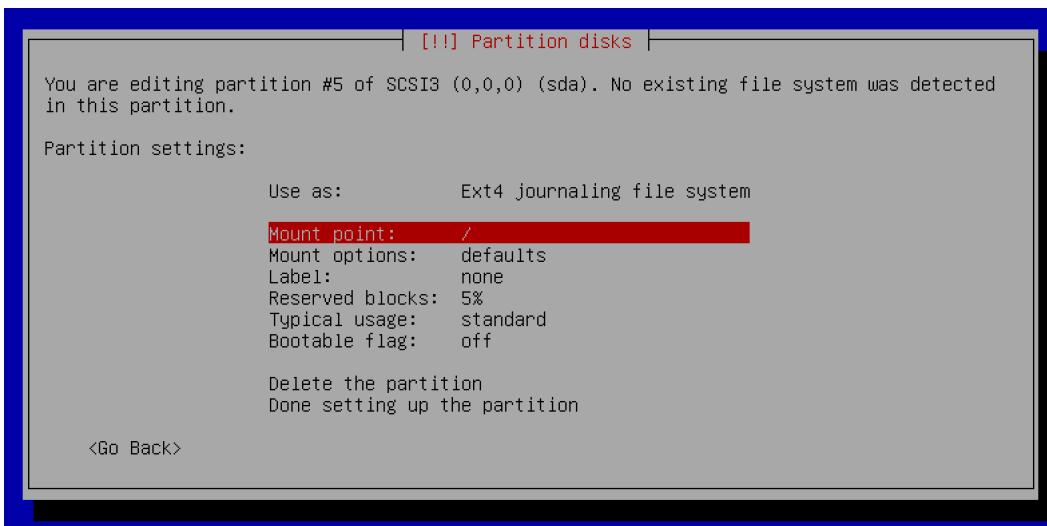
14. Seleccionamos el tamaño máximo.



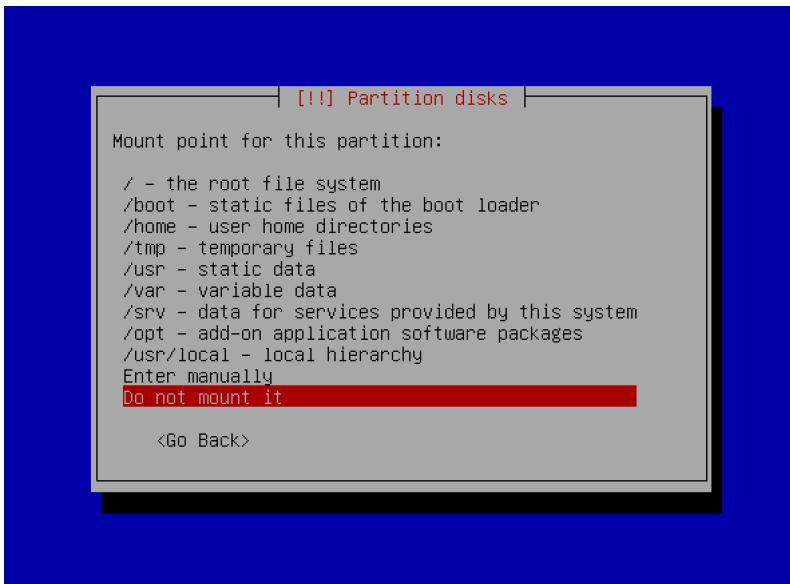
15 ° Seleccionamos el tipo de particion, en este caso lógica.



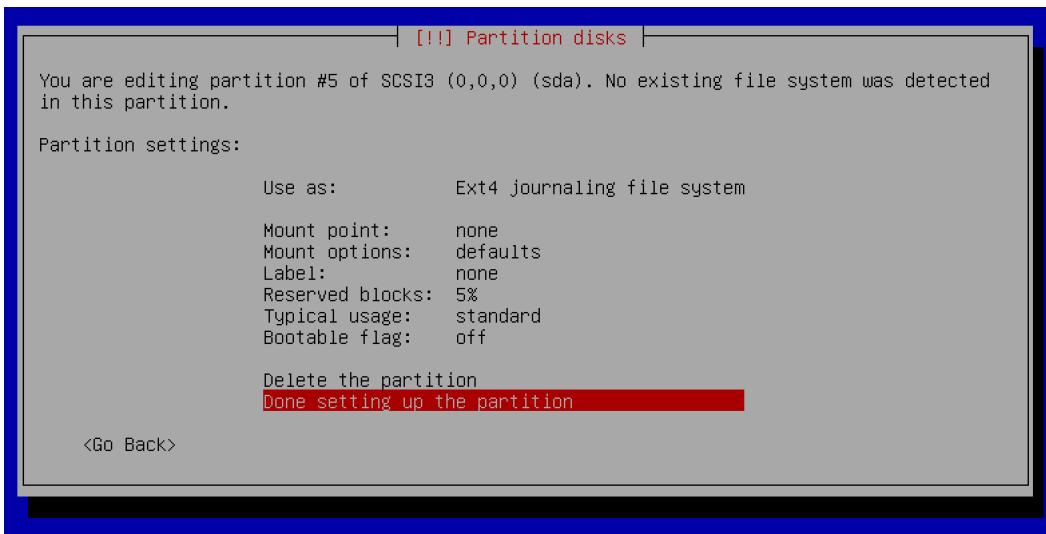
16 ° Modificaremos el punto de montaje.



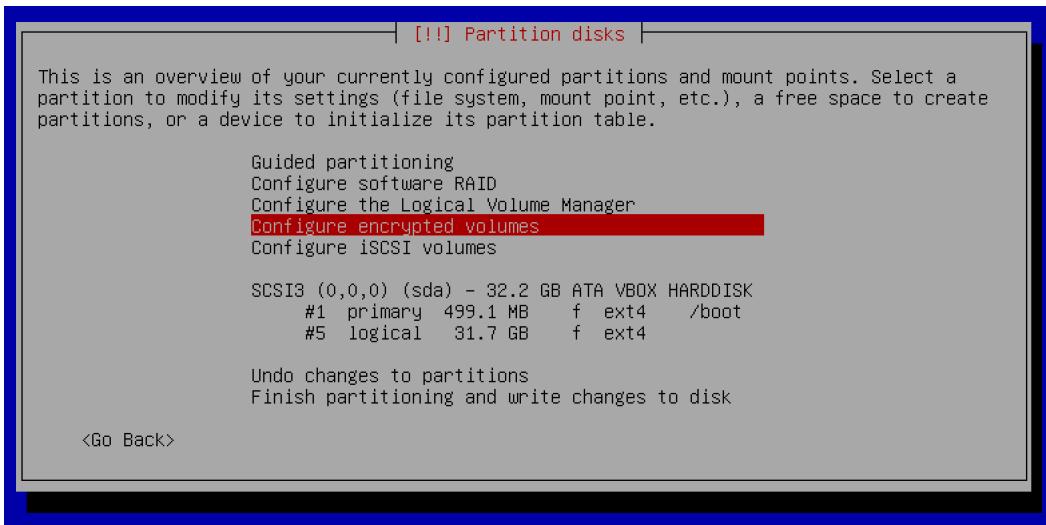
17 ° Escogeremos la opción de no montarlo.



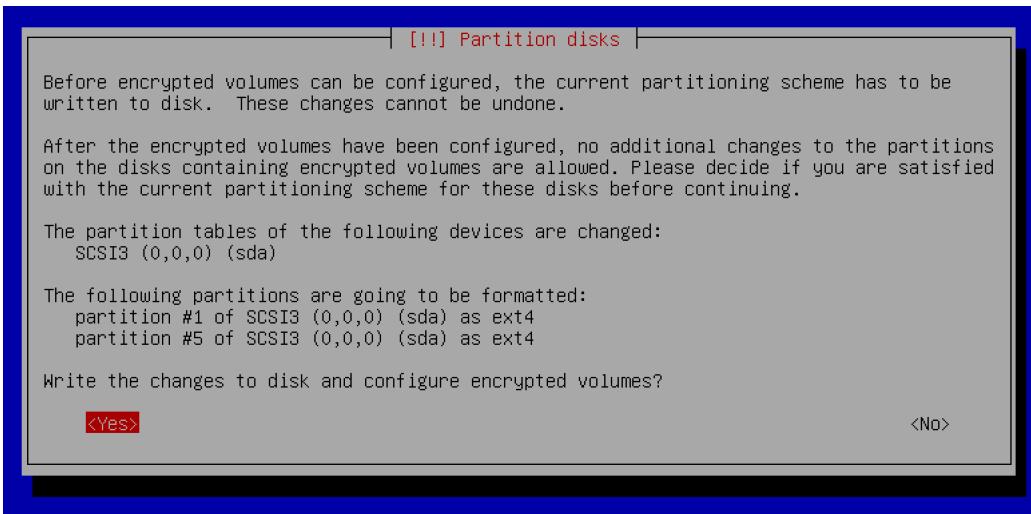
18 ° Terminamos de configurar la partición actual.



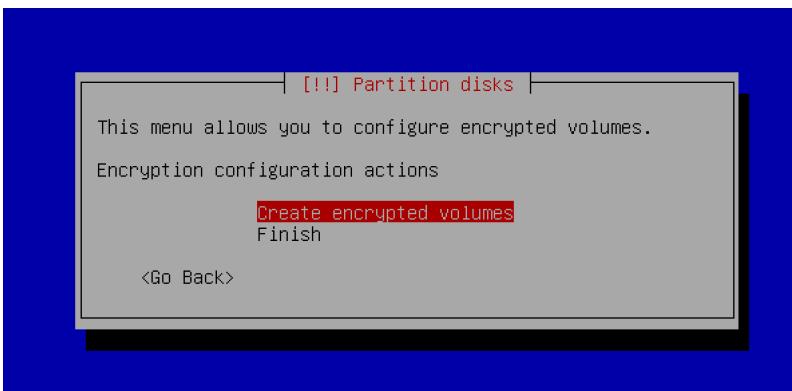
19 ° Configuraremos volúmenes encriptados. Para así poder encriptar nuestra partición.



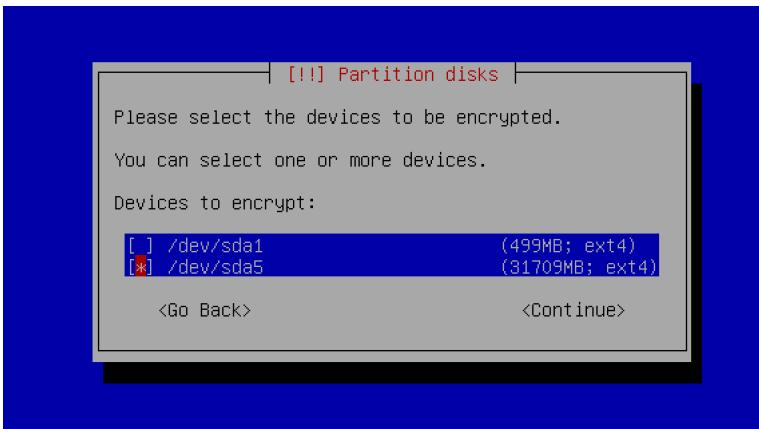
20 ° Aceptamos el mensaje de confirmación.



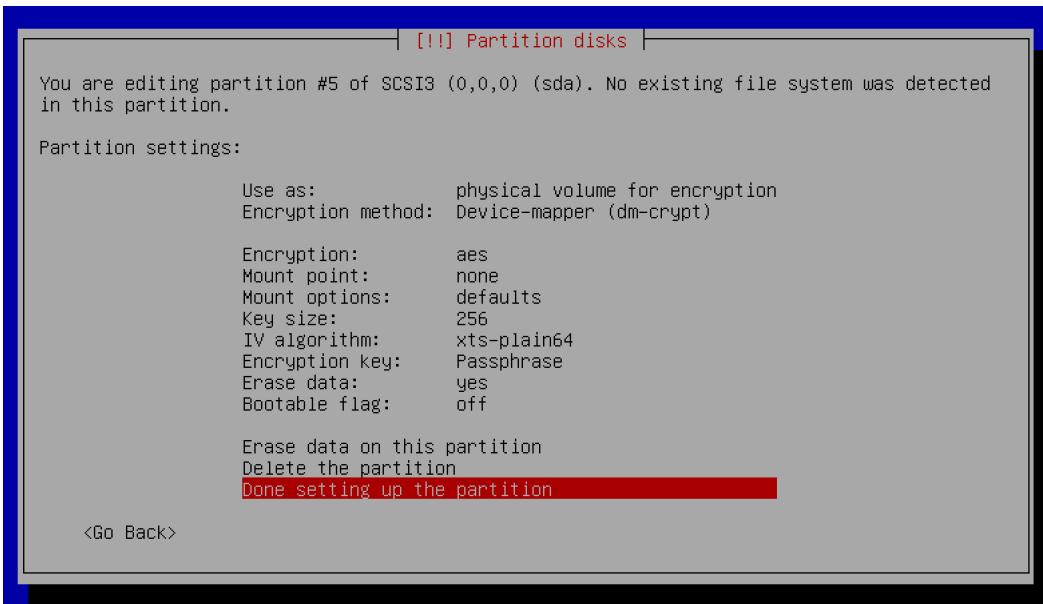
21 ° Creamos los volúmenes encriptados.



22 ° Seleccionamos en que partición queremos realizar la encriptación.



23 ° Terminamos de configurar la partición actual.



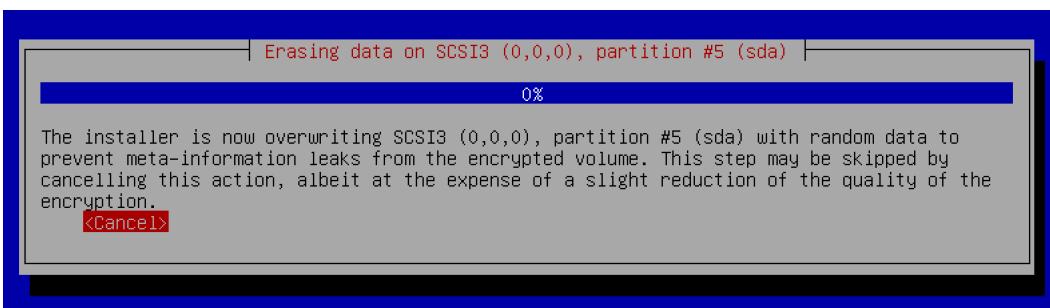
24º Finalizamos ya que no queremos crear mas volúmenes encriptados.



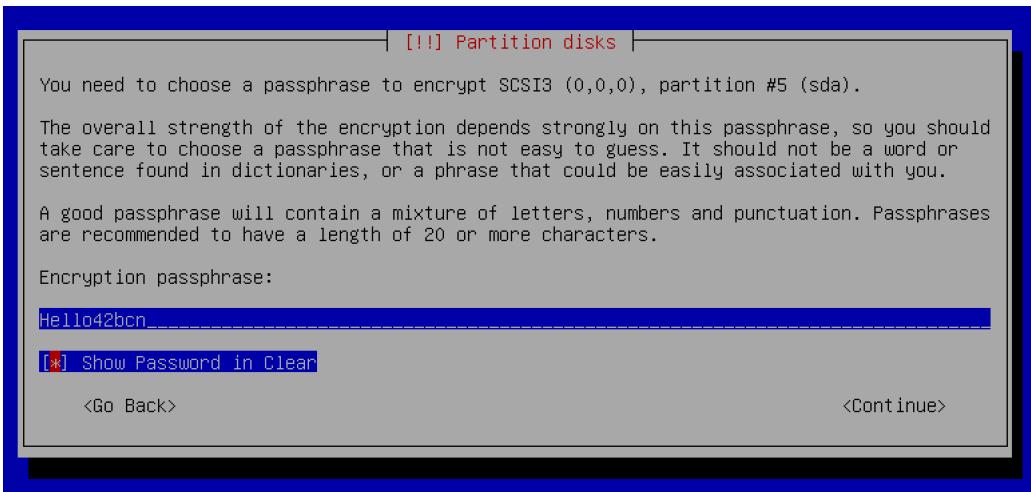
25º Aceptamos el mensaje de confirmación. Nos comenta que que se encriptara todo lo que hay dentro de la partición y que no debe tardar m en terminar.



26º Nos da igual si tarda mucho o poco , le damos a cancel ya que no hay nada que encriptar ya que la partición esta vacía.



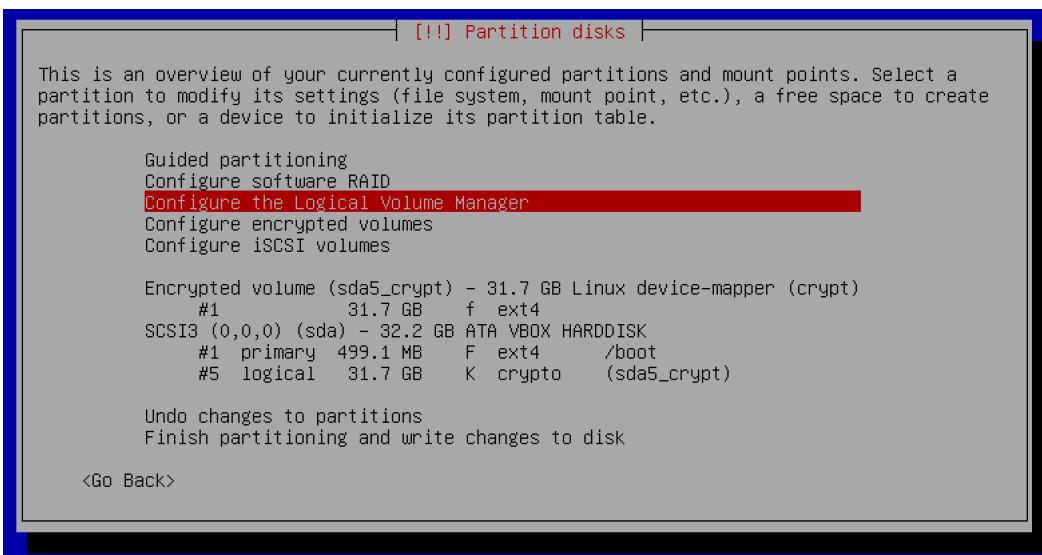
27º De nuevo deberemos poner una contraseña, esta vez será la frase de encriptación. Como te he comentado previamente deberás repetir el proceso y la debes anotar ya que será importante en un futuro.



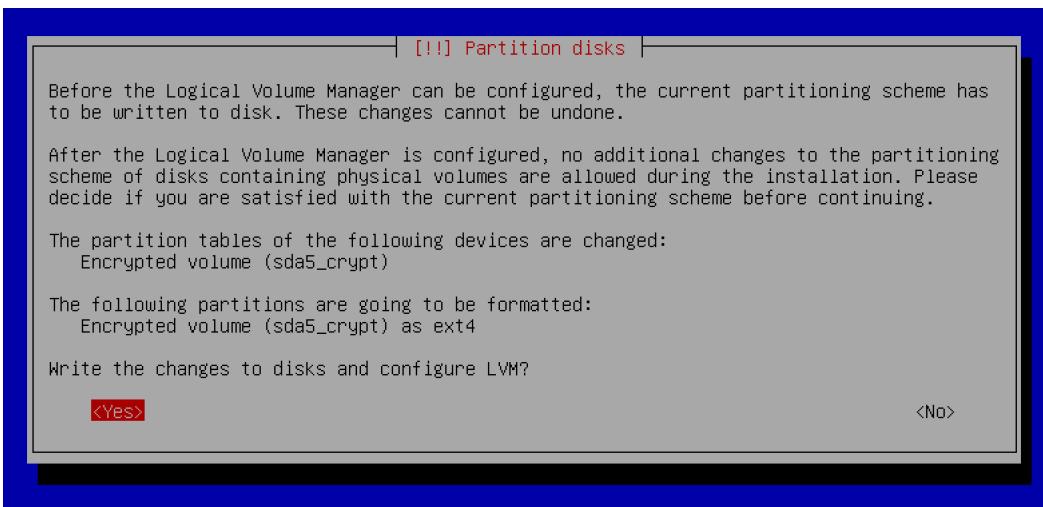
28 ° Repetimos la frase de encriptación.



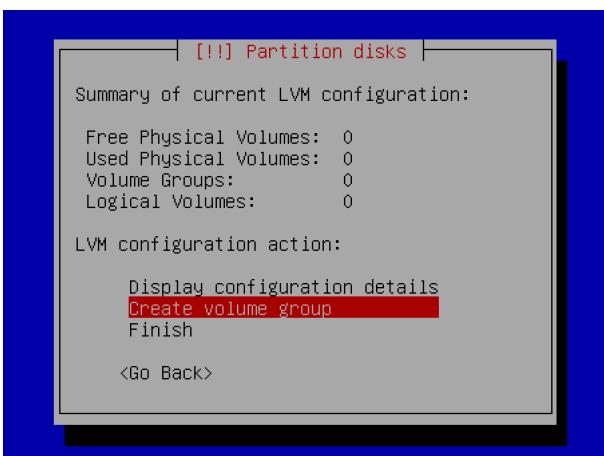
29 ° Configuraremos el gestor de volumenes logicos.



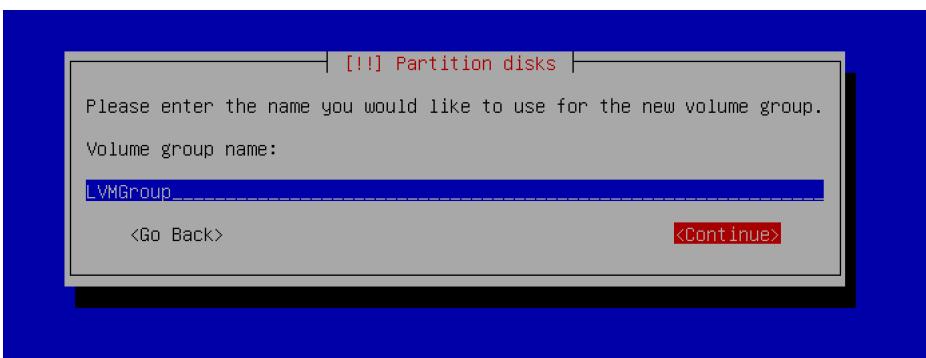
30 ° Aceptaremos en mensaje de confirmación ya que estamos de acuerdo con que se guarden los cambios en el disco.



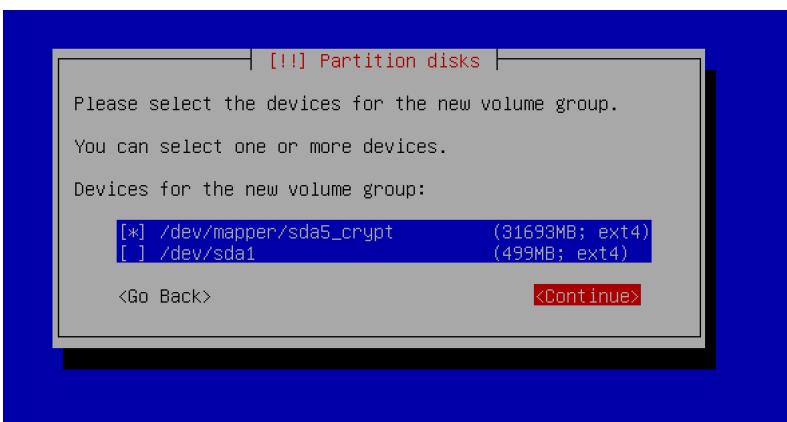
31 ◦ Crearemos un nuevo grupo de volumen. Los grupos de volúmenes agrupan particiones.



32 ◦ Introduciremos el nombre que queremos darle. `LVMGroup` tal y como indica el subject.

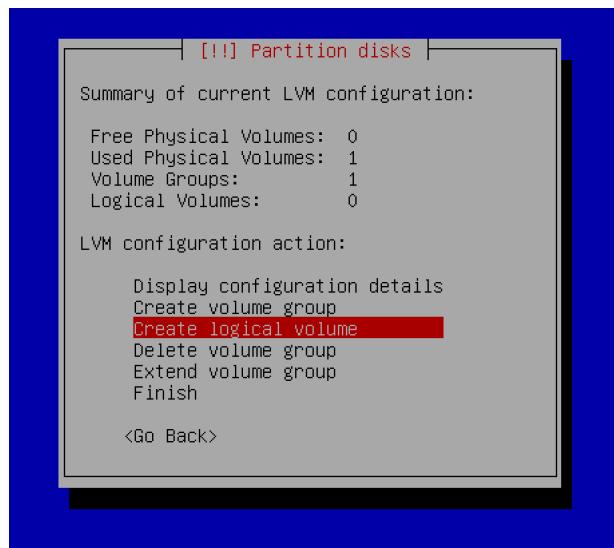


33 ◦ Seleccionaremos la partición donde queremos crear el grupo.

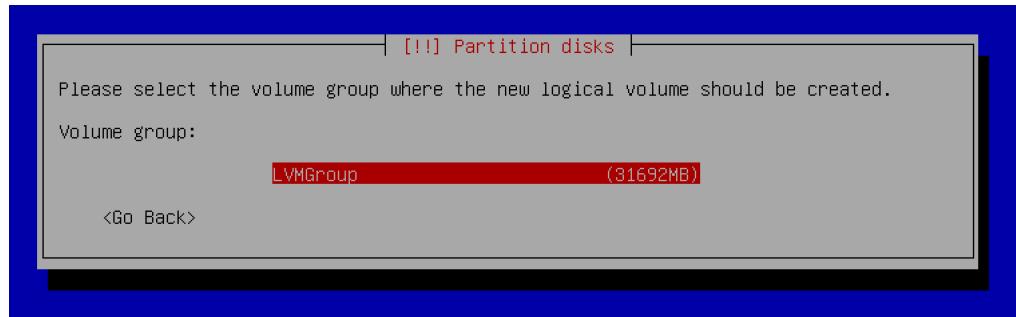


34 ° Ahora debemos crear todas las particiones lógicas. Al tener que repetir las mismas acciones varias veces hay capturas que no serán documentadas.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	30.8G	0	disk	
sda1	8:1	0	500M	0	part	/boot
sda2	8:2	0	1K	0	part	
sda5	8:5	0	30.3G	0	part	
└ sda5_crypt	254:0	0	30.3G	0	crypt	
└ LVMGroup-root	254:1	0	10G	0	lvm	/
└ LVMGroup-swap	254:2	0	2.3G	0	lvm	[SWAP]
└ LVMGroup-home	254:3	0	5G	0	lvm	/home
└ LVMGroup-var	254:4	0	3G	0	lvm	/var
└ LVMGroup-srv	254:5	0	3G	0	lvm	/srv
└ LVMGroup-tmp	254:6	0	3G	0	lvm	/tmp
└ LVMGroup-var--log	254:7	0	4G	0	lvm	/var/log



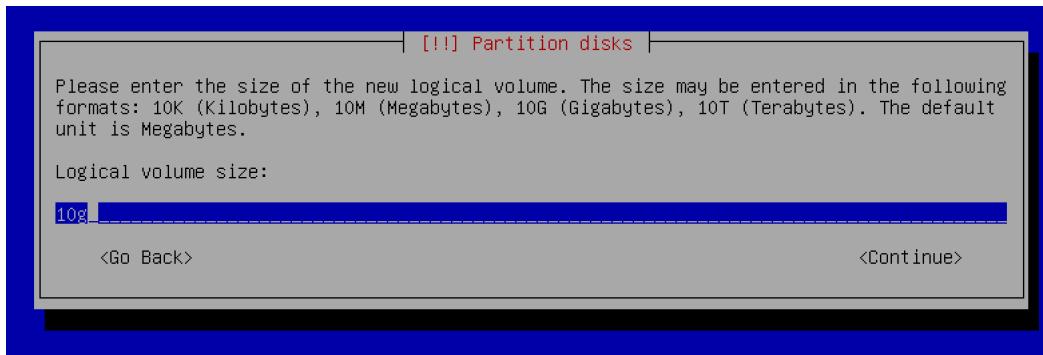
35 ° Empezaremos escogiendo el grupo donde queremos que se creen. Seleccionamos el único disponible (el que acabamos de crear).



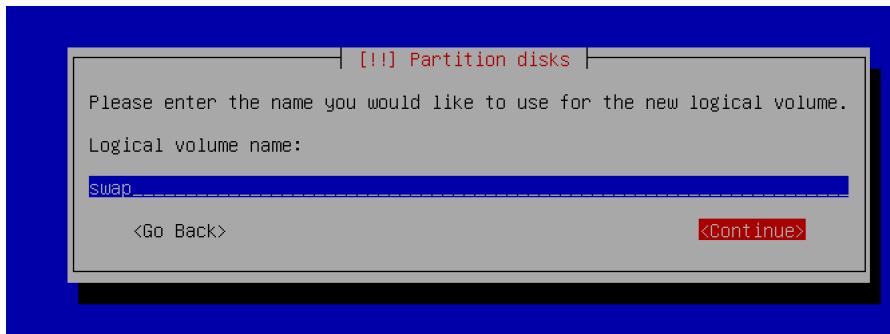
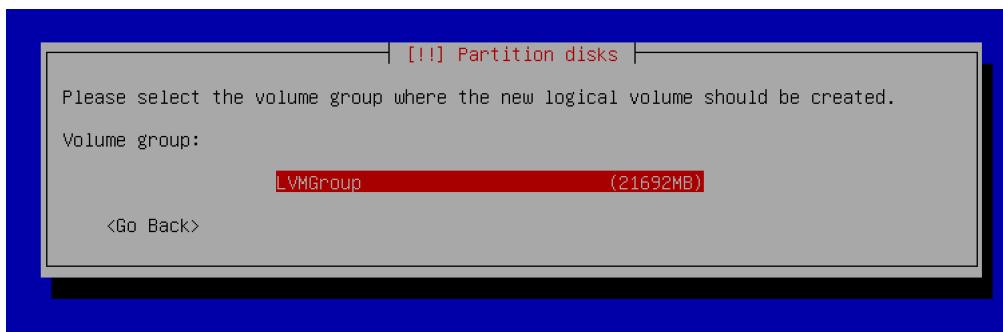
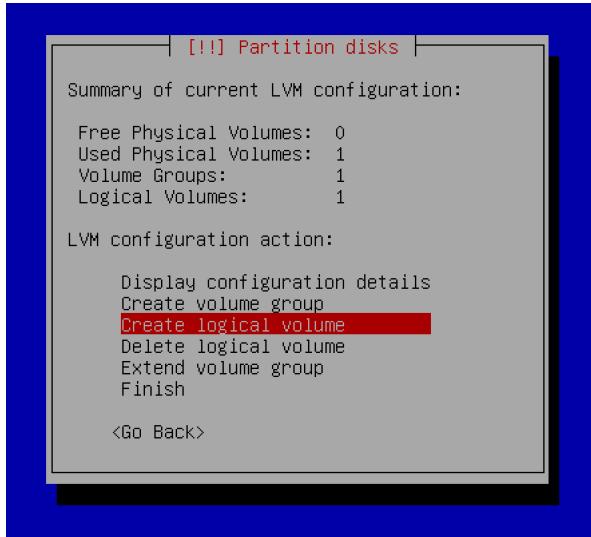
36 ° El orden de la creación de las unidades lógicas será el mismo que indica el subject así que empezaremos por root y acabaremos por var-log. Entonces seleccionaremos el nombre del volumen lógico.

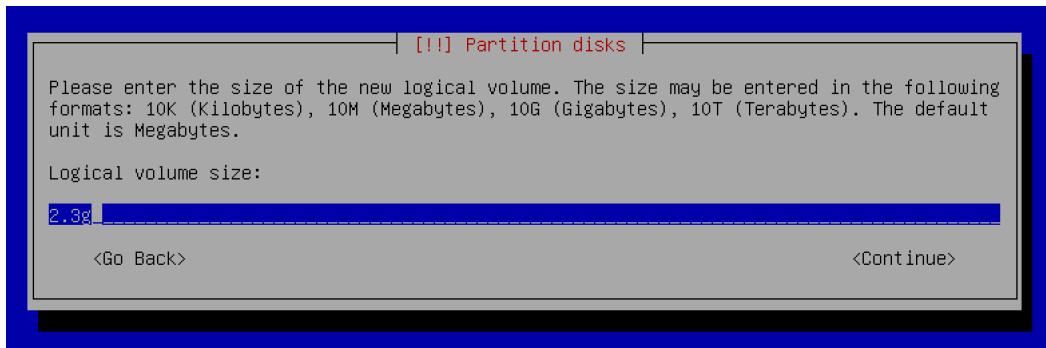


37 ° Tamaño como bien indica el subject será de 10g.

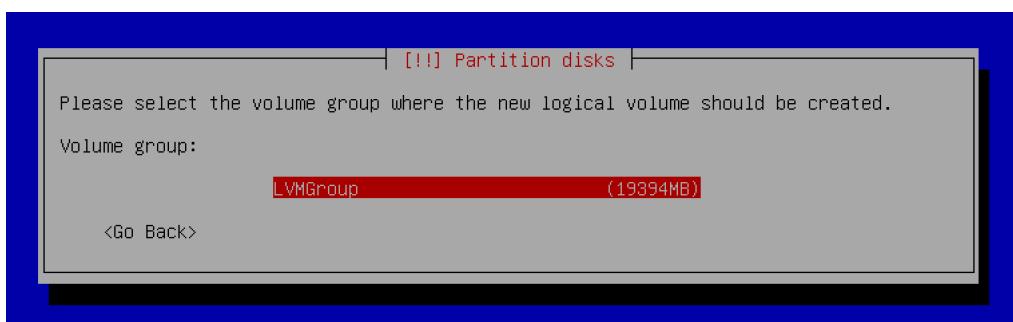
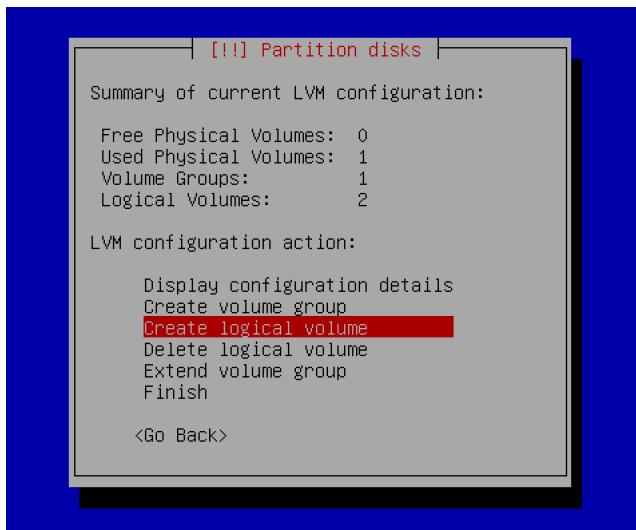


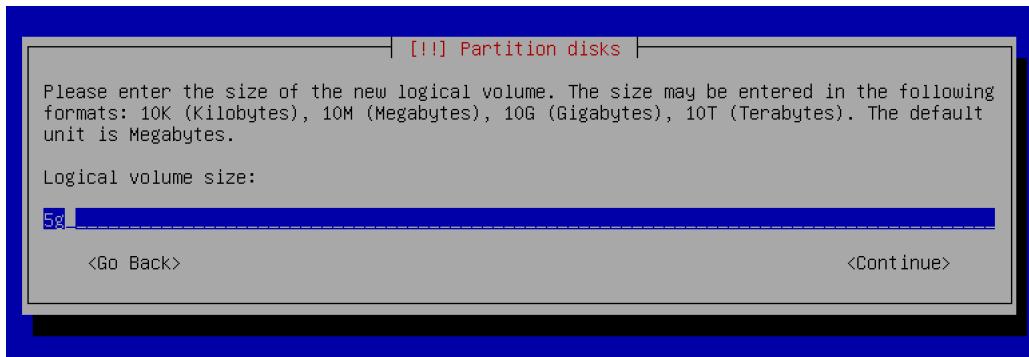
38 • Repetimos el proceso para `swap`. Solo cambiaremos el nombre y el tamaño.



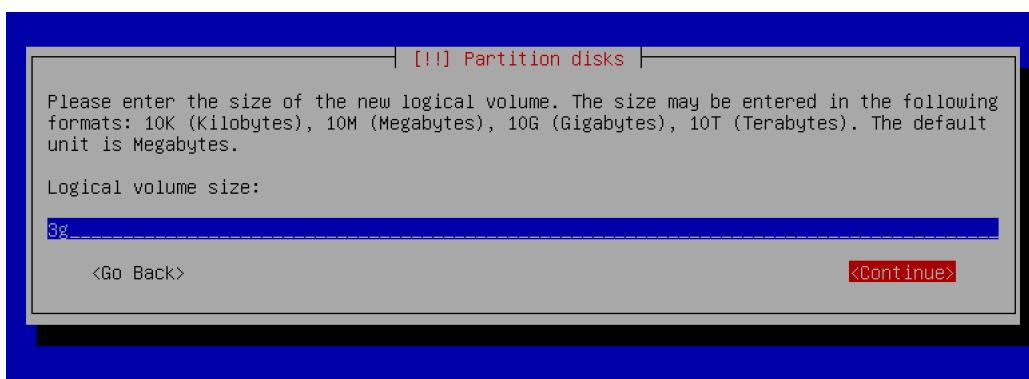
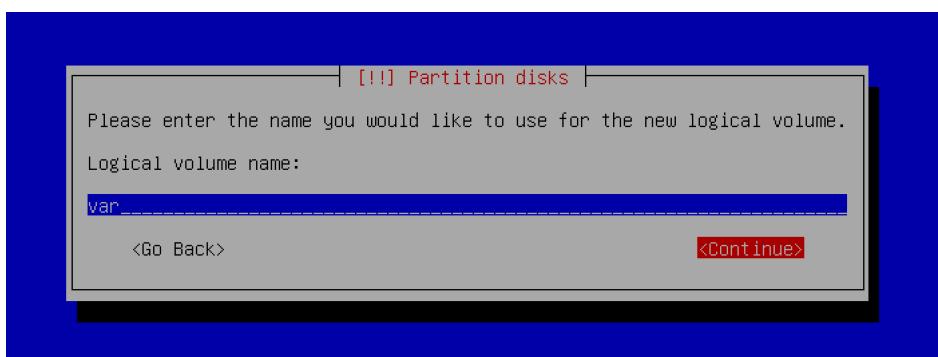
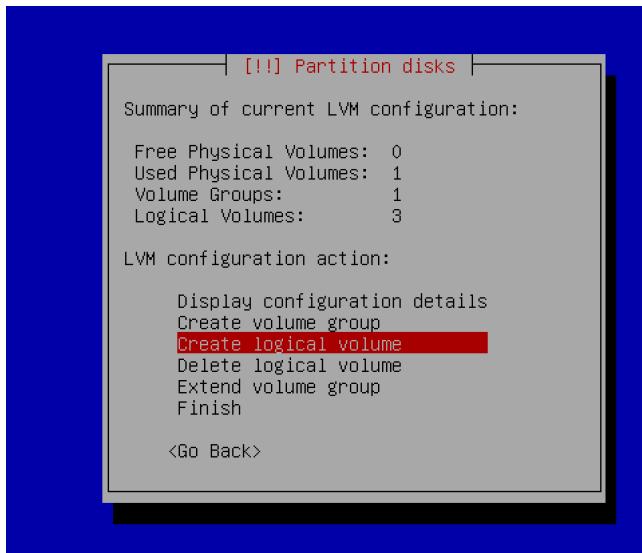


39. Repetimos el proceso para `home`. Solo cambiaremos el nombre y el tamaño.

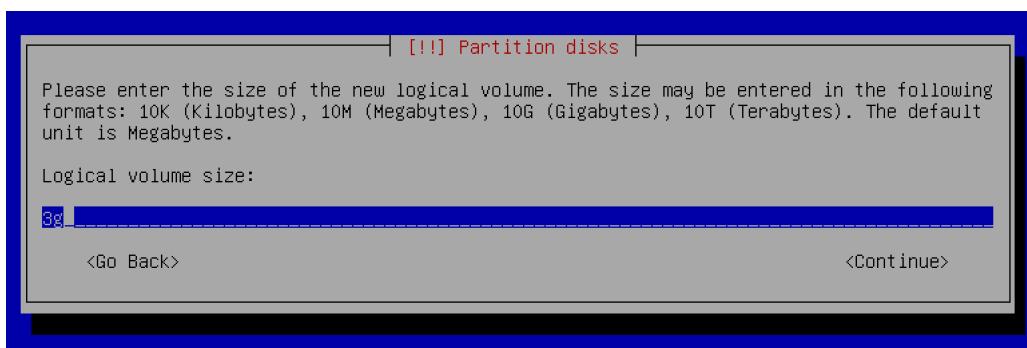
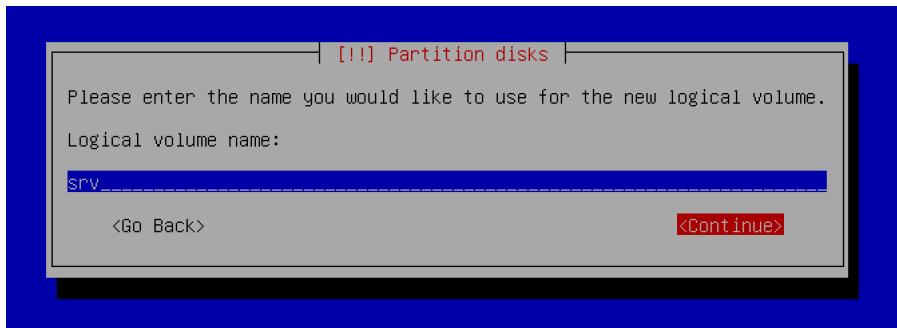
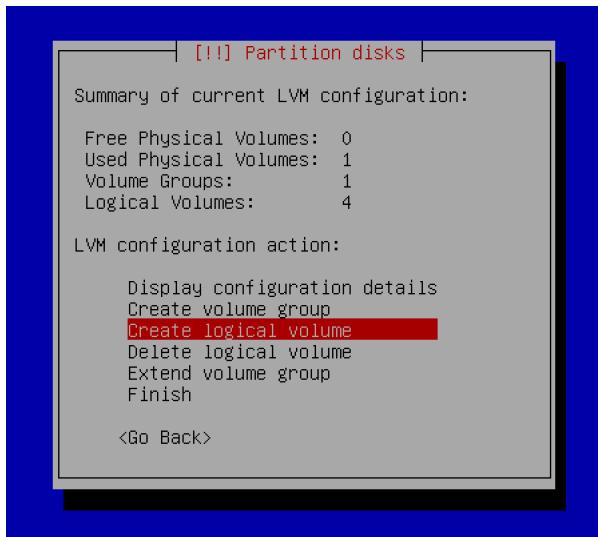




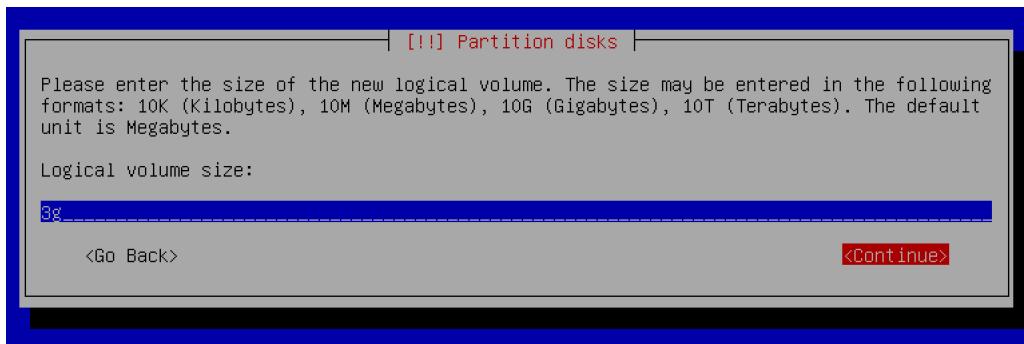
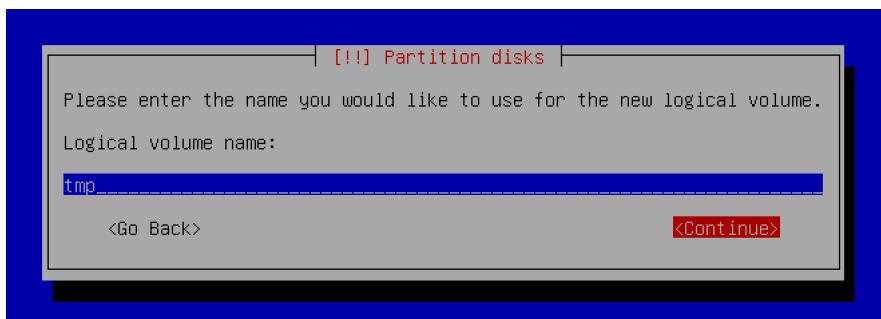
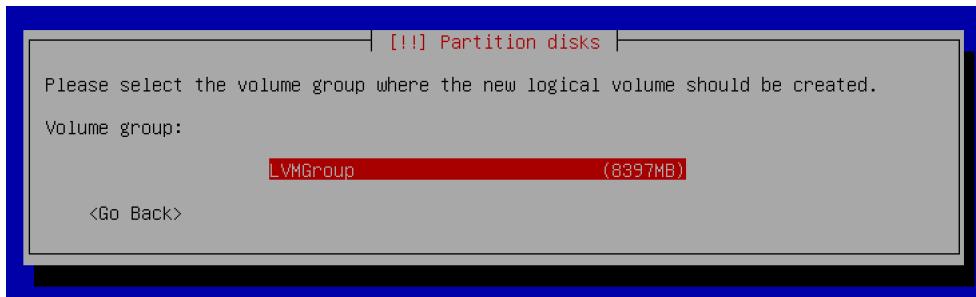
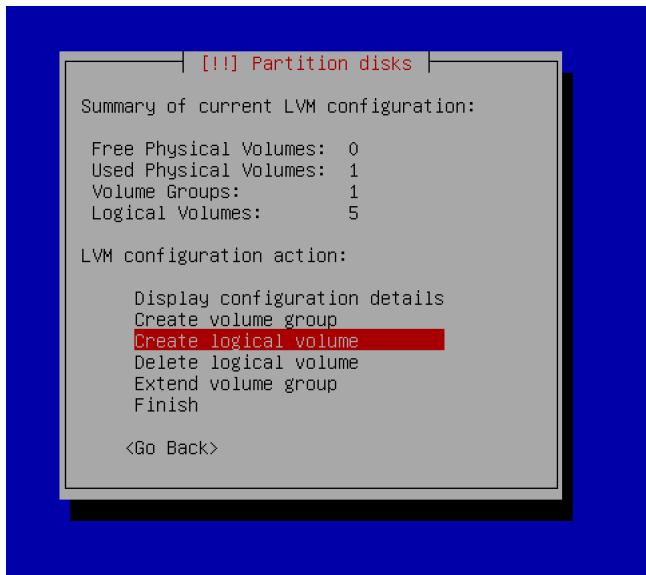
40 ° Repetimos el proceso para `var`. Solo cambiaremos el nombre y el tamaño.



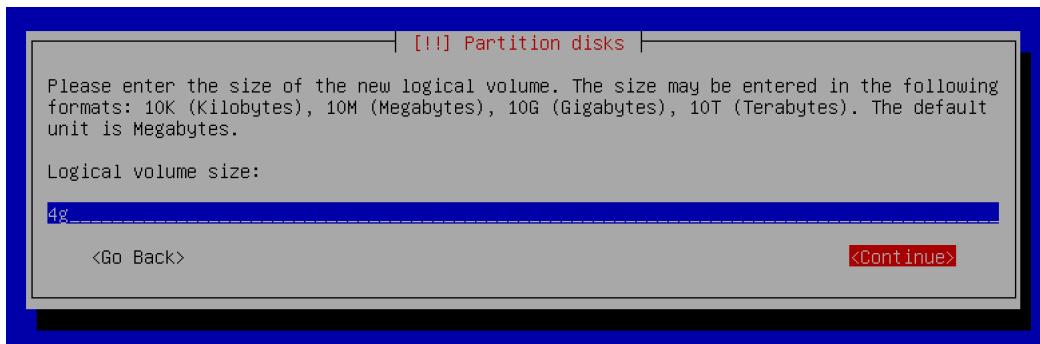
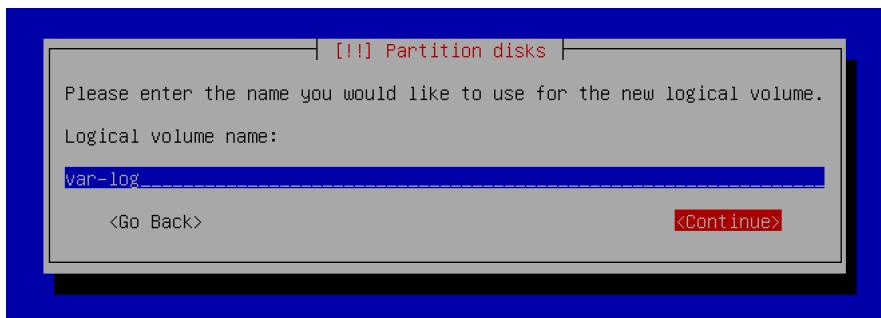
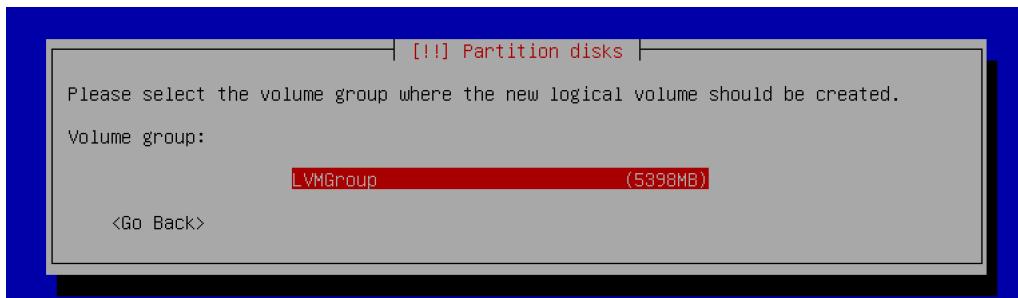
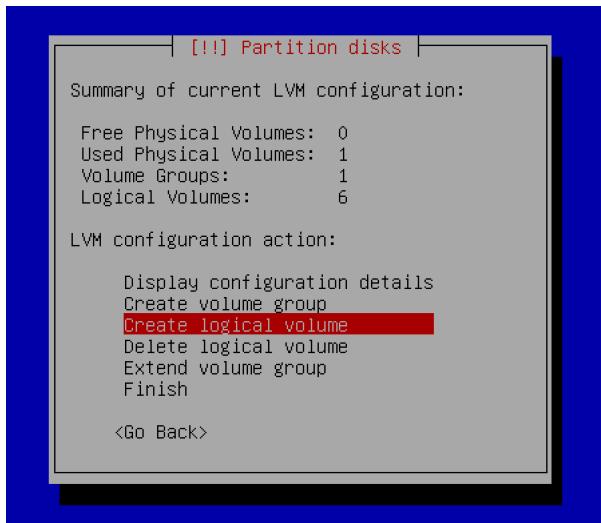
41 ° Repetimos el proceso para `srv`. Solo cambiaremos el nombre.



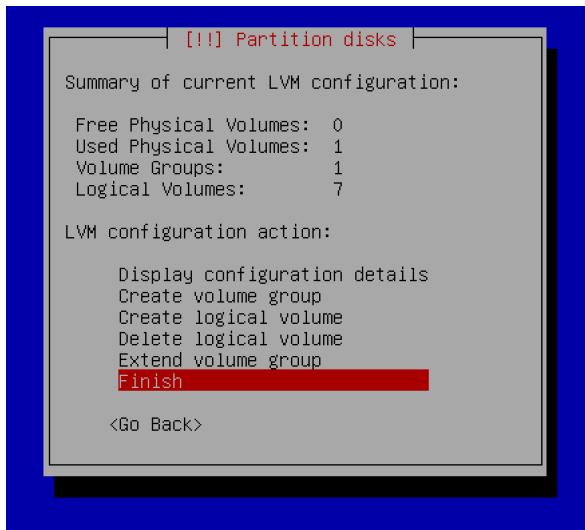
42 ° Repetimos el proceso para `tmp`. Solo cambiaremos el nombre.



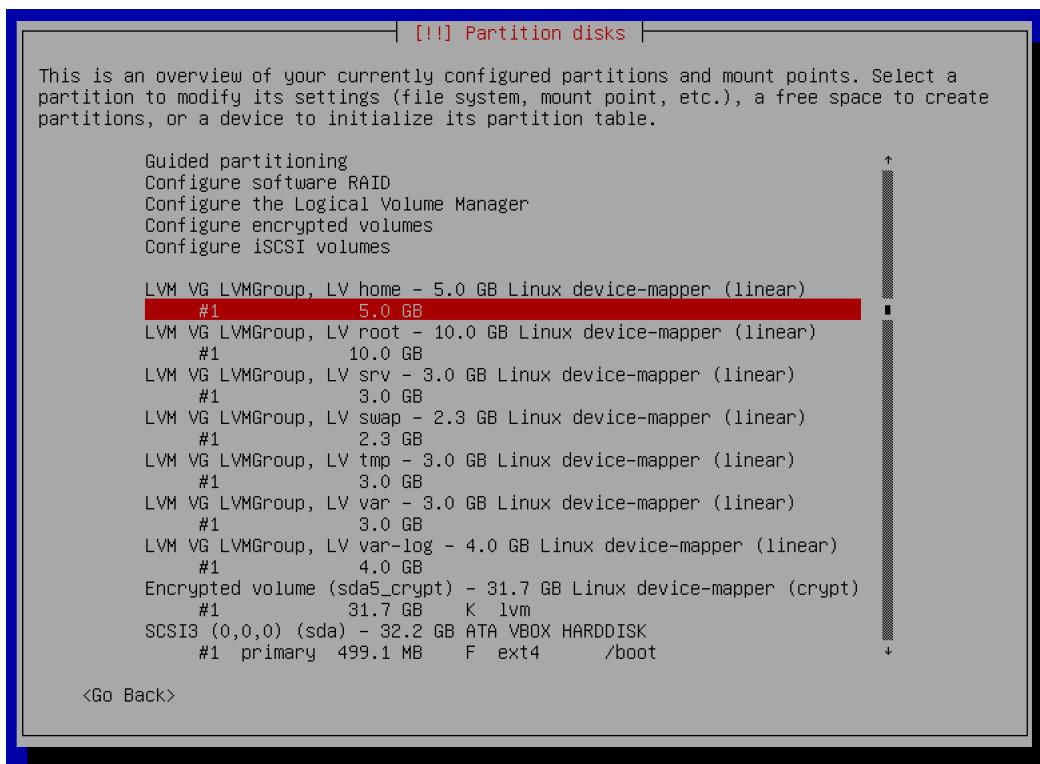
43 ° Por último repetimos el proceso para `var-log`. Solo cambiaremos el nombre y el tamaño.



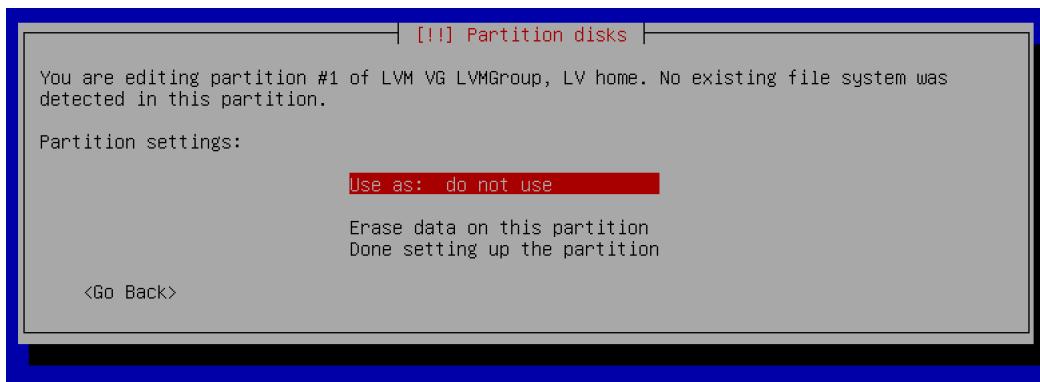
44 ◦ Una vez hayamos completado todos los pasos anteriores finalizaremos la configuración del gestor de volúmenes lógicos.



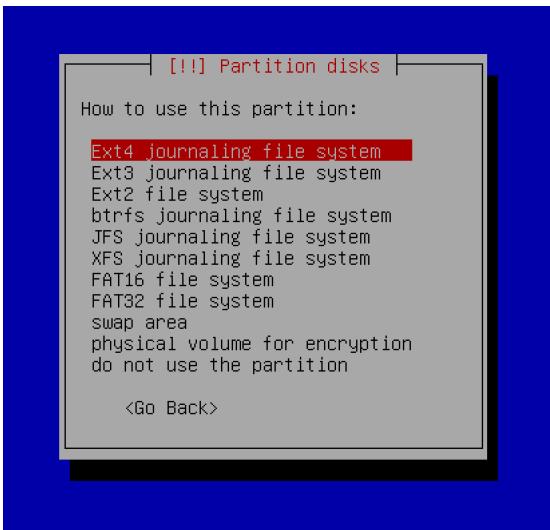
45 ° Ahora podemos observar como en el apartado donde nos muestran todas nuestras particiones y espacio libre ya aparecen todas las partitologicas que acabamos de crear. Bien , debemos configurar todas para seleccionar el sistema de archivos que queremos y el punto de montaje c indica el subject. De nuevo iremos por orden y seleccionaremos la primera que nos aparece que es `home` .



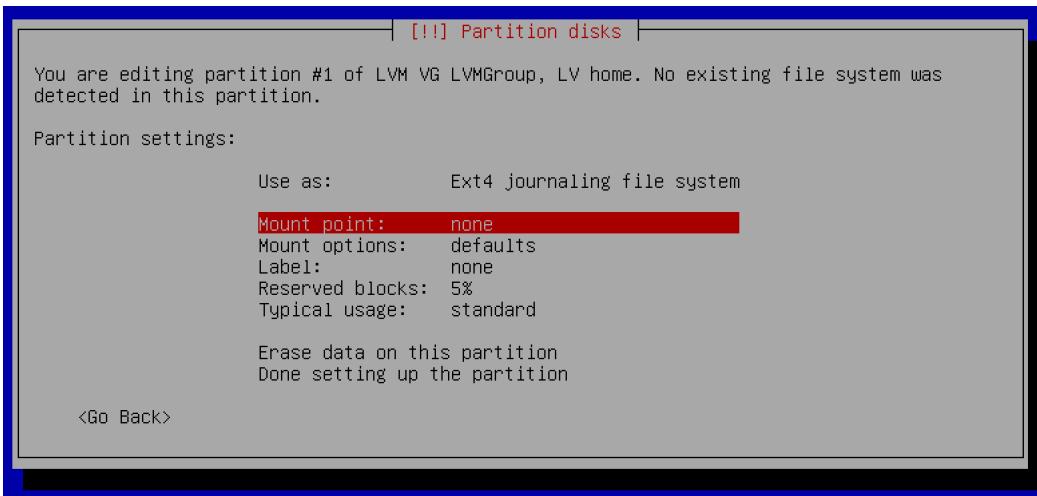
46 ° Nos muestra la configuración de la partición. Debemos escoger un sistema de ficheros ya que actualmente no tiene.



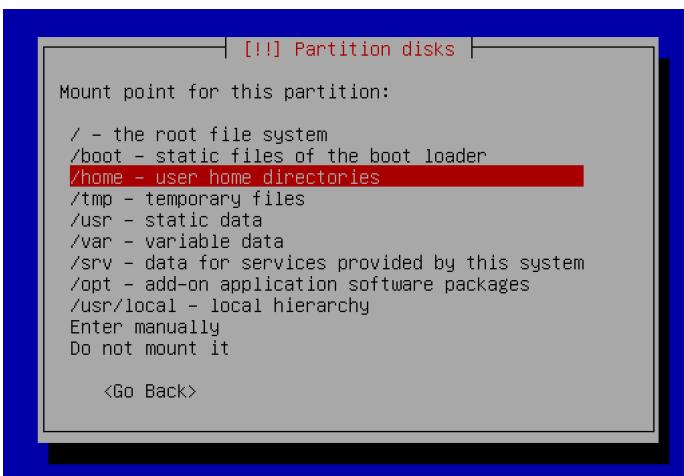
47 ° Escogemos el sistema de archivos Ext4, es el sistema de archivos más utilizado en distribuciones Linux.



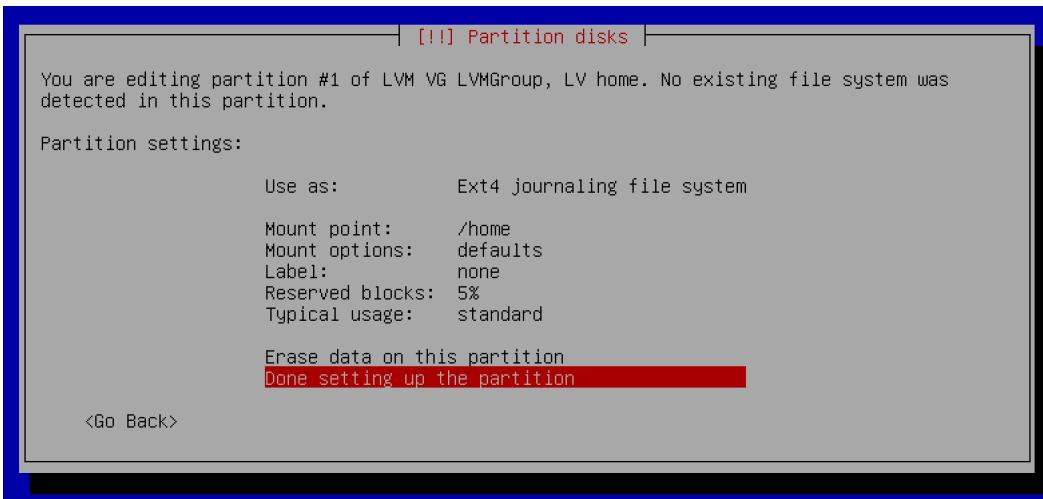
48 ° Ahora debemos seleccionar el punto de montaje.



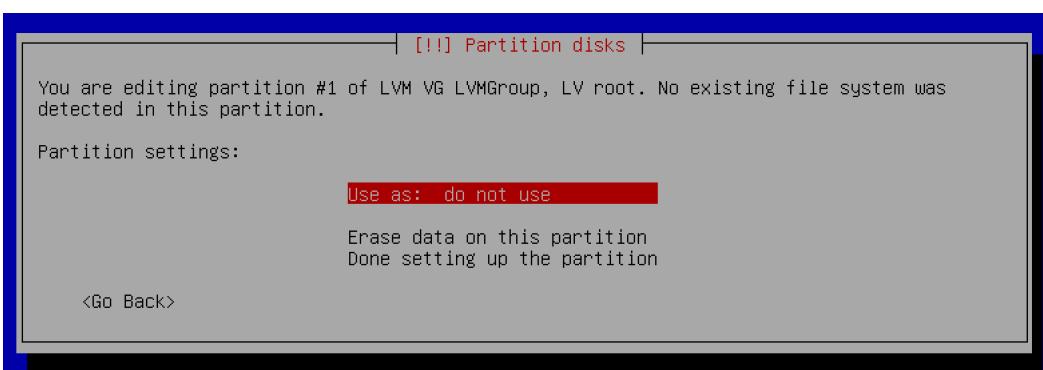
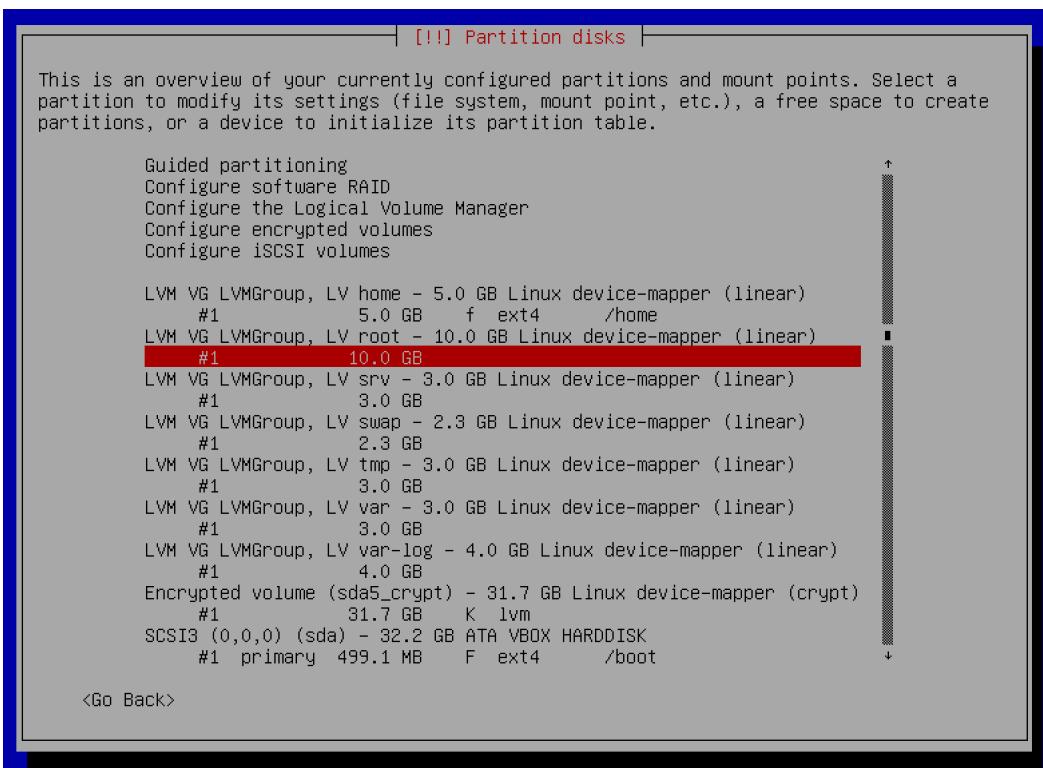
49 ° Seleccionamos `home` como bien indica el subject.

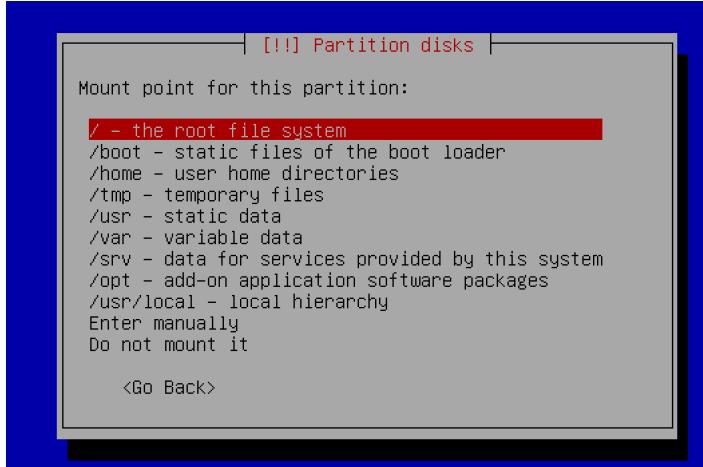
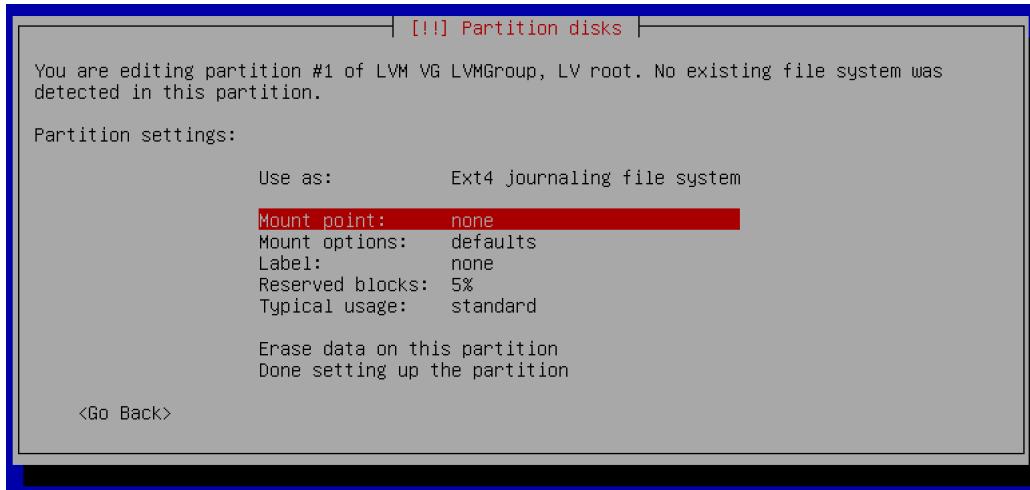
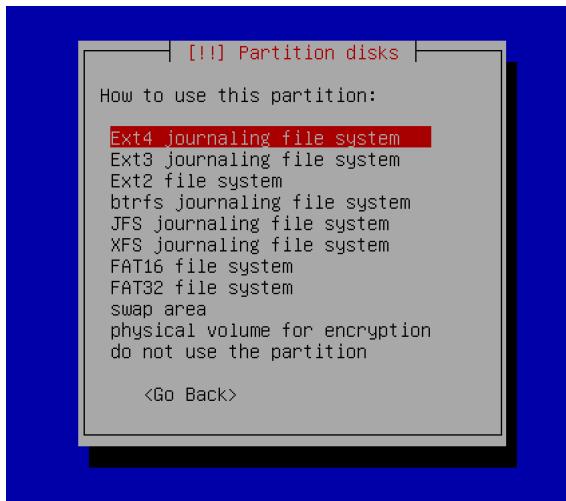


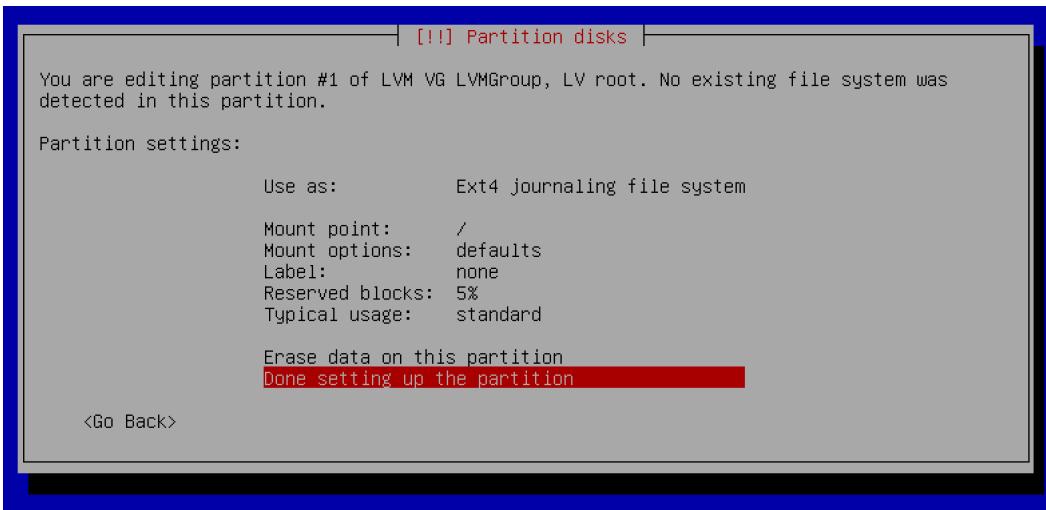
50 ° Una vez ya lo hemos seleccionado terminaremos la configuración de la partición.



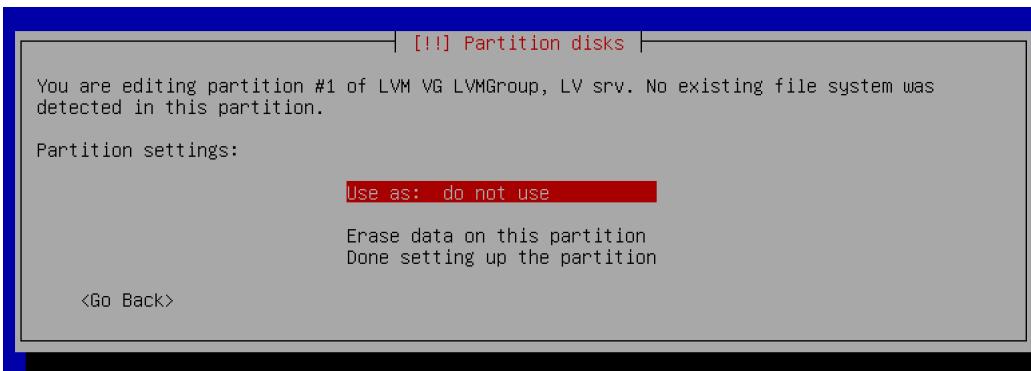
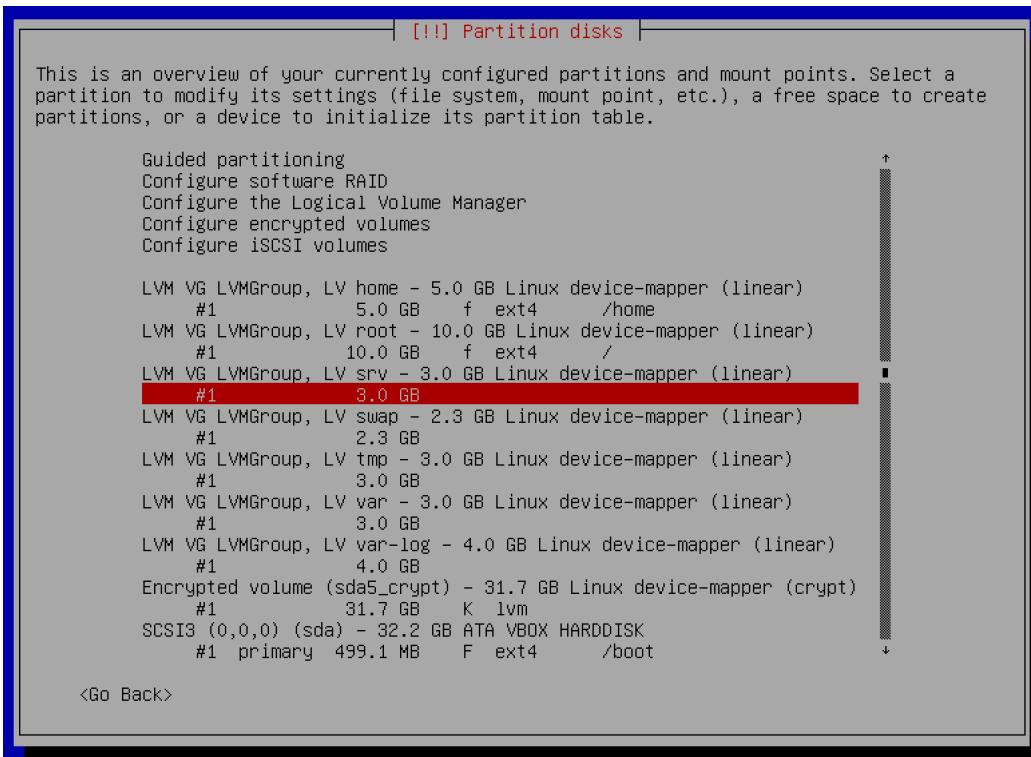
51 ° De nuevo estos pasos se pueden volver muy repetitivos así que no comentare mucho. Repetimos todo igual (excepto el punto de montaje) | `root`.

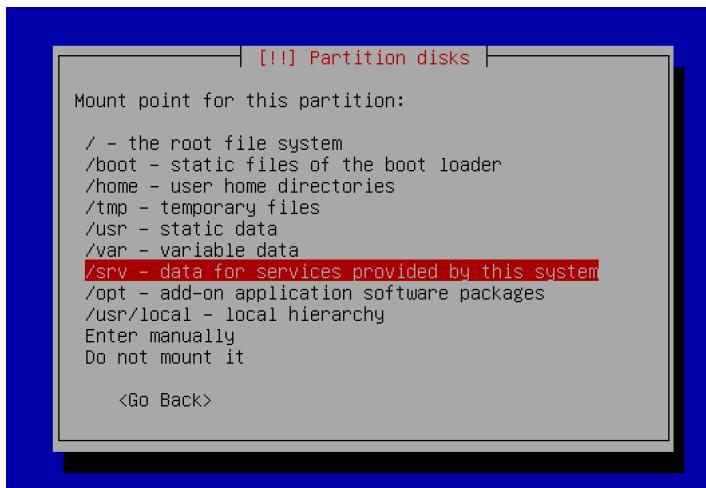
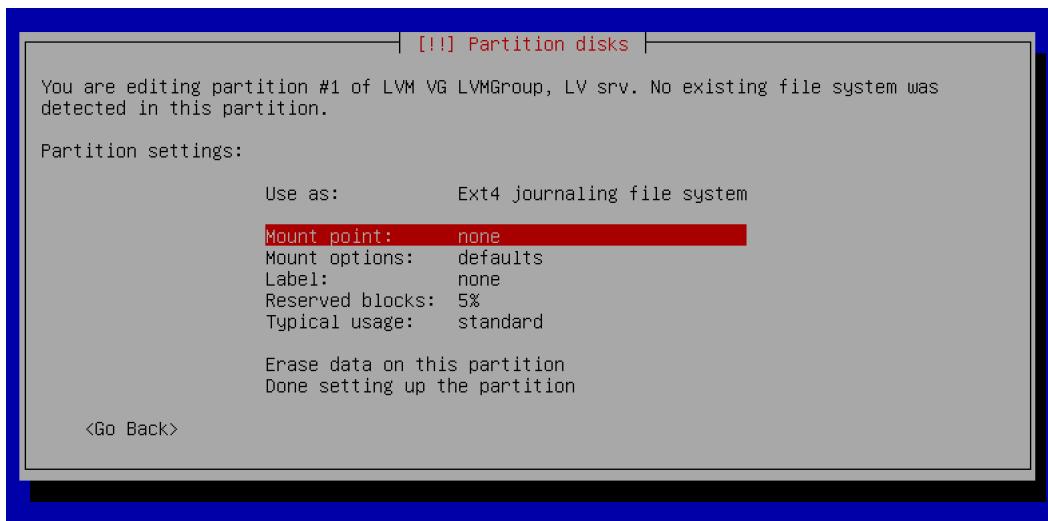
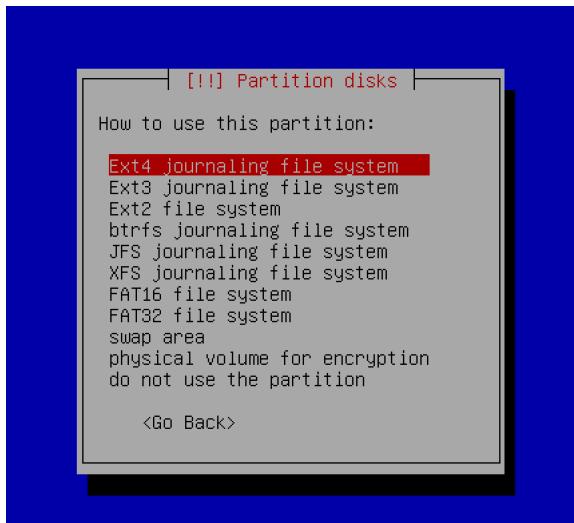


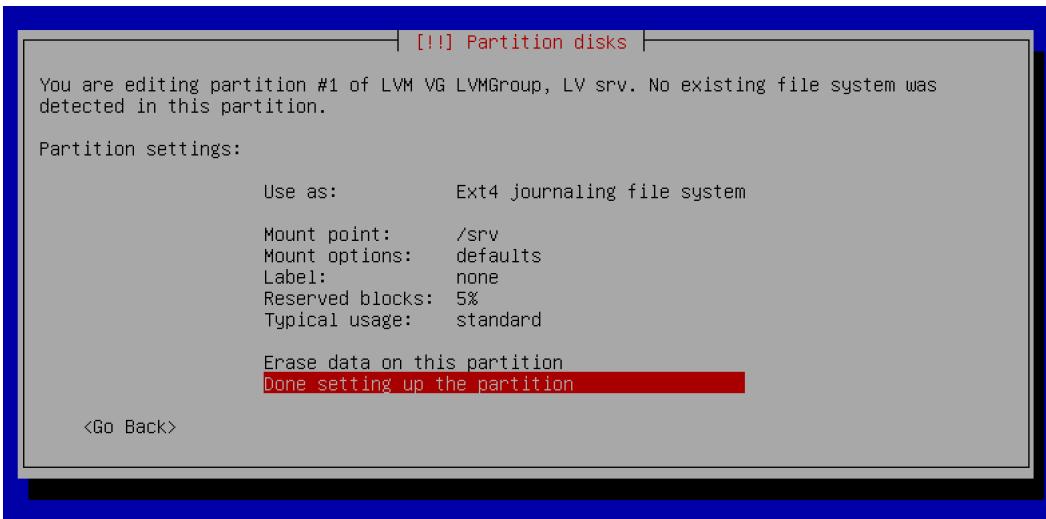




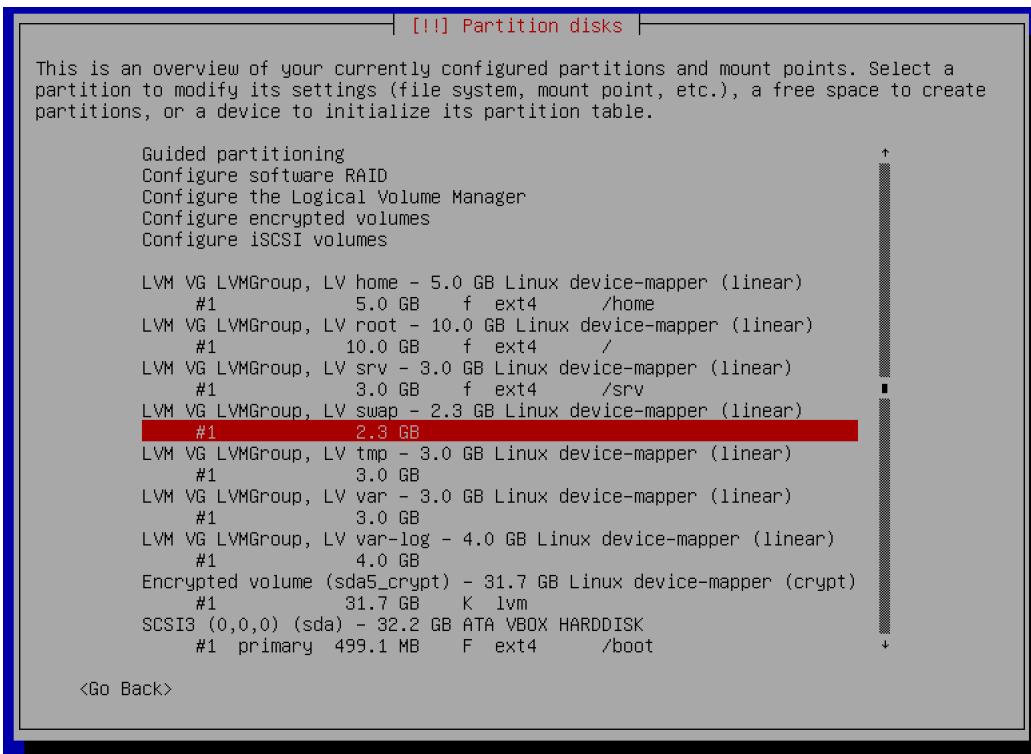
52. Repetimos el proceso para `srv` y cambiaremos el punto de montaje.



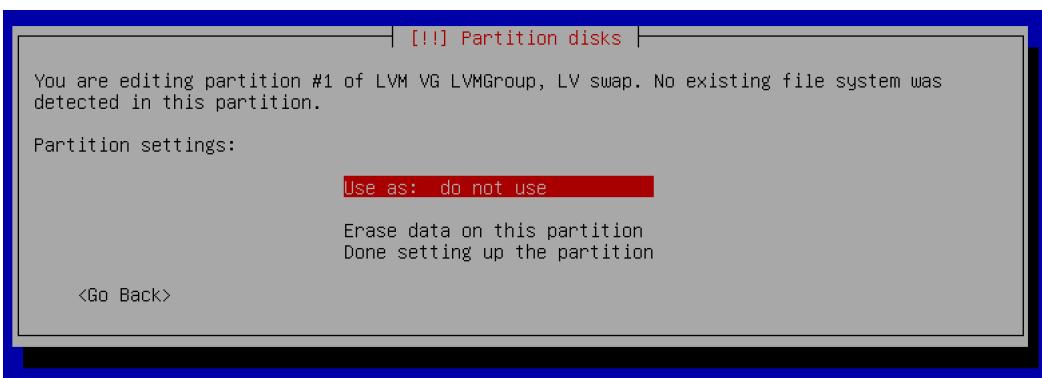




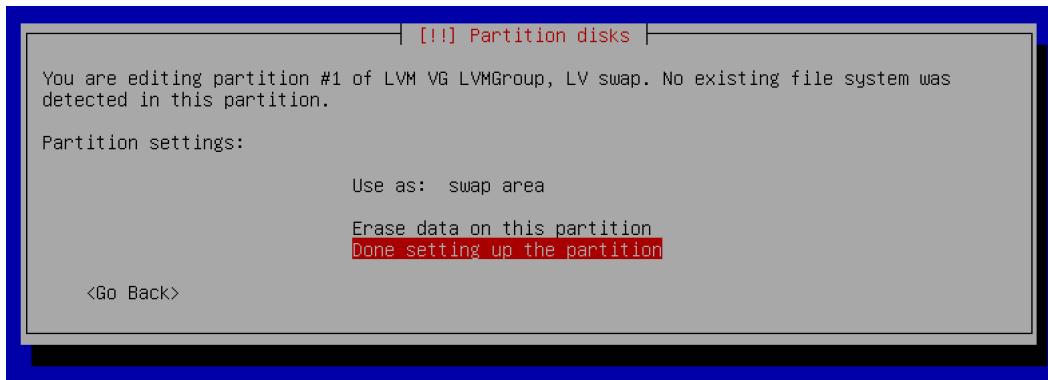
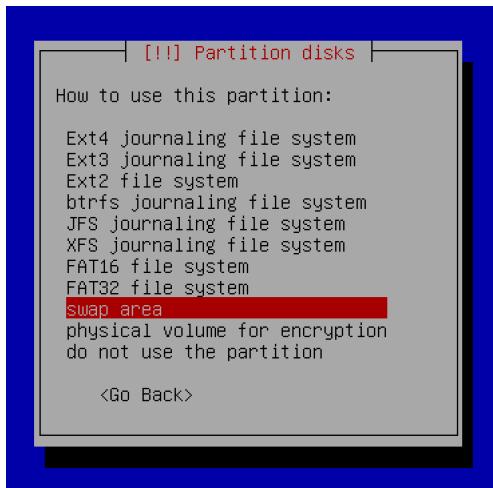
53. Para `swap` haremos una excepción ya el sistema de archivos será diferente. Seleccionamos `swap`.



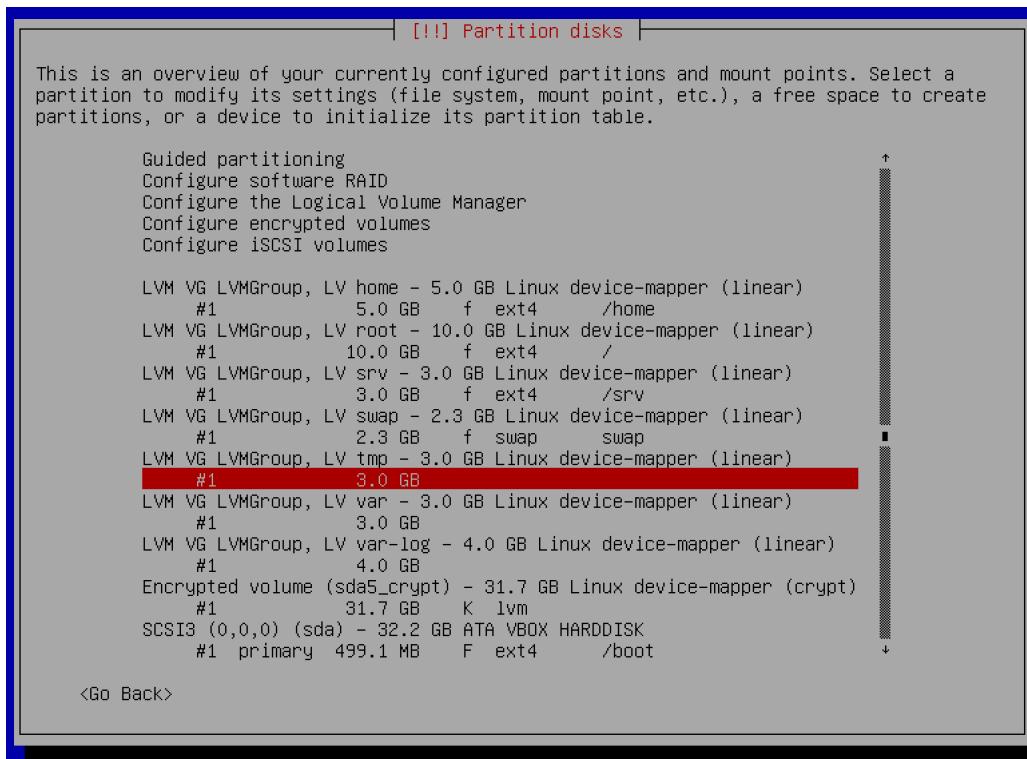
54. En el momento de seleccionar el sistema de archivos lo dejamos en `swap` area.

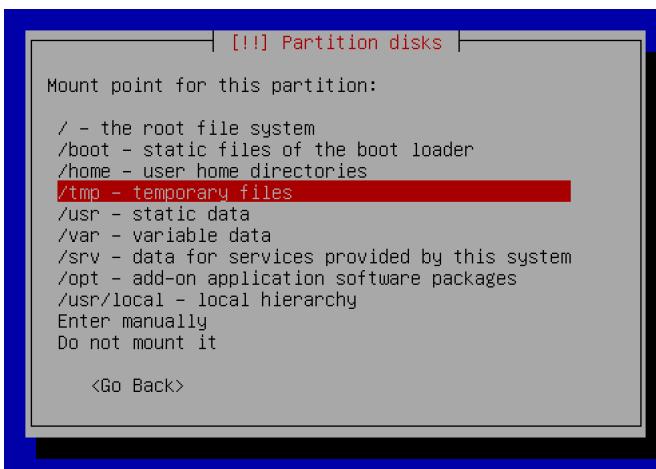
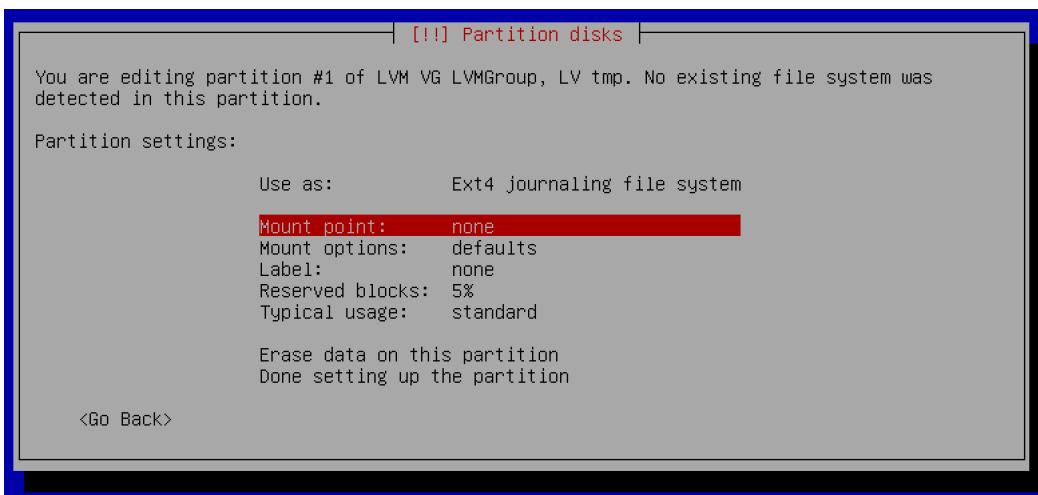
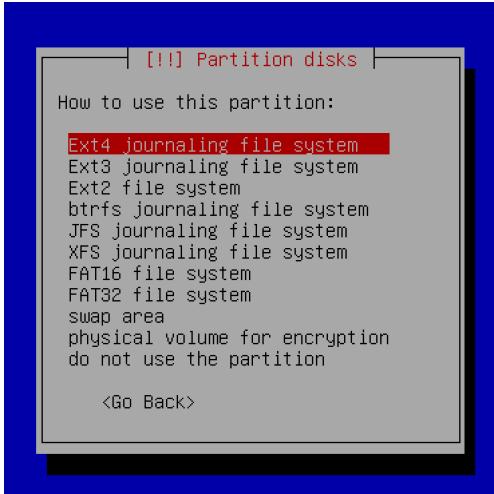


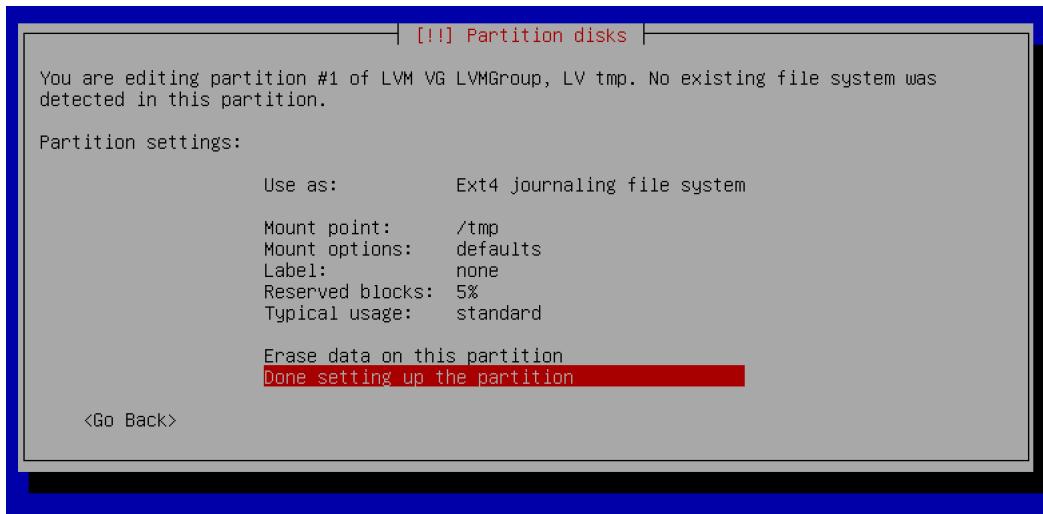
55. Una vez realizado el paso anterior terminaremos la configuración de la partición.



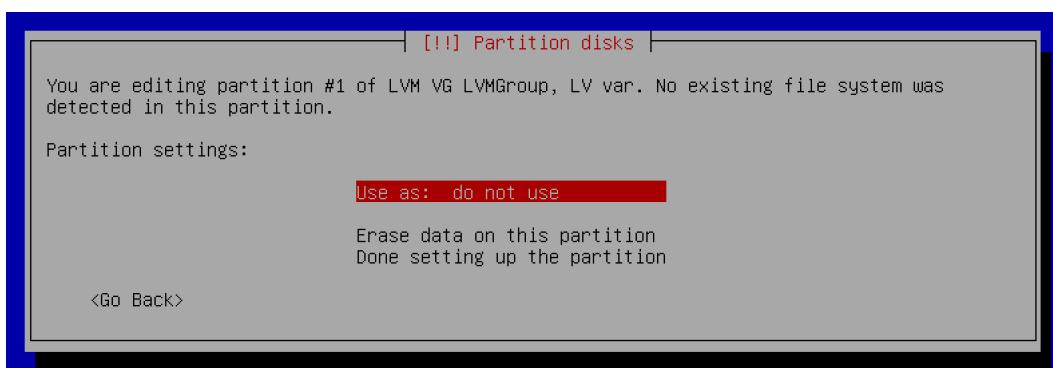
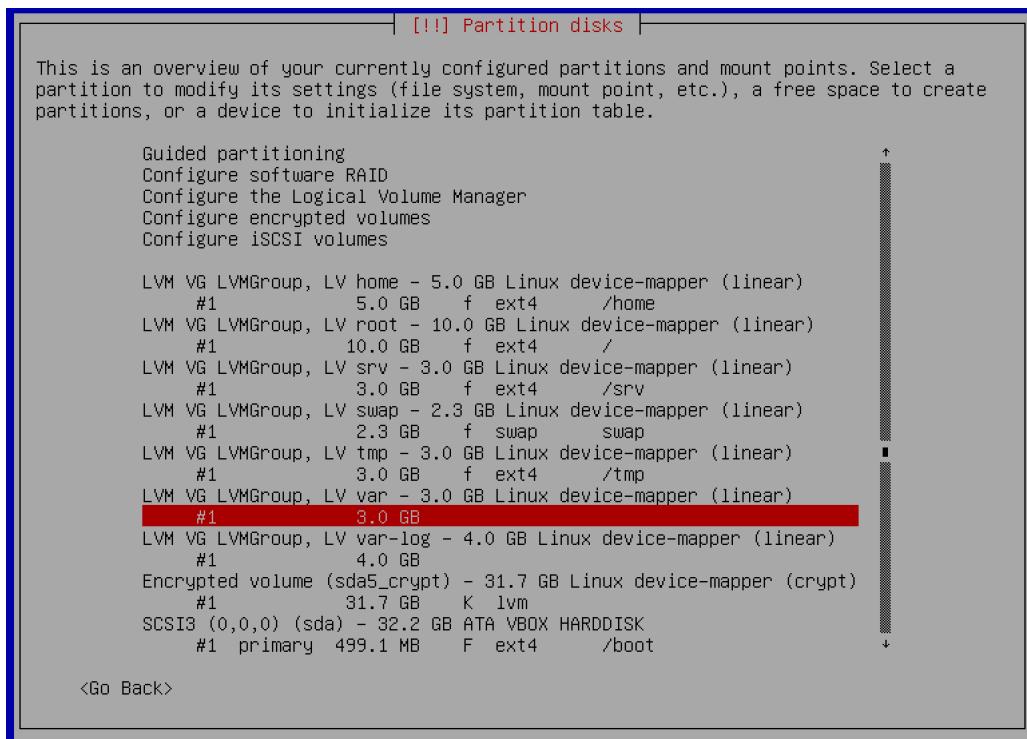
56 ° Ahora si volveremos a hacer lo mismo que antes pero ahora lo haremos con `tmp` y cambiando el punto de montaje.

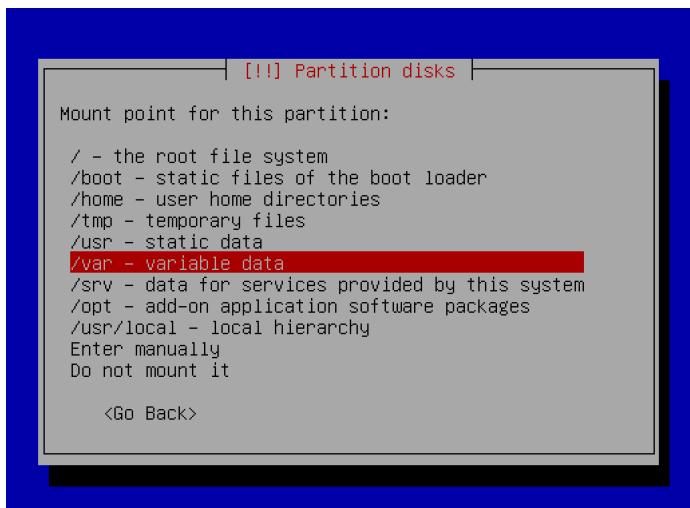
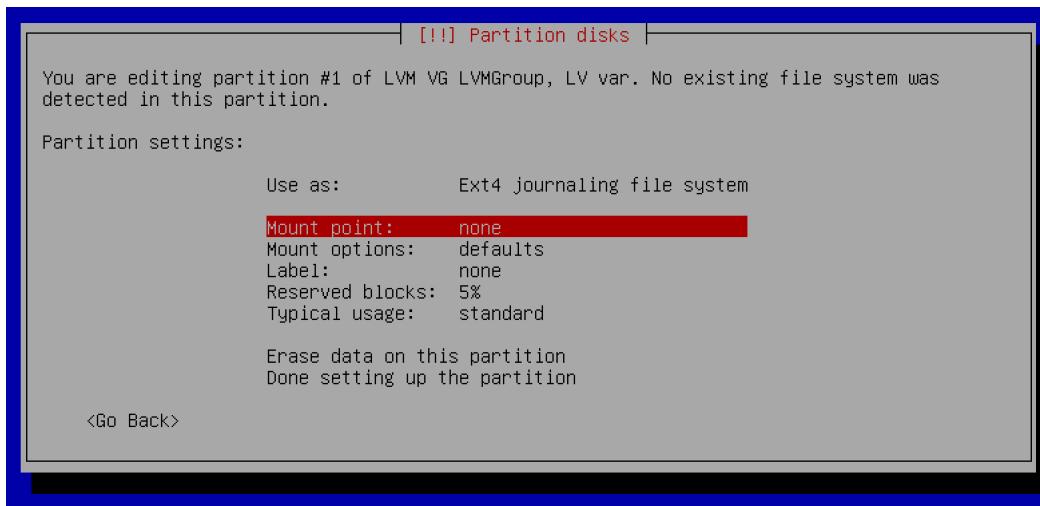
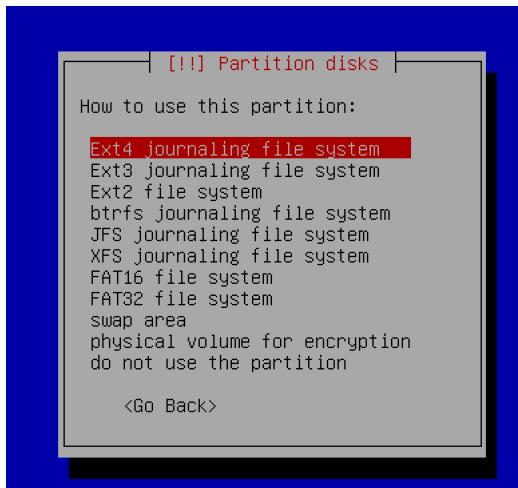


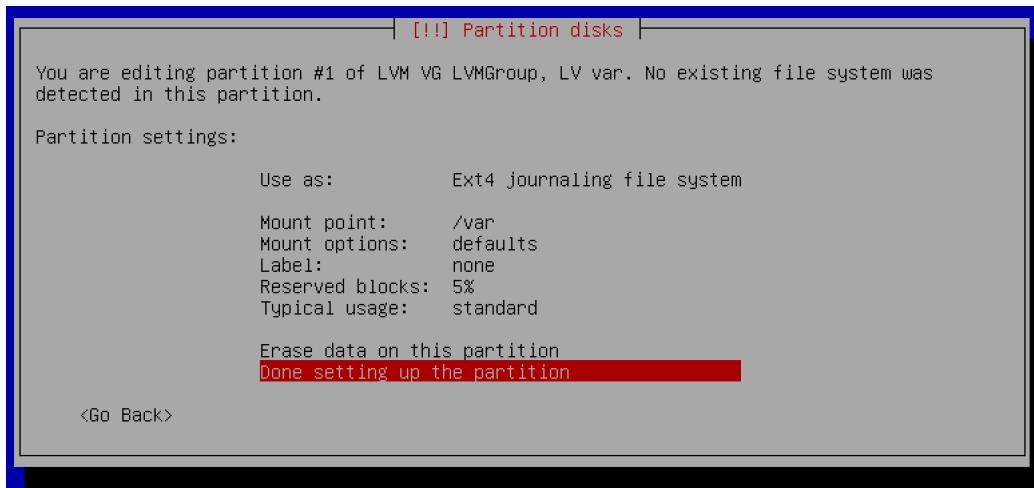




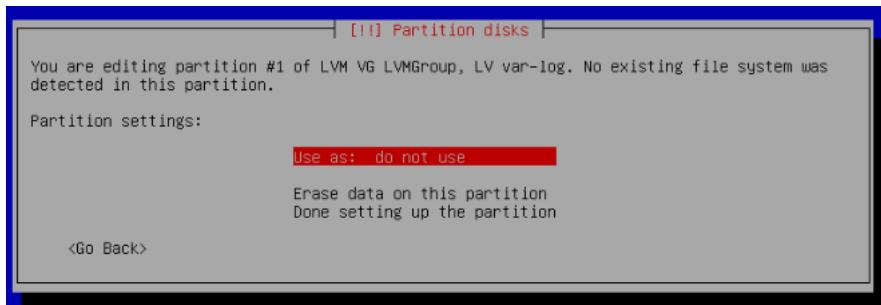
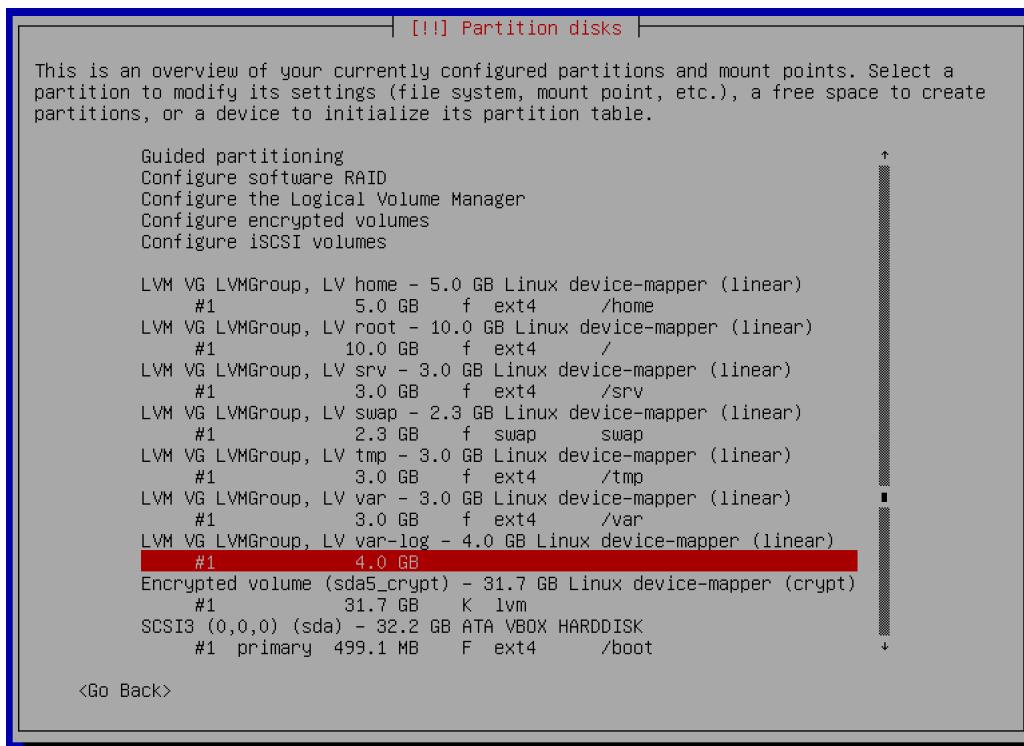
57º Repetimos de nuevo el proceso para var cambiando el punto de montaje.

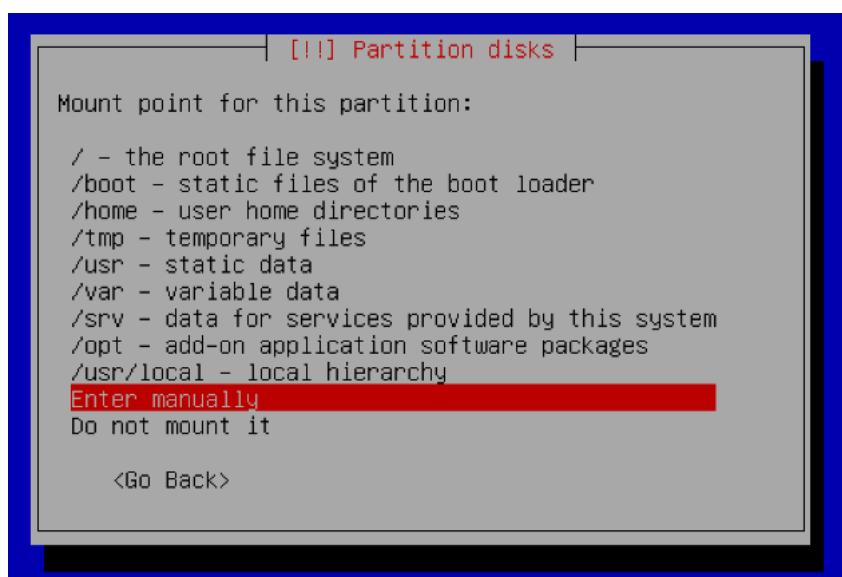
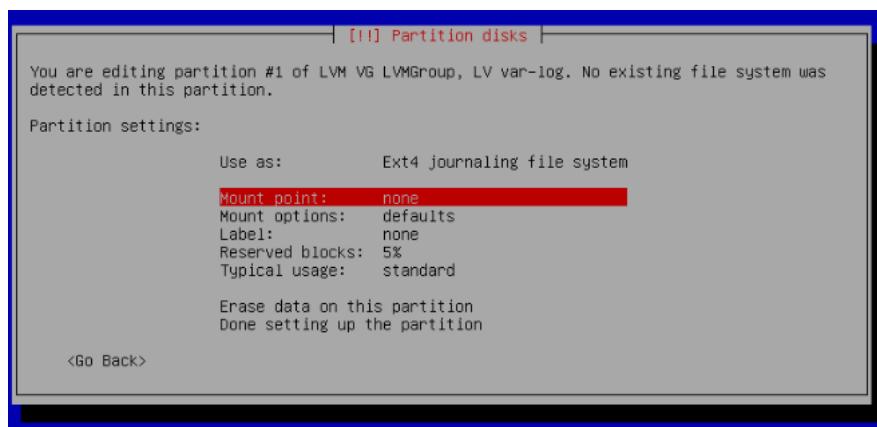
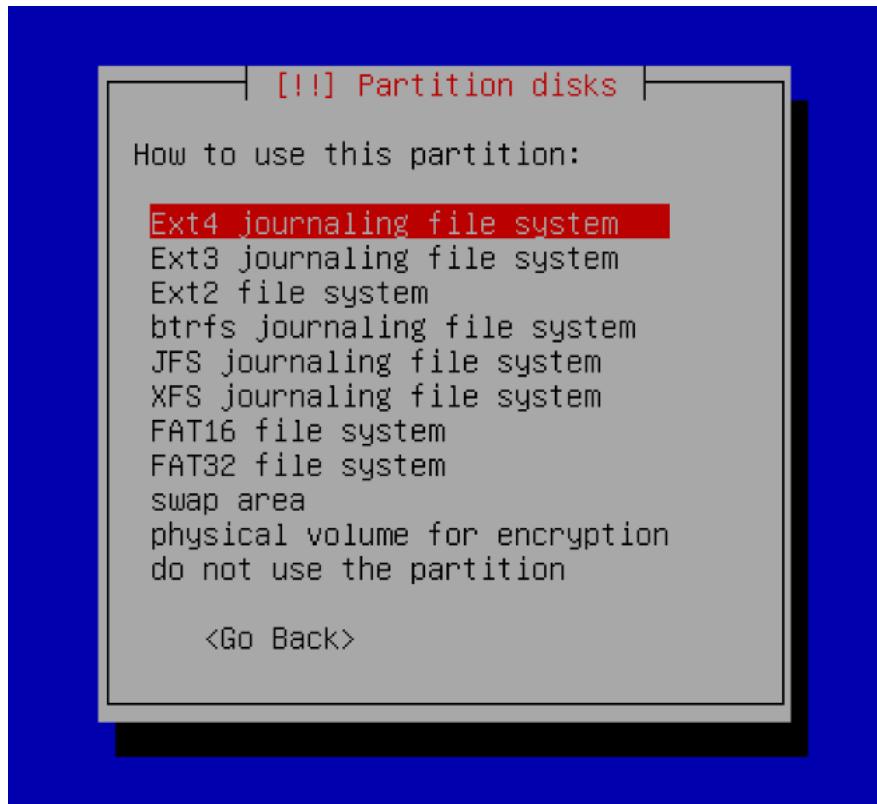


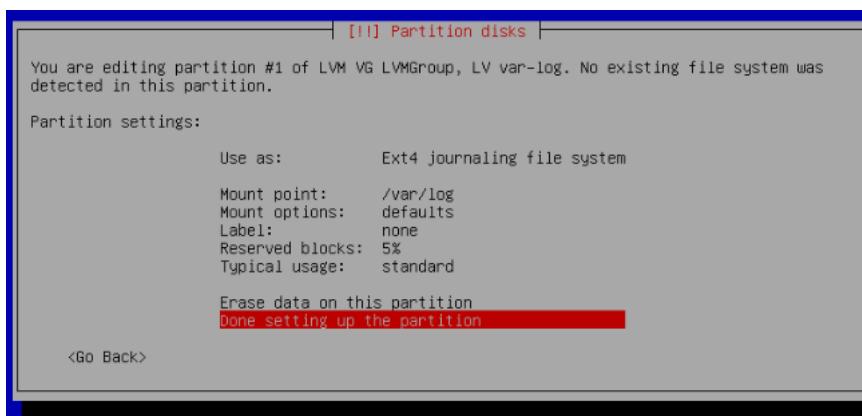
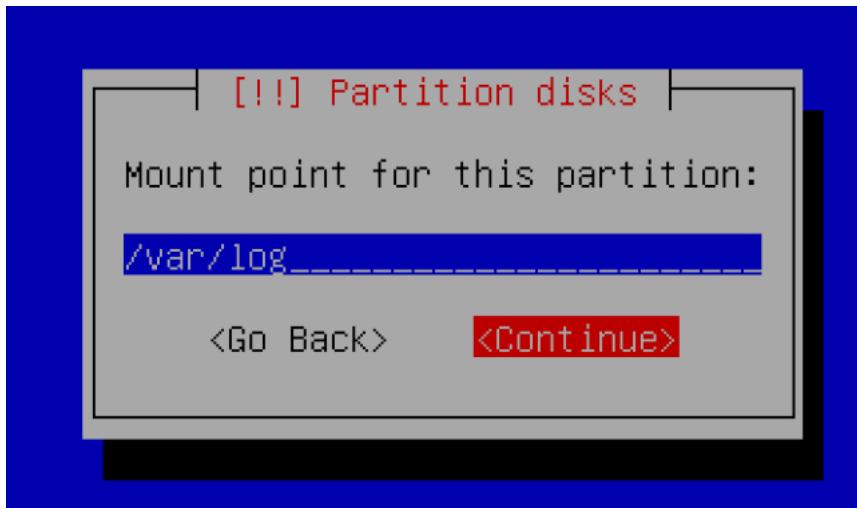




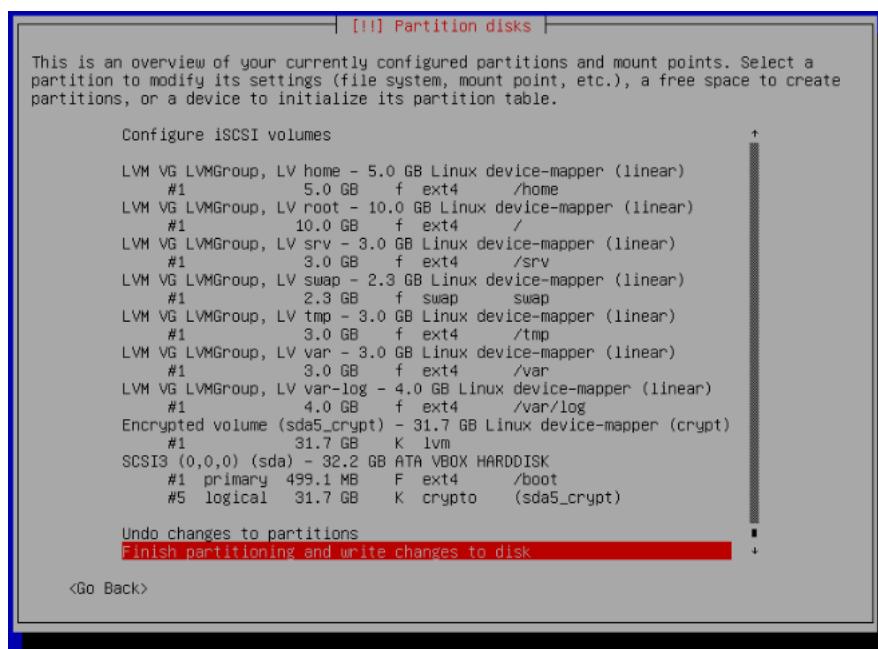
58° Por último repetimos de nuevo el proceso para `var-log` en este deberemos introducir manualmente el punto de montaje.



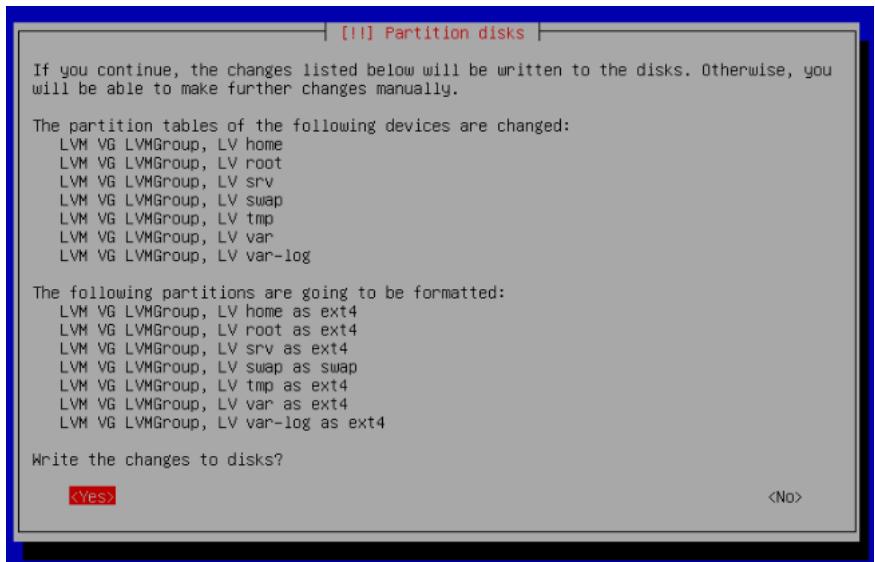




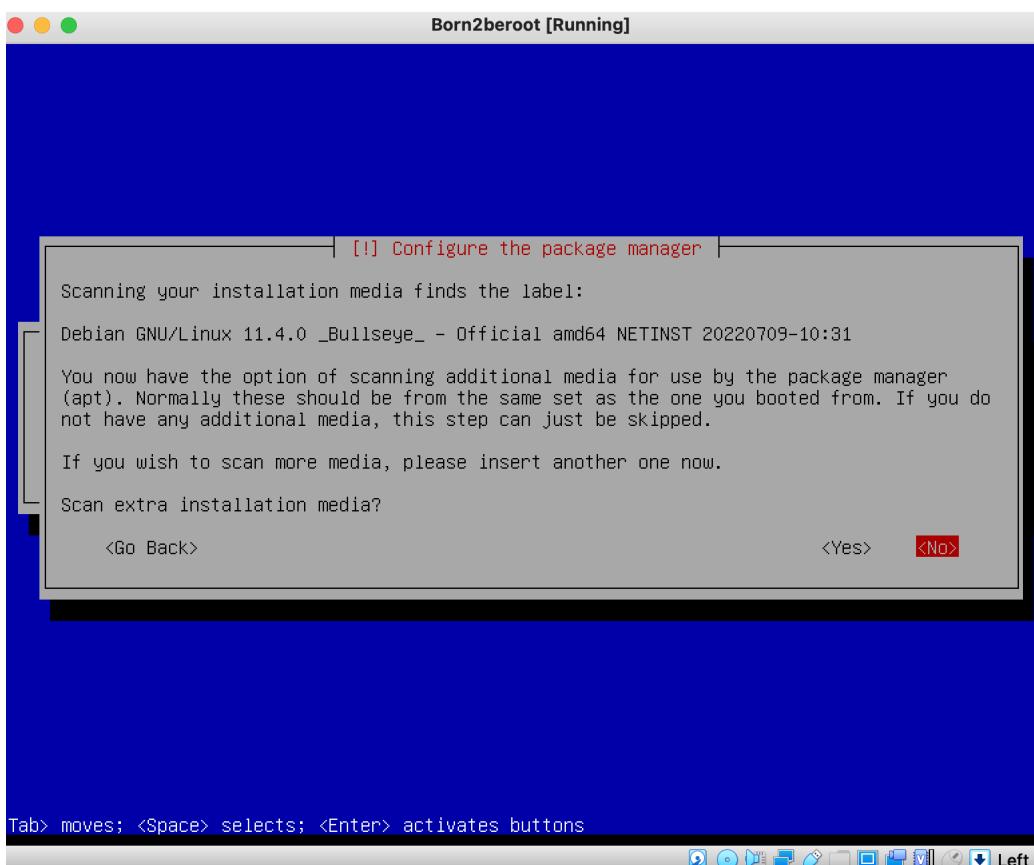
59 ° Una vez hemos completado todos los pasos anteriores ya casi hemos acabado, debemos darle a finalizar el particionado y asi se guarden t los cambios en el disco.



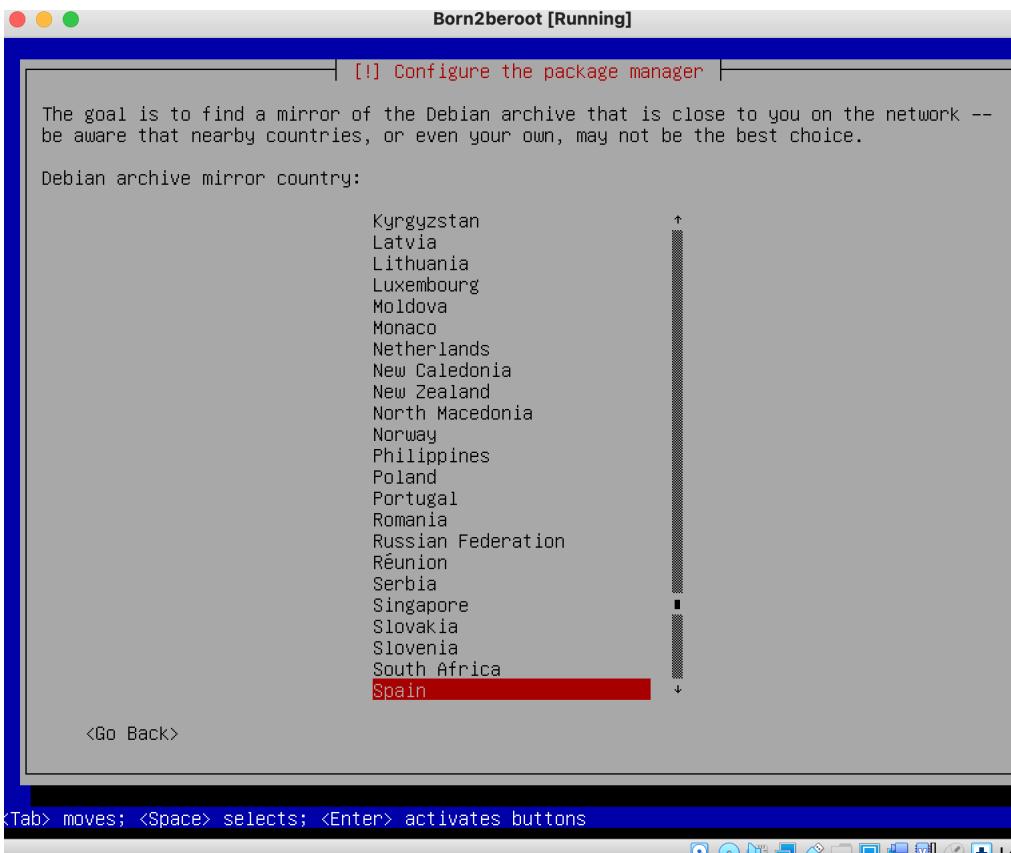
60 ° Aceptamos el mensaje y asi se guardaran los cambios. Asegurate que todas las particiones quedan igual que en la captura.



61 ° Seleccionamos la opción No ya que no necesitamos paquetes adicionales.



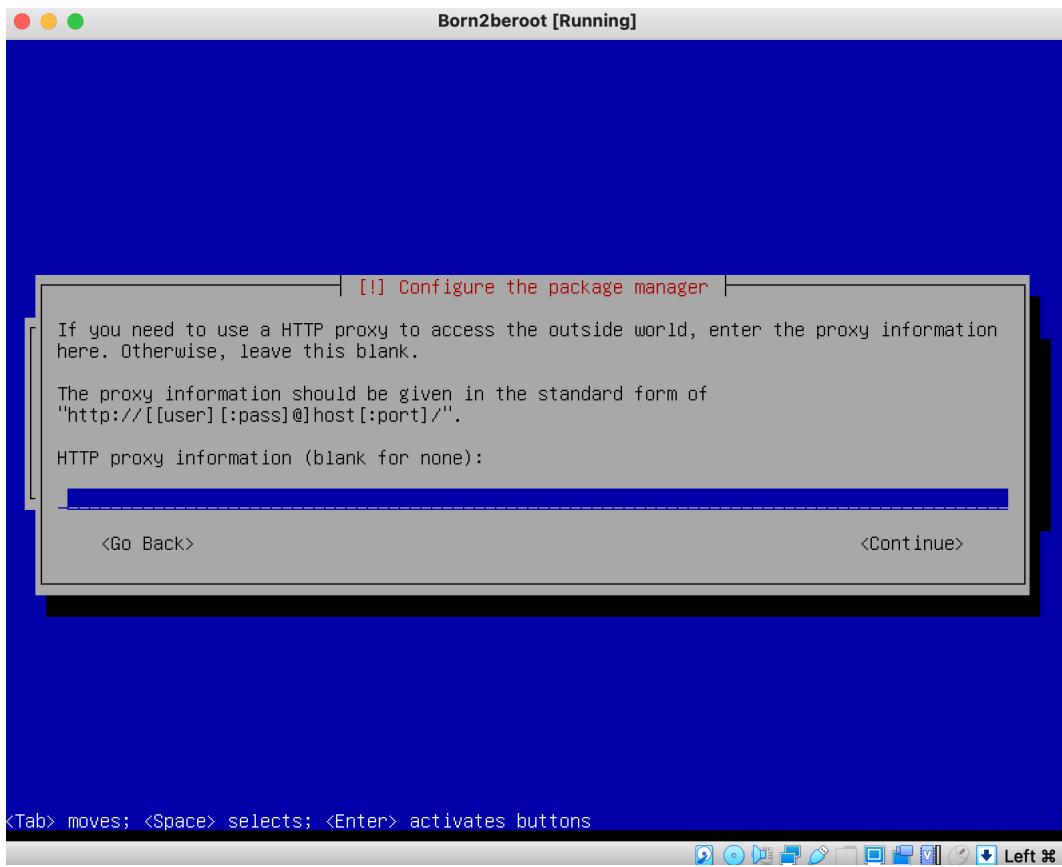
62 ° Escogemos nuestro País.



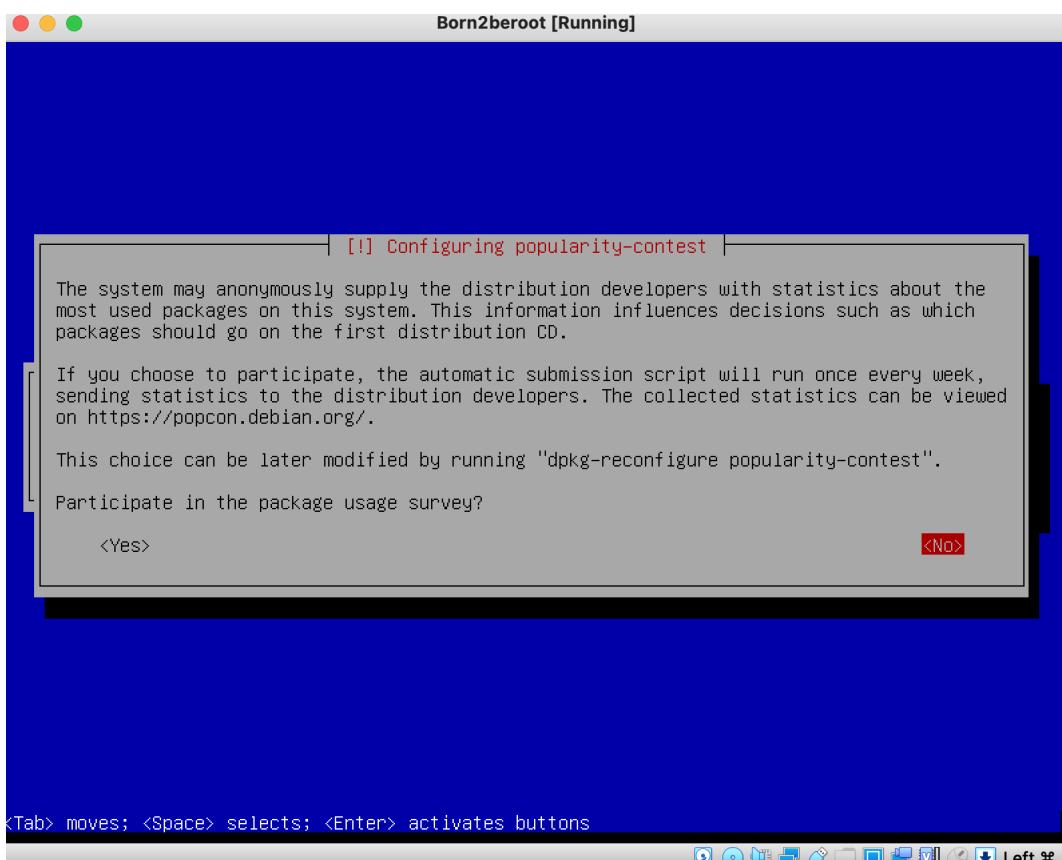
63 ° Escogemos `deb.debian.org` ya que teniendo en cuenta nuestra region es donde tendremos una mejor conexión.



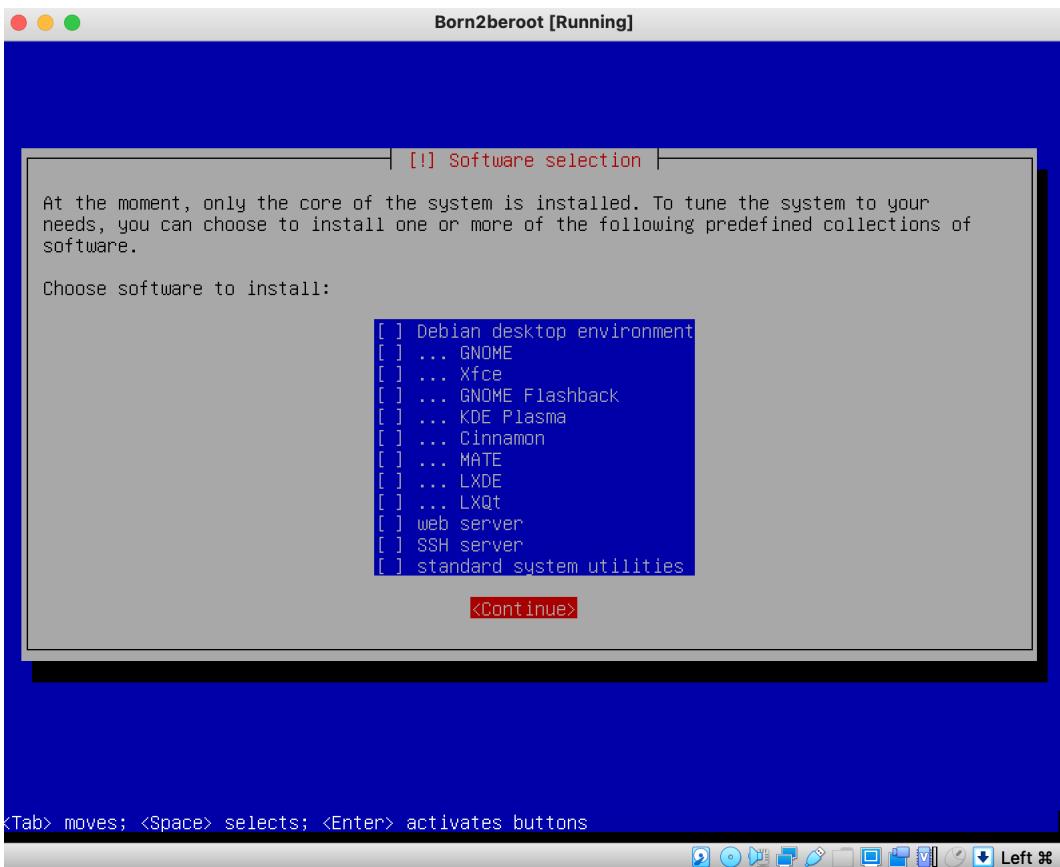
64 ° Esta opción la dejaremos vacía le daremos directamente a `Continue`.



65 ° Seleccionamos la opción `No` ya que no queremos que los developers vean nuestras estadísticas aunque sean anónimas.



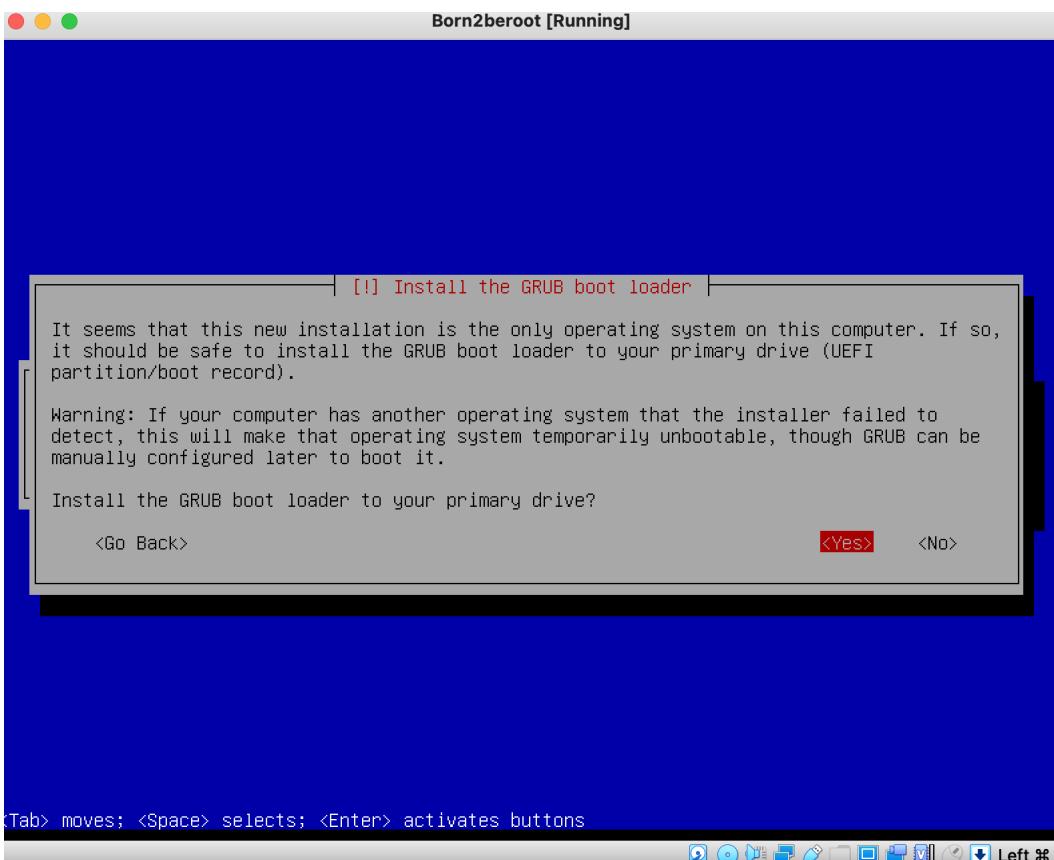
66 ° Quitaremos todas las opciones de software (con la barra espaciadora) y le daremos a `Continue`.



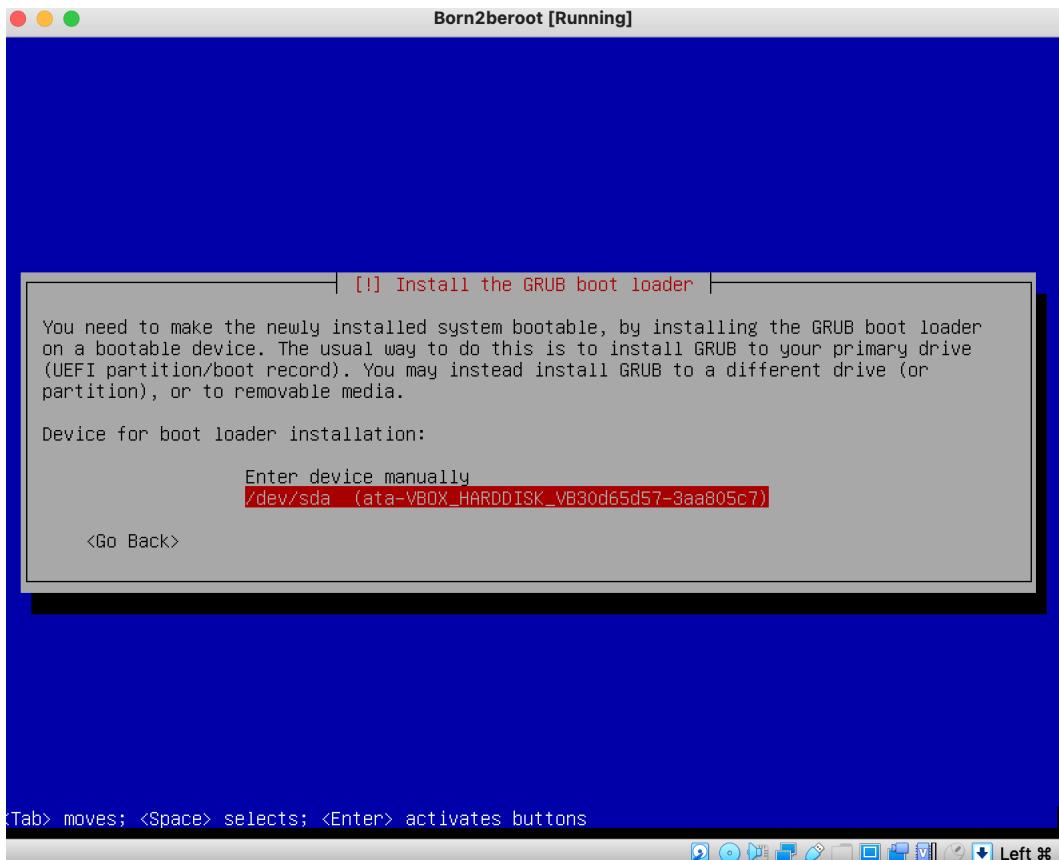
<Tab> moves; <Space> selects; <Enter> activates buttons



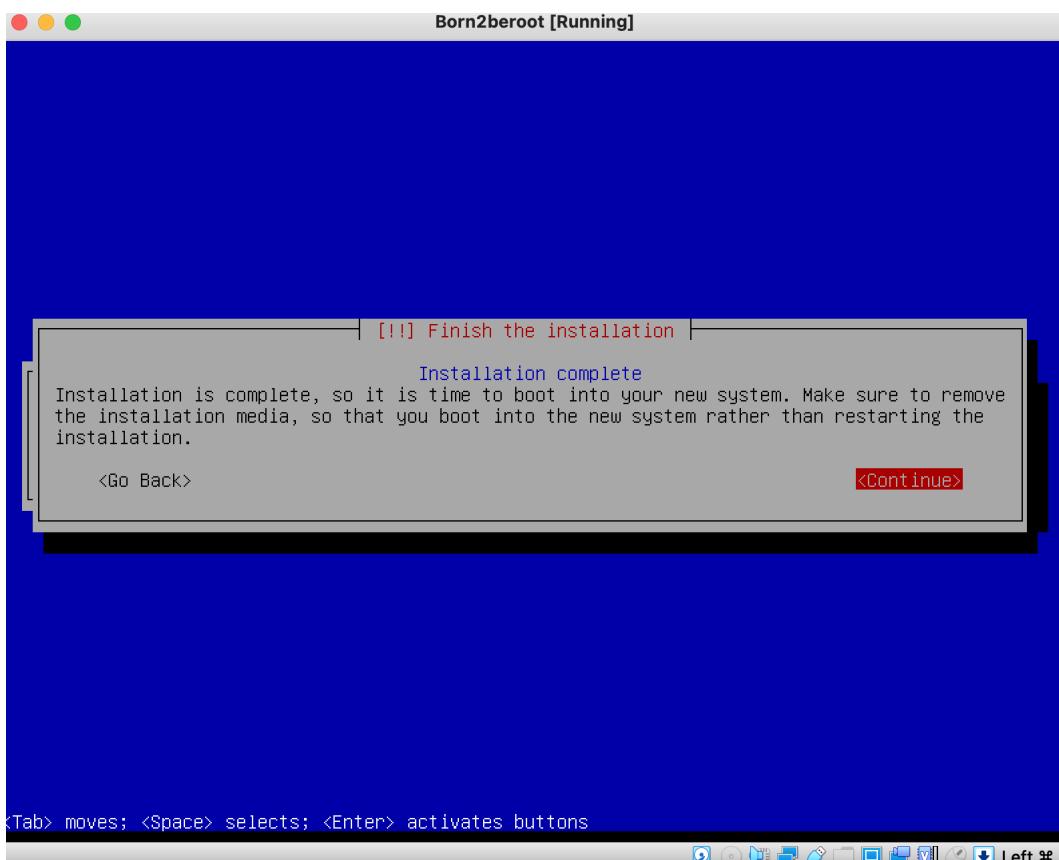
67 ° Seleccionaremos Yes para instalar [GRUB boot](#) en el disco duro.



68 ° Escogeremos el dispositivo para la instalación del cargador de arranque `/dev/sda (ata_VBOX_HARDDISK)`.



69 ° Le daremos a `Continue` para finalizar la instalación.



70 ° Una vez hemos terminado con la instalación de debian debemos configurar nuestra máquina virtual.

[Click aquí para dirigirte a la configuración de la máquina virtual ☺](#)

'8.2 - Wordpress y configuración de servicios ☺

' Lighttpd

💡 **Que es Lighttpd ?** Es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares. Está optimizado para entornos donde la velocidad es muy importante. Esto se debe a que consume menos CPU y memoria RAM que otros servidores.

1 ° Instalación de paquetes de lighttpd.

```
root@gemartin42:~# sudo apt install lighttpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core galera-4 icc-profiles-free
  javascript-common libcgi-fast-perl libcgiconf-perl libclone-perl libconfig-inifiles-perl
  libdbd-mariadb-perl libdbd-perl libdeflate perl libencode-locale-perl libfcgi-bin libfcgi-perl
  libfcgioldperl libfontconfig1 libgd3 libhtml-parser-perl libhttplib-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl libjbig2 libjpeg62-turbo libjs-bootstrap4
  libjs-codemirror libjs-jquery libjs-jquery-mousewheel libjs-jquery-timepicker libjs-jquery-ui
  libjs-openlayers libjs-popper.js libjs-sizzle libjs-sphinxdoc libjs-underscore
  liblwp-mediatypes-perl libmariadb3 libonig5 libsnappy1v5 libterm-readkey-perl libtiff5
  libtimedate-perl liburi-perl libwebp6 libxml2 libxslt1.1 libzip4 lsof mariadb-client-10.5
  mariadb-client-core-10.5 mariadb-common mariadb-server-10.5 mariadb-server-core-10.5
  mysql-common node-jquery php-bz2 php-curl php-gd php-google-recaptcha php-mariadb-mysql-kbs
  php-mbstring php-phpmyadmin-motranslator php-phpmyadmin-shapefile php-phpmyadmin-sql-parser
  php-phplib php-psr-cache php-psr-container php-psr-log php-symfony-cache
  php-symfony-cache-contracts php-symfony-config php-symfony-dependency-injection
  php-symfony-expression-language php-symfony-filesystem php-symfony-service-contracts
  php-symfony-var-exporter php-symfony-yaml php-tcpdf php-twigi18n-extension php-xml
  php-zip php7.4-bz2 php7.4-curl php7.4-gd php7.4-mbstring php7.4-xml php7.4-zip rsync socat
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  lighttpd-mod-deflate lighttpd-mod-openssl
Suggested packages:
  rrdtool php-fpm apache2-utils lighttpd-doc lighttpd-mod-authn-gssapi lighttpd-mod-authn-pam
  lighttpd-mod-authn-sasl lighttpd-mod-geoip lighttpd-mod-maxminddb lighttpd-mod-trigger-b4-d1
  lighttpd-mod-vhostdb-pgsql lighttpd-mod-webdav lighttpd-modules-dbi lighttpd-modules-ldap
  lighttpd-modules-lua lighttpd-modules-mysql
The following NEW packages will be installed:
  lighttpd lighttpd-mod-deflate lighttpd-mod-openssl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 388 kB of archives.
After this operation, 1,373 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

2 ° Permitimos las conexiones mediante el puerto 80 con el comando `sudo ufw allow 80`.

```
root@gemartin42:~# sudo ufw allow 80
```

3 ° Checkeamos que realmente hayamos permitido. Debe aparecer el puerto 80 y allow.

```
root@gemartin42:~# sudo ufw status
Status: active

To                         Action      From
--                         --          --
4242                       ALLOW       Anywhere
80                        ALLOW       Anywhere
4242 (v6)                   ALLOW       Anywhere (v6)
80 (v6)                     ALLOW       Anywhere (v6)

root@gemartin42:~#
```

4 ° Añadimos la regla que incluya el puerto 80. Si no recuerdas como se añadian reglas en el reenvío de puertos. Configuración de la máquina –> Reenvío de puertos –> Replicar la captura.

Nombre	Protocolo	IP anfitrión	Puerto anfitrión	IP invitado	Puerto invitado
Rule 1	TCP		4242		4242
Rule 2	TCP		80		80

' WordPress

💡 **Que es Wordpress ?** Es un sistema de gestión de contenidos enfocado a la creación de cualquier tipo de página web.

1 ° Para instalar la última versión de WordPress primero debemos instalar wget y zip. Para ello haremos uso del siguiente comando `sudo apt install wget zip`.

💡 **Que es wget ?** Es una herramienta de línea de comandos que se utiliza para descargar archivos de la web.

💡 **Que es zip ?** Es una utilidad de línea de comandos para comprimir y descomprimir archivos en formato ZIP.

```
root@gemartin42:/home/gemartin# sudo apt install wget zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
zip is already the newest version (3.0-12).
The following NEW packages will be installed:
  wget
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
Need to get 964 kB of archives.
After this operation, 3,559 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

2º Una vez hayamos instalado los paquetes nos debemos ubicar en la carpeta /var/www/ con el comando cd accederemos a ella `cd /var/www`

```
root@gemartin42:/home/gemartin# cd /var/www/_
```

3º Una vez estemos en la ruta /var/www/ deberemos descargar la última versión de WordPress. Como mi idioma nativo es el español yo selección la última versión en español. Utilizaremos el siguiente comando: `sudo wget https://es.wordpress.org/latest-es_ES.zip`.

```
root@gemartin42:/var/www# sudo wget https://es.wordpress.org/latest-es_ES.zip
--2022-10-27 15:58:02--  https://es.wordpress.org/latest-es_ES.zip
Resolving es.wordpress.org (es.wordpress.org)... 198.143.164.252
Connecting to es.wordpress.org (es.wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25208367 (24M) [application/zip]
Saving to: 'latest-es_ES.zip.1'

latest-es_ES.zip.1      100%[=====]  24.04M  9.82MB/s   in 2.4s

2022-10-27 15:58:04 (9.82 MB/s) - 'latest-es_ES.zip.1' saved [25208367/25208367]
root@gemartin42:/var/www# _
```

4º Descomprimimos el archivo que acabamos de descargar con el comando `sudo unzip latest-es_ES.zip`.

```
root@gemartin42:/var/www# sudo unzip latest-es_ES.zip
```

5º Renombraremos la carpeta html y la llamaremos html_old. `sudo mv html/ html_old/`.

```
root@gemartin42:/var/www# sudo mv html/ html_old/
```

6º Ahora renombraremos la carpeta wordpress y la llamaremos html. `sudo mv wordpress/ html`.

```
root@gemartin42:/var/www# sudo mv wordpress/ html
```

7º Por último estableceremos estos permisos en la carpeta html. Daremos uso del comando `sudo chmod -R 755 html`. El número 7 indica que propietario tiene permisos de lectura, escritura y ejecución. El número 5 indica que el grupo y otros solo tienen permisos de lectura y ejecución

```
root@gemartin42:/var/www# sudo chmod -R 755 html
```

› Mariadb

⌚ **Que es MariaDB ?** Es una base de datos. Se utiliza para diversos fines, como el almacenamiento de datos, el comercio electrónico, funciones de nivel empresarial y las aplicaciones de registro.

1º Instalaremos los paquetes con el comando `sudo apt install mariadb-server`

```
root@gemartin42:~# sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core icc-profiles-free
  Javascript-common libdeflate0 libfontconfig1 libgd3 libjbig0 libjpeg62-turbo libjs-bootstrap4
  libjs-codemirror libjs-jquery libjs-jquery-mousewheel libjs-jquery-timepicker libjs-jquery-ui
  libjs-openlayers libjs-popper.js libjs-sizzle libjs-sphinxdoc libjsunderscore libonig5 libtiff5
  libwebp6 libxpm4 libxml1.1 libzip4 node-jquery php-bz2 php-curl php-gd php-google-recaptcha
  php-mariadb-mysql-kbs php-mbstring php-pharadmin-motranslator php-pharadmin-shapefile
  php-pharadmin-sql-parser php-phpseclib php-psr-cache php-psr-container php-psr-log
  php-symfony-cache php-symfony-cache-contracts php-symfony-config
  php-symfony-dependency-injection php-symfony-expression-language php-symfony-filesystem
  php-symfony-service-contracts php-symfony-var-exporter php-symfony-yaml php-tcpdf php-twig
  php-twig-i18n-extension php-xml php-zip php7.4-bz2 php7.4-curl php7.4-gd php7.4-mbstring
  php7.4-xml php7.4-zip
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  mariadb-server
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 35.3 kB of archives.
After this operation, 72.7 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 mariadb-server all 1:10.5.15-0+deb11u1 [35.3 kB]
Fetched 35.3 kB in 0s (871 kB/s)
Selecting previously unselected package mariadb-server.
(Reading database ... 34905 files and directories currently installed.)
Preparing to unpack ....mariadb-server_1%3a10.5.15-0+deb11u1_all.deb ...
Unpacking mariadb-server (1:10.5.15-0+deb11u1) ...
Setting up mariadb-server (1:10.5.15-0+deb11u1) ...
root@gemartin42:~# _
```

2º Debido a que la configuración predeterminada deja su instalación de MariaDB poco segura, utilizaremos un script que proporciona el paquete `mariadb-server` para restringir el acceso al servidor y eliminar las cuentas no utilizadas. Ejecutaremos el script con el siguiente comando `sudo mysql_secure_installation`. Una vez ejecutemos el script nos hará una serie de preguntas. Preguntará si deseamos cambiar a la autenticación de socket de Unix. Como ya tenemos una cuenta root protegida escribiremos `N`.

```
Switch to unix_socket authentication? → N
Change the root password? → N
Remove anonymous users? → Y
Disallow root login remotely? → Y
Remove test database and access to it? → Y
Reload privilege tables now? → Y
```

```
root@gemartin42:~# sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] N_
```

```
Switch to unix_socket authentication [Y/n] N
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] N
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
```

```
Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your Mariadb
installation should now be secure.

Thanks for using MariaDB!
root@gemartin42:/home/gemartin# _
```

Switch to unix_socket autentication? Escogemos `N` porque no deseamos que cambie a la autenticación de socket de Unix ya tenemos una cuenta root protegida.

Change the root password? Escogemos `N`. No deseamos cambiar la contraseña del usuario root. Por defecto no tenemos contraseña pero en mariadb realmente no es root ya que debemos darle permisos de administrador.

Remove anonymous users? Escogemos `Y`. Por defecto cuando instalas mariadb tiene un usuario anonimo, lo que permite que cualquier persona inicie sesión en mariadb sin tener que crear una cuenta de usuario propia. Esto esta diseñado para realizar pruebas y que la instalación sea más fluida. Cuando dejemos el entorno de desarrollo y queramos pasar a un entorno de producción debemos eliminar los usuarios anonimos.

Disallow root login remotely? Escogemos `Y`. Al deshabilitar el inicio de sesión en root de forma remota evitaremos que alguien pueda adivinar contraseña root. Solo podremos conectarnos al root desde localhost.

Remove test database and acces to it? Escogemos `Y`. De esta manera se eliminará la base de datos de prueba y cualquier usuario que tenga acceso a ella.

Reload privilege tables now? Escogemos `Y`. Así se recargarán las tablas de permisos de MySQL para que los cambios en la configuración de seguridad entren en vigor de inmediato.

1º Una vez hayamos terminado con la instalación de mariadb debemos crear la base de datos y el usuario para el WordPress. Lo primero debe ser acceder a mariadb.

```
root@gemartin42:/home/gemartin# mariadb
```

2º Creamos una base de datos para el WordPress. En mi caso le voy a llamar `wp_database`. Todo esto lo haremos con el comando `CREATE DATABASE wp_database;`

```
MariaDB [(none)]> CREATE DATABASE wp_database;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]>
```

3 ° Para asegurarnos que se ha creado la base de datos para el WordPress podemos ver todas las bases existentes con el comando `SHOW DATABASES;`

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| wp_database    |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> _
```

4 ° Acto seguido debemos crearemos un usuario dentro de la base de datos. Utilizaremos el comando `CREATE USER 'gemartin'@'localhost' IDENTIFIED BY '12345';`

```
MariaDB [(none)]> CREATE USER 'gemartin'@'localhost' IDENTIFIED BY '12345';
Query OK, 0 rows affected (0.011 sec)

MariaDB [(none)]> _
```

5 ° Vinculamos el nuevo usuario a nuestra base de datos de manera que le otorguemos los permisos necesario para poder trabajar. Daremos el comando `GRANT ALL PRIVILEGES ON wp_database.* TO 'gemartin'@'localhost';`

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wp_database.* TO 'gemartin'@'localhost';
Query OK, 0 rows affected (0.009 sec)

MariaDB [(none)]> _
```

6 ° Actualizamos los permisos para que los cambios tengan efecto con el comando `FLUSH PRIVILEGES;`

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> _
```

7 ° Una vez hemos completado el paso anterior ya podemos salir de mariadb.

```
MariaDB [(none)]> exit
Bye
root@gemartin42:/var/www/html#
```

› PHP

• **Que es PHP ?** Es un lenguaje de programación. Se utiliza principalmente para desarrollar aplicaciones web dinámicas y sitios web interactivos. Se ejecuta en el lado del servidor.

1 ° Instalamos los paquetes necesarios para poder ejecutar aplicaciones web escritas en lenguaje PHP y que necesiten conectarse a una base de MySQL. Ejecutaremos el siguiente comando `sudo apt install php-cgi php-mysql`.

```
root@gemartin42:/home/gemartin# sudo apt install php-cgi php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php-cgi is already the newest version (2:7.4+76).
php-mysql is already the newest version (2:7.4+76).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@gemartin42:/home/gemartin#
```

› Configuración WordPress

1 ° Accedemos al directorio /var/www/html con el comando: `cd /var/www/html`

```
root@gemartin42:~# cd /var/www/html
root@gemartin42:/var/www/html# _
```

2 ° Copiamos el fichero wp-config-sample.php y lo renombraremos wp-config.php

```
root@gemartin42:/var/www/html# cp wp-config-sample.php wp-config.php
```

3 ° Una vez lo hayamos renombrado editaremos el fichero wp-config.php `nano wp-config.php` y modificaremos los siguientes valores.

```
GNU nano 5.4                               wp-config.php
* This file contains the following configurations:
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );
/** Database username */
define( 'DB_USER', 'username_here' );
/** Database password */
define( 'DB_PASSWORD', 'password_here' );
/** Database hostname */
define( 'DB_HOST', 'localhost' );
/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
/**#@+
 * Authentication unique keys and salts.
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  M-U Undo
^X Exit      ^R Read File   ^Y Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

Hay que remplazarlos por los valores que hemos puesto anteriormente cuando creábamos la base de datos y el usuario para que WordPress pi conector y hacer uso de ella.

```
GNU nano 5.4                               wp-config.php *
* The wp-config.php creation script uses this file during the installation.
* You don't have to use the web site, you can copy this file to "wp-config.php"
* and fill in the values.
*
* This file contains the following configurations:
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://wordpress.org/support/article/editing-wp-config-php/
*
* @package WordPress
*/
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wp_database' );
/** Database username */
define( 'DB_USER', 'gemartin' );
/** Database password */
define( 'DB_PASSWORD', '12345' );
/** Database hostname */
define( 'DB_HOST', 'localhost' );
/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The database collate type. Don't change this if in doubt. */
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  M-U Undo
^X Exit      ^R Read File   ^Y Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

4º Habilitamos el módulo fastcgi-php en Lighttpd para mejorar el rendimiento y la velocidad de las aplicaciones web en el servidor. `sudo light enable-mod fastcgi`

```
gemartin@gemartin42:~$ sudo lighty-enable-mod fastcgi  
already enabled  
Run "service lighttpd force-reload" to enable changes
```

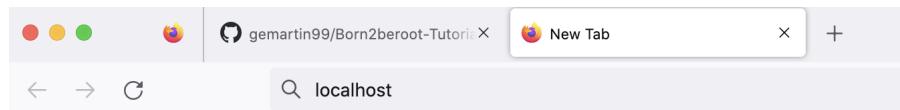
5º Habilitamos el módulo fastcgi-php en Lighttpd para mejorar el rendimiento y la velocidad de las aplicaciones web basadas en PHP en el servidor. Podemos hacerlo con el comando `sudo lighty-enable-mod fastcgi-php`.

```
gemartin@gemartin42:~$ sudo lighty-enable-mod fastcgi-php  
already enabled  
Run "service lighttpd force-reload" to enable changes  
gemartin@gemartin42:~$
```

6º Actualizamos y aplicamos los cambios en la configuración con el comando `sudo service lighttpd force-reload`.

```
gemartin@gemartin42:~$ sudo service lighttpd force-reload  
gemartin@gemartin42:~$ _
```

7º Una vez ya hemos completado los pasos anteriores podemos volver a dirigirnos a nuestro navegador y escribiremos `localhost`. Nos deberíamos dirigir a la siguiente:





Hola

¡Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña

*@NJIrg1Rn%%DjG%60

Ocultar

Fuerte

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Tu correo electrónico

Comprueba bien tu dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda

Pedir a los motores de búsqueda que no indexen este sitio

Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

8º Debemos llenar todos los campos. En mi caso he puesto lo siguiente:



Hola

Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	Gemartin WP
Nombre de usuario	wp-gemartin
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.	
Contraseña	<input type="password"/> 12345 Ocultar Muy débil
Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.	
Confirma la contraseña	<input checked="" type="checkbox"/> Confirma el uso de una contraseña débil.
Tu correo electrónico	gemartin@student.42barcelor
Comprueba bien tu dirección de correo electrónico antes de continuar.	
Visibilidad en los motores de búsqueda	<input type="checkbox"/> Pedir a los motores de búsqueda que no indexen este sitio Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

9º Una vez hayamos llenado todos los campos debemos darle a [Instalar WordPress](#) y ya habremos terminado la instalación. Nos saldrá la siguiente pestaña. Ahora WordPress puede crear las tablas y volcar todos los datos que necesita para funcionar en la base de datos que le hemos asignado.



¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

Nombre de usuario wp-gemartin

Contraseña La contraseña que has elegido.

[Acceder](#)

10 ° Si accedemos de nuevo a nuestro localhost desde el navegador ya podemos ver nuestra página funcional.



¡Hola, mundo!

Bienvenido a WordPress. Esta es tu primera entrada. Editala o bórrala, ¡luego empieza a escribir!

abril 1, 2023

Funciona gracias a [WordPress](#)

11 ° Si queremos acceder al panel de administrador para hacer cambios en nuestra página deberemos poner en el navegador `localhost/wp-admin` iniciaremos sesión con nuestra cuenta.



Nombre de usuario o correo electrónico

Contraseña
 

Recuérdame Acceder

[¿Has olvidado tu contraseña?](#)
[← Ir a Gemartin WP](#)

 Español  Cambiar



Ahora estás desconectado.

Nombre de usuario o correo electrónico

Contraseña
 

Recuérdame Acceder

[¿Has olvidado tu contraseña?](#)
[← Ir a Gemartin WP](#)

 Español  Cambiar

12 ° Una vez accedamos ya podemos modificar lo que queramos a gusto propio. Personalizar la página es algo opcional, como no está especificado el subject en esta guía no se tratará nada al respecto.

The screenshot shows the WordPress dashboard at localhost/wp-admin/. The top navigation bar includes links for 'localhost' and 'Gemartin WP'. The sidebar on the left is titled 'Escritorio' and lists various menu items: Inicio (Actualizaciones 6), Entradas, Medios, Páginas, Comentarios, Apariencia, Plugins (1), Usuarios, Herramientas, Ajustes, and Cerrar menú. A prominent yellow banner at the top right says '¡Ya está disponible WordPress 6.2! Por favor, actualiza ahora.' Below the banner, the main content area features a large '6.0' graphic and the text '¡Bienvenido a WordPress! Aprende más sobre la versión 6.0.3.' There are three call-to-action boxes: 'Crea contenido rico con bloques y patrones', 'Personaliza todo tu sitio con temas de bloques', and 'Cambia la apariencia de tu sitio con los estilos'.

8.3 - Servicio adicional +

LiteSpeed ↗

Que es LiteSpeed ? Es un software de servidor web patentado. Es el cuarto servidor web más popular, y se estima que lo utiliza el 10% de los sitios web.

1 ° Antes de instalar cualquier software, es importante asegurarse de que el sistema esté actualizado.

```
sudo apt update
```

```
root@gemartin42:/home/gemartin# sudo apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Get:2 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:3 http://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Ign:4 http://rpms.litespeedtech.com/debian bullseye InRelease
Hit:5 http://rpms.litespeedtech.com/debian bullseye Release
Fetched 92.4 kB in 0s (203 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@gemartin42:/home/gemartin# _
```

```
sudo apt upgrade
```

```
root@gemartin42:/home/gemartin# sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@gemartin42:/home/gemartin# _
```

2 ° De forma predeterminada, OpenLiteSpeed está disponible en el repositorio base de Debian 11. Entonces, debes ejecutar el siguiente comando para agregar el repositorio OpenLiteSpeed a su sistema Debian:

```
 wget -O - https://repo.litespeed.sh | sudo bash
```

Como el comando es largo me he conectado via ssh.

```
gemartin@car12s2 ~ % ssh gemartin@localhost -p 4242
gemartin@localhost's password:
Linux gemartin42 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 25 03:03:33 2022 from 10.0.2.2
gemartin@gemartin42:~$ su
Password:
root@gemartin42:/home/gemartin# wget -O - http://rpms.litespeedtech.com/debian/enable_lst_debian_repo.sh | sudo bash
--2022-11-25 03:05:36-- http://rpms.litespeedtech.com/debian/enable_lst_debian_repo.sh
Resolving rpms.litespeedtech.com (rpms.litespeedtech.com)... 52.55.120.73
Connecting to rpms.litespeedtech.com (rpms.litespeedtech.com)|52.55.120.73|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3457 (3.4K) [application/x-sh]
Saving to: 'STDOUT'

[  0%] 100%[=====] 3.38K --.-KB/s   in 0s

2022-11-25 03:05:36 (617 MB/s) - written to stdout [3457/3457]

detecting OS type :
detected OS: debian - 11
now enable the LiteSpeed Debian Repo
register LiteSpeed GPG key
--2022-11-25 03:05:36-- http://rpms.litespeedtech.com/debian/lst_debian_repo.gpg
Resolving rpms.litespeedtech.com (rpms.litespeedtech.com)... 52.55.120.73
Connecting to rpms.litespeedtech.com (rpms.litespeedtech.com)|52.55.120.73|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1198 (1.2K) [application/octet-stream]
Saving to: '/etc/apt/trusted.gpg.d/lst_debian_repo.gpg'

/etc/apt/trusted.gpg.d/lst_debian_repo.g 100%[=====] 1.17K --.-KB/s   in 0s

2022-11-25 03:05:36 (262 MB/s) - '/etc/apt/trusted.gpg.d/lst_debian_repo.gpg' saved [1198/1198]

--2022-11-25 03:05:36-- http://rpms.litespeedtech.com/debian/lst_repo.gpg
Resolving rpms.litespeedtech.com (rpms.litespeedtech.com)... 52.55.120.73
Connecting to rpms.litespeedtech.com (rpms.litespeedtech.com)|52.55.120.73|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2336 (2.3K) [application/octet-stream]
Saving to: '/etc/apt/trusted.gpg.d/lst_repo.gpg'

/etc/apt/trusted.gpg.d/lst_repo.gpg 100%[=====] 2.28K --.-KB/s   in 0s

2022-11-25 03:05:37 (194 MB/s) - '/etc/apt/trusted.gpg.d/lst_repo.gpg' saved [2336/2336]

update the repo
Hit:1 http://deb.debian.org/debian bullseye InRelease
Get:2 http://security.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Ign:4 http://rpms.litespeedtech.com/debian bullseye InRelease
Hit:5 http://rpms.litespeedtech.com/debian bullseye Release
Fetched 48.4 kB in 0s (126 kB/s)
Reading package lists... Done
All done, congratulations and enjoy !
root@gemartin42:/home/gemartin#
```

3º De nuevo, actualizamos los paquetes y instalaremos OpenLiteSpeed.

```
sudo apt update
```

```
root@gemartin42:/home/gemartin# sudo apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://deb.debian.org/debian bullseye-updates InRelease
Hit:3 http://security.debian.org/debian-security bullseye-security InRelease
Ign:4 http://rpms.litespeedtech.com/debian bullseye InRelease
Hit:5 http://rpms.litespeedtech.com/debian bullseye Release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@gemartin42:/home/gemartin#
```

```
sudo apt install openlitespeed
```

```

root@gemartin42:/home/gemartin# sudo apt install openlitespeed
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  openlitespeed
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,171 kB of archives.
After this operation, 31.2 MB of additional disk space will be used.
Get:1 http://rpms.litespeedtech.com/debian bullseye/main amd64 openlitespeed amd64 1.7.16-3+bullseye [8,171 kB]
Fetched 8,171 kB in 2s (4,637 kB/s)
Selecting previously unselected package openlitespeed.
(Reading database ... 44405 files and directories currently installed.)
Preparing to unpack .../openlitespeed_1.7.16-3+bullseye_amd64.deb ...
Unpacking openlitespeed (1.7.16-3+bullseye) ...
Setting up openlitespeed (1.7.16-3+bullseye) ...
root@gemartin42:/home/gemartin# 

```

4 ° La contraseña predeterminada para OpenLiteSpeed es 123456. Cambiaremos la contraseña a algo más seguro con el siguiente comando.

```
sudo /usr/local/lsws/admin/misc/admpass.sh
```

```

root@gemartin42:/home/gemartin# sudo /usr/local/lsws/admin/misc/admpass.sh
Please specify the user name of administrator.
This is the user name required to login the administration Web interface.

User name [admin]: idroot

Please specify the administrator's password.
This is the password required to login the administration Web interface.

Password:
Retype password:
Administrator's username/password is updated successfully!
root@gemartin42:/home/gemartin# 

```

5 ° Configuramos el firewall para permitir las conexiones mediante los puertos 8088 y 7080. Acto seguido agregaremos las reglas en el reenvío de puertos.

```
sudo ufw allow 8088/tcp
```

```

root@gemartin42:/home/gemartin# sudo ufw allow 8088/tcp
Rule added
Rule added (v6)
root@gemartin42:/home/gemartin# 

```

```
sudo ufw allow 7080/tcp
```

```

root@gemartin42:/home/gemartin# sudo ufw allow 7080/tcp
Rule added
Rule added (v6)
root@gemartin42:/home/gemartin# 

```

```
sudo ufw reload
```

```

root@gemartin42:/home/gemartin# sudo ufw reload
Firewall reloaded
root@gemartin42:/home/gemartin# 

```

Reglas en el reenvío de puertos.

Nombre	Protocolo	IP anfitrión	Puerto anfitrión	IP invitado	Puerto invitado
Rule 1	TCP		4242		4242
Rule 2	TCP		80		80
Rule 3	TCP		7080		7080
Rule 4	TCP		8088		8088

6 ° Una vez completado el paso anterior ya podemos conectarnos. Pondremos en el buscador de nuestro navegador `localhost:7080` y proporcionaremos nuestras credenciales de inicio de sesión y ya tendremos acceso a todo.



OpenLiteSpeed

Invalid credentials.

User Name

Password

Copyright © 2014-2022 LiteSpeed Technologies, Inc.

OpenLiteSpeed CURRENT VERSION: OpenLiteSpeed 1.7.16 English

Gemartin42

Dashboard Server Configuration Listeners Virtual Hosts VHost Templates Tools WebAdmin Settings Help

LSWS PID 654 SYSTEM LOAD A 0, 0, 0

Live Feeds Realtime

Http In (KB) Http Out (KB) Https In (KB) Https Out (KB) Http Used Http Idle Https Used Requests in Processing Requests/Sec

HTTP IN (KB) HTTP OUT (KB) HTTPS IN (KB) HTTPS OUT (KB) HTTP USED HTTP IDLE HTTPS USED REQUESTS IN PROCESSING REQUESTS/SEC

LSWS Uptime: 01:09:32 Total Requests: 140 Anti-DDoS Blocked IP Count: 0 Free Conn: 9999 Http Used: 0 Max Conn: 10000 Free SSL Conn: 9999 Https Used: 1 Max SSL Conn: 10000 Requests in Processing: 0 Requests/Sec: 8

Listeners 1 Virtual Hosts 1

Name Address

Default *:8088

Showing 1 to 1 of 1 entries Previous 1

Server Error Log (Last 20KB) 122 Debug Log Go to Log Viewer

Time	Level	Message
2022-11-24 20:49:13.413571	NOTICE	[663] [AdminPHP] add child process pid: 1982
2022-11-24 20:49:13.410590	NOTICE	[663] [LocalWorker::workerExec] VHost:_AdminVHost suExec check uid 997 gid 65534 setuidmode 2.
2022-11-24 20:49:13.410626	NOTICE	[663] [LocalWorker::workerExec] Config[AdminPHP]: suExec uid -1 gid -1 cmd /usr/local/lsws/admin/fcgi-bin/admin ..conf/php.ini, final uid 997 gid 65534, flags: 0.
2022-11-24 20:49:11.536065	NOTICE	[663] [AdminPHP] add child process pid: 1981

Este tutorial ha llevado mucho trabajo, si crees que te ha sido útil agradecería mucho starred ⭐ para que así se comparta y pueda ayudar a más estudiantes 🎓🎓🎓

› 9- Hoja de corrección ✓

Preliminaries

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report it. Take this decision calmly, wisely, and please, use this button with caution.

Preliminary tests

- Defense can only happen if the student being evaluated or group is present. This way everybody learns by sharing knowledge with each other.
- If no work has been submitted (or wrong files, wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.
- For this project, you have to clone their Git repository on their station.

Yes

No

General instructions

General instructions

- During the defense, as soon as you need help to verify a point, the student evaluated must help you.
- Ensure that the "signature.txt" file is present at the root of the cloned repository.
- Check that the signature contained in "signature.txt" is identical to that of the ".vdi" file of the virtual machine to be evaluated. A simple "diff" should allow you to compare the two signatures. If necessary, ask the student being evaluated where their ".vdi" file is located.
- As a precaution, you can duplicate the initial virtual machine in order to keep a copy.
- Start the virtual machine to be evaluated.
- If something doesn't work as expected or the two signatures differ, the evaluation stops here.

Yes

No

Mandatory part

The project consists of creating and configuring a virtual machine following strict rules. The student being evaluated will have to help you during the defense. Make sure that all of the following points are observed.

Project overview

- The student being evaluated should explain to you simply:
 - How a virtual machine works.
 - Their choice of operating system.
 - The basic differences between Rocky and Debian.
 - The purpose of virtual machines.
 - If the evaluated student chose Rocky: what SELinux and DNF are.
 - If the evaluated student chose Debian: the difference between aptitude and apt, and what APPArmor is. During the defense, a script must display information all every 10 minutes. Its operation will be checked in detail later. If the explanations are not clear, the evaluation stops here.

Yes

No

Simple setup

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Ensure that the machine does not have a graphical environment at launch. A password will be requested before attempting to connect to this machine. Finally, connect with a user with the help of the student being evaluated. This user must not be root. Pay attention to the password chosen, it must follow the rules imposed in the subject.
- Check that the UFW service is started with the help of the evaluator.
- Check that the SSH service is started with the help of the evaluator.
- Check that the chosen operating system is Debian or Rocky with the help of the evaluator. If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

User

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

The subject requests that a user with the login of the student being evaluated is present on the virtual machine. Check that it has been added and that it belongs to the "sudo" and "user42" groups.

Make sure the rules imposed in the subject concerning the password policy have been put in place by following the following steps.

First, create a new user. Assign it a password of your choice, respecting the subject rules. The student being evaluated must now explain to you how they were able to set up the rules requested in the subject on their virtual machine.

Normally there should be one or two modified files. If there is any problem, the evaluation stops here.

- Now that you have a new user, ask the student being evaluated to create a group named "evaluating" in front of you and assign it to this user. Finally, check that this user belongs to the "evaluating" group.
- Finally, ask the student being evaluated to explain the advantages of this password policy, as well as the advantages and disadvantages of its implementation. Of course, answering that it is because the subject asks for it does not count.

If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

Hostname and partitions

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the hostname of the machine is correctly formatted as follows: login42 (login of the student being evaluated).
- Modify this hostname by replacing the login with yours, then restart the machine. If on restart, the hostname has not been updated, the evaluation stops here.
- You can now restore the machine to the original hostname.
- Ask the student being evaluated how to view the partitions for this virtual machine.
- Compare the output with the example given in the subject. Please note: if the student evaluated makes the bonuses, it will be necessary to refer to the bonus example.

This part is an opportunity to discuss the scores! The student being evaluated should give you a brief explanation of how LVM works and what it is all about.

If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

SUDO

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the "sudo" program is properly installed on the virtual machine.
- The student being evaluated should now show assigning your new user to the "sudo" group.
- The subject imposes strict rules for sudo. The student being evaluated must first explain the value and operation of sudo using examples of their choice. In a second step, it must show you the implementation of the rules imposed by the subject.
- Verify that the "/var/log/sudo/" folder exists and has at least one file. Check the contents of the files in this folder. You should see a history of the commands used with sudo. Finally, try to run a command via sudo. See if the file (s) in the "/var/log/sudo/" folder have been updated. If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

UFW / Firewalld

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the "UFW" (or "Firewalld" for rocky) program is properly installed on the virtual machine.
- Check that it is working properly.
- The student being evaluated should explain to you basically what UFW (or Firewalld) is and the value of using it.
- List the active rules in UFW (or Firewalld). A rule must exist for port 4242.
- Add a new rule to open port 8080. Check that this one has been added by listing the active rules.
- Finally, delete this new rule with the help of the student being evaluated. If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

SSH

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

- Check that the SSH service is properly installed on the virtual machine.
- Check that it is working properly.
- The student being evaluated must be able to explain to you basically what SSH is and the value of using it.
- Verify that the SSH service only uses port 4242.
- The student being evaluated should help you use SSH in order to log in with the newly created user. To do this, you can use a key or a simple password. It will depend on the student being evaluated. Of course, you have to make sure that you cannot use SSH with the "root" user as stated in the subject. If something does not work as expected or is not clearly explained, the evaluation stops here.

 Yes

 No

Script monitoring

Remember: Whenever you need help checking something, the student being evaluated should be able to help you.

The student being evaluated should explain to you simply:

- How their script works by showing you the code.
- What "cron" is.
- How the student being evaluated set up their script so that it runs every 10 minutes from when the server starts. Once the correct functioning of the script has been verified, the student being evaluated should ensure that this script runs every minute. You can run whatever you want to make sure the script runs with dynamic values correctly. Finally, the student being evaluated should make the script stop running when the server has started up, but without modifying the script itself. To check this point, you will have to restart the server one last time. At startup, it will be necessary to check that the script still exists in the same place, that its rights have remained unchanged, and that it has not been modified. If something does not work as expected or is not clearly explained, the evaluation stops here.

Yes

No

Bonus

Evaluate the bonus part if, and only if, the mandatory part has been entirely and perfectly done, and the error management handles unexpected or bad usage. In case all the mandatory points were not passed during the defense, bonus points must be totally ignored.

Bonus

Check, with the help of the subject and the student being evaluated, the bonus points authorized for this project:

- Setting up partitions is worth 2 points.
- Setting up WordPress, only with the services required by the subject, is worth 2 points.
- The free choice service is worth 1 point. Verify and test the proper functioning and implementation of each extra service. For the free choice service, the student being evaluated has to give you a simple explanation about how it works and why they think it is useful. Please note that NGINX and Apache2 are prohibited.

Rate it from 0 (failed) through 5 (excellent)



'9-1 Respuestas de la evaluación 100

' • Que es una maquina virtual ?

Es un software que simula un sistema de computación y puede ejecutar programas como si fuese una computadora real. Permite crear múltiples entornos simulados o recursos dedicados desde un solo sistema de hardware físico.

' • Porque has escogido Debian ?

Esto es algo personal para cada uno, mi opinión: El propio subject explica que es más sencillo hacerlo en Debian y si buscas documentación/tutoriales hay muchos y todos se han hecho en Debian.

' • Diferencias basicas entre Rocky y Debian

CentOS vs Debian

CentOS Debian

This section contains a large green header with the title 'CentOS vs Debian'. Below the header is a dark grey horizontal bar with the words 'CentOS' and 'Debian' in white. Underneath this bar are two light grey boxes containing small black bar charts. A vertical dashed line separates the two boxes.



CentOS is more stable and supported by a large community.

Debian has relatively less market preference.

CentOS



Mission critical servers are hosted on CentOS.

Debian



Ubuntu is fast catching up. A lot of people are betting on it.

CentOS



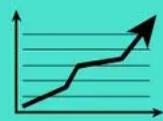
CentOS does not support many different architectures.

Debian



Debian has more packages.

CentOS



Both support desktop applications but CentOS has slight edge over Debian

Debian



Debian based distributors for more used for servers.

CentOS

Debian



CentOS versions are maintained for 10 years meaning there is great support for enterprise applications.



Certain amount of good Linux knowledge is required to work with Debian particularly to install new software and do customization.

CentOS



CentOS new versions are released usually after long gap and hence these systems are very stable. However, minor release does happen every now and then.

Debian



New versions of Debian are usually released with a 2 years gap so there is enough time to test and fix bugs. Hence these systems are more stable.

CentOS



After a major release, the CentOS code is frozen and is never changed except for security flaws or security bugs. This makes some issues while working with it as the next update usually happens after 5 years and many application software changes in this duration. For example, CentOS 5 supports MySQL 5.1 only where as there are newer versions of MySQL available which CentOS does not support.

Debian



Due to rapid development and short testing cycle, most major vendors still prefer CentOS over Debian. For example Oracle or MySQL team prefer CentOS because these are more stable and thoroughly tested.

Most of the developers who build application software on Linux uses Ubuntu as the desktop and still uses CentOS as servers.

CentOS



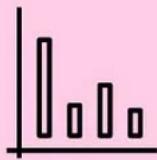
Usually it is very difficult to upgrade a version of CentOS locally. Official sources recommend installing a newer version than to upgrade an older one.

Debian



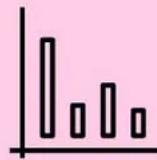
Upgrading Debian from one stable version to another is easy and not painful.

CentOS



It does not have an easy GUI.

Debian



It has desktop friendly applications and GUI.

CentOS



CentOS is released so late that sometimes it lags the Red Hat release.

Debian



Most people's opinion is that Debian systems are not as stable or trouble free as RHEL/CentOS.

CentOS



core software of CentOS such as the RHEL/CentOS components, also the kernel and all its utilities come from the distribution while the add-on software like Apache, PHP, Java, and MySQL come from newer sources such as Fedora or from vendors directly such as MySQL.

Debian



real reason to use Debian is if they provide unique functionality that is necessary for a system in such cases switching to Debian makes sense. apt repositories in package managers has latest source code for several open source languages and frameworks like ruby, rails, PostgreSQL, Golang, selenium, angular2-dart etc.ubuntu is very suitable to work with when using Docker file/s docker containers.

CentOS

Debian



Crashers are so rare in CentOS.



Same goes here.

• Cual es el proposito de las maquinas virtuales ?

Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los componentes de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.

• Diferencias entre apt y aptitude ↴

Aptitude es una versión mejorada de apt. APT es un administrador de paquetes de nivel inferior y aptitude es un administrador de paquetes de nivel superior. Otra gran diferencia es la funcionalidad que ofrecen ambas herramientas. Aptitude ofrece una mejor funcionalidad en comparación con APT. Ambos son capaces de proporcionar los medios necesarios para realizar la gestión de paquetes. Sin embargo, si se busca un enfoque con más características, debería ser, Aptitude.

• Que es APPArmor ?

Es un módulo de seguridad del kernel Linux que permite al administrador del sistema restringir las capacidades de un programa.

• Que es LVM ?

Es un gestor de volúmenes lógicos. Proporciona un método para asignar espacio en dispositivos de almacenamiento masivo, que es más flexible que los esquemas de particionado convencionales para almacenar volúmenes.

• 9-2 Comandos de la evaluación 🖥

1 ° Comprobar que no haya ninguna interfaz gráfica en uso.

Utilizaremos el comando `ls /usr/bin/*session` y nos debe aparecer el mismo resultado que en la captura. Si aparece algo diferente se está utilizando una interfaz gráfica.

```
gemartin@gemartin42:~$ ls /usr/bin/*session
/usr/bin/dbus-run-session
gemartin@gemartin42:~$
```

2 ° Comprobar que el servicio UFW está en uso.

```
sudo ufw status
```

```
root@gemartin42:/home/gemartin# sudo ufw status
Status: active
To                         Action      From
--                         --          --
4242                       ALLOW      Anywhere
80                        ALLOW      Anywhere
4242 (v6)                   ALLOW      Anywhere (v6)
80 (v6)                     ALLOW      Anywhere (v6)
root@gemartin42:/home/gemartin#
```

```
sudo service ufw status
```

```
root@gemartin42:/home/gemartin# sudo service ufw status
● ufw.service - Uncomplicated firewall
  Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
  Active: active (exited) since Thu 2022-11-24 01:19:28 CET; 5min ago
    Docs: man:ufw(8)
 Process: 316 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
 Main PID: 316 (code=exited, status=0/SUCCESS)
    CPU: 41ms

Nov 24 01:19:28 gemartin42 systemd[1]: Finished Uncomplicated firewall.
Warning: journal has been rotated since unit was started, output may be incomplete.
root@gemartin42:/home/gemartin# _
```

3º Comprobar que el servicio SSH esta en uso.

```
sudo service ssh status
```

```
root@gemartin42:/home/gemartin# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-11-24 01:19:30 CET; 7min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
 Process: 552 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 613 (sshd)
   Tasks: 1 (limit: 1127)
  Memory: 3.8M
     CPU: 18ms
    CGroup: /system.slice/ssh.service
             └─613 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Nov 24 01:19:30 gemartin42 systemd[1]: Starting OpenBSD Secure Shell server...
Nov 24 01:19:30 gemartin42 sshd[613]: Server listening on 0.0.0.0 port 4242.
Nov 24 01:19:30 gemartin42 sshd[613]: Server listening on :: port 4242.
Nov 24 01:19:30 gemartin42 systemd[1]: Started OpenBSD Secure Shell server.
root@gemartin42:/home/gemartin# _
```

4º Comprobar que utilizas el sistema operativo Debian o Centos.

```
uname -v o uname --kernel-version
```

```
root@gemartin42:~# uname -v
#1 SMP Debian 5.10.149-2 (2022-10-21)
root@gemartin42:~# _
```

5º Comprobar que tu usuario este dentro de los grupos "sudo" y "user42".

```
getent group sudo
```

```
getent group user42
```

```
root@gemartin42:~# getent group sudo
sudo:x:27:gemartin
root@gemartin42:~# getent group user42
user42:x:1001:gemartin
root@gemartin42:~# _
```

6º Crear un nuevo usuario y mostrar que sigue la politica de contraseñas que hemos creado.

```
sudo adduser name_user y introducimos una contraseña que siga la politica.
```

```
root@gemartin42:~# sudo adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1002) ...
Adding new user `newuser' (1001) with group `newuser' ...
Creating home directory `/home/newuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
      Full Name []: _
```

7º Creamos un nuevo grupo llamado "evaluating".

```
sudo addgroup evaluating
```

```
root@gemartin42:~# sudo addgroup evaluating
Adding group `evaluating' (GID 1003) ...
Done.
root@gemartin42:~# _
```

8º Añadimos el nuevo usuario al nuevo grupo.

```
sudo adduser name_user evaluating
```

```
root@gemartin42:~# sudo adduser newuser evaluating
Adding user `newuser' to group `evaluating' ...
Adding user newuser to group evaluating
Done.
root@gemartin42:~# _
```

Para comprobar que se haya introducido correctamente.

```
root@gemartin42:~# getent group evaluating
evaluating:x:1003:newuser
root@gemartin42:~#
```

9º Comprobar que el hostname de la maquina es correcto login42.

```
root@gemartin42:~# hostname
gemartin42
root@gemartin42:~#
```

10º Modificar hostname para remplazar tu login por el del evaluador. En este caso lo reemplazare por student42.

```
sudo nano /etc/hostname y remplazamos nuestro login por el nuevo.
```

```
root@gemartin42:/home/gemartin# sudo nano /etc/hostname
```

```
GNU nano 5.4                               /etc/hostname *
student42
```

sudo nano /etc/hosts y remplazamos nuestro login por el nuevo.

```
root@gemartin42:/home/gemartin# sudo nano /etc/hosts
```

```
GNU nano 5.4                               /etc/hosts *
127.0.0.1      localhost
127.0.1.1      student42

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

Reiniciamos la maquina.

```
root@gemartin42:/home/gemartin# sudo reboot_
```

Una vez nos hemos logueado de nuevo podemos ver como el hostname se ha cambiado correctamente.

```
gemartin@student42:~$ hostname
student42
gemartin@student42:~$
```

11º Comprobar que todas las particiones son como indica el subject.

```
lsblk
```

```
gemartin@gemartin42:~$ lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda            8:0    0   80G  0 disk 
├─sda1         8:1    0  476M  0 part /boot
├─sda2         8:2    0   1K  0 part 
└─sda5         8:5    0 29.5G  0 part 
    └─sda5_crypt 254:0  0 29.5G  0 crypt
        ├─LVMGroup-root 254:1  0  9.3G  0 lvm   /
        ├─LVMGroup-swap 254:2  0  2.1G  0 lvm   [SWAP]
        ├─LVMGroup-home 254:3  0  4.7G  0 lvm   /home
        ├─LVMGroup-var  254:4  0  2.8G  0 lvm   /var
        ├─LVMGroup-srv  254:5  0  2.8G  0 lvm   /srv
        ├─LVMGroup-tmp  254:6  0  2.8G  0 lvm   /tmp
        └─LVMGroup-var--log 254:7  0  3.7G  0 lvm   /var/log
sr0           11:0   1 1024M  0 rom
```

12º Comprobar que sudo esta instalado.

```
which sudo
```

```
gemartin@gemartin42:~$ which sudo  
/usr/bin/sudo  
gemartin@gemartin42:~$ _
```

Utilizar which realmente no es una buena practica ya que no todos los paquetes se encuentran en las rutas donde which busca, aun asi para la evaluacion es mejor ya que es un comando sencillo y facil de aprender. Para un mejor uso haremos uso del siguiente comando:

```
dpkg -s sudo
```

```
gemartin@gemartin42:~$ dpkg -s sudo  
Package: sudo  
Status: install ok installed  
Priority: optional  
Section: admin  
Installed-Size: 4589  
Maintainer: Sudo Maintainers <sudo@packages.debian.org>  
Architecture: amd64  
Version: 1.9.5p2-3  
Replaces: sudo-ldap  
Depends: libaudit1 (>= 1:2.2.1), libc6 (>= 2.27), libpam0g (>= 0.99.7.1), libselinux1 (>= 3.1~), zlib1g (>= 1:1.2.0.2), libpam-modules, lsb-base  
Conflicts: sudo-ldap  
Conffiles:  
/etc/init.d/sudo 1153f6e6fa7c0e2166779df6ad43f1a8  
/etc/pam.d/sudo 85da64f888739f193fc0fa896680030e  
/etc/sudo.conf cdb3df319152dbf3a1ccab9d5bd01ad0  
/etc/sudo_logsrvd.conf 8f2d34058527c9b8155de178aacff2cd  
/etc/sudoers b1f89c8342752a2a29bc5a3f8fd70437  
/etc/sudoers.d/README 8d3cf36d1713f40a0ddc38e1b21a51b6  
Description: Provide limited super user privileges to specific users  
Sudo is a program designed to allow a sysadmin to give limited root  
privileges to users and log root activity. The basic philosophy is to give  
as few privileges as possible but still allow people to get their work done.  
This version is built with minimal shared library dependencies, use the  
sudo-ldap package instead if you need LDAP support for sudoers.  
Homepage: https://www.sudo.ws/  
gemartin@gemartin42:~$
```

13 ° Introducimos el nuevo usuario dentro del grupo sudo.

```
sudo adduser name_user sudo
```

```
root@gemartin42:/home/gemartin# sudo adduser newuser sudo  
Adding user `newuser' to group `sudo' ...  
Adding user newuser to group sudo  
Done.  
root@gemartin42:/home/gemartin# _
```

Comprobamos que esta dentro del grupo.

```
root@gemartin42:/home/gemartin# getent group sudo  
sudo:x:27:gemartin,newuser  
root@gemartin42:/home/gemartin# _
```

14 ° Muestra la aplicacion de las reglas impuestas para sudo por el subject.

```
root@gemartin42:/var/log/sudo# nano /etc/sudoers.d/sudo_config
```

```
GNU nano 5.4                               /etc/sudoers.d/sudo_config  
Defaults passwd_tries=3  
Defaults badpass_message="Clave incorrecta"  
Defaults logfile="/var/log/sudo_config"  
Defaults log_input, log_output  
Defaults iolog_dir="/var/log/sudo"  
Defaults requiretty  
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
```

15 ° Muestra que la ruta /var/log/sudo/ existe y contiene almenos un fichero, en este se deberia ver un historial de los comandos utilizados con

```
root@gemartin42:/var/log/sudo# cd  
root@gemartin42:~# cd /var/log/sudo  
root@gemartin42:/var/log/sudo# ls  
00  seq  sudo_config  
root@gemartin42:/var/log/sudo#
```

```
root@gemartin42:/var/log/sudo# cat sudo_config
Nov 24 05:16:28 : root : TTY=tty1 ; PWD=/var/log/sudo ; USER=root ; TSID=00001W
; COMMAND=/usr/bin/nano 00
Nov 24 05:17:21 : root : TTY=tty1 ; PWD=/var/log/sudo ; USER=root ; TSID=00001X
; COMMAND=/usr/bin/nano hellogithub
root@gemartin42:/var/log/sudo#
```

Ejecuta un comando con sudo y comprueba que se actualiza el fichero.

```
root@gemartin42:/var/log/sudo# sudo nano hello42world
```

```
root@gemartin42:/var/log/sudo# cat sudo_config
Nov 24 05:16:28 : root : TTY=tty1 ; PWD=/var/log/sudo ; USER=root ; TSID=00001W
; COMMAND=/usr/bin/nano 00
Nov 24 05:17:21 : root : TTY=tty1 ; PWD=/var/log/sudo ; USER=root ; TSID=00001X
; COMMAND=/usr/bin/nano hellogithub
Nov 24 05:23:10 : root : TTY=tty1 ; PWD=/var/log/sudo ; USER=root ; TSID=00001Y
; COMMAND=/usr/bin/nano hello42world
root@gemartin42:/var/log/sudo#
```

16. Comprueba que el programa UFW esta instalado en la maquina virtual y comprueba que funciona correctamente.

```
dpkg -s ufw
```

```
root@gemartin42:~# dpkg -s ufw
Package: ufw
Status: install ok installed
Priority: optional
Section: admin
Installed-Size: 837
Maintainer: Jamie Strandboge <jamie@ubuntu.com>
Architecture: all
Version: 0.36-7.1
Depends: iptables, lsb-base (>= 3.0-6), ucf, python3:any, debconf (>= 0.5) | debconf-2.0
Suggests: rsyslog
Conffiles:
 /etc/default/ufw a921dd9d167380b04de4bc911915ea44
 /etc/init.d/ufw 4156943ab8a824fcf4b04cc1362eb230
 /etc/logrotate.d/ufw 12b1fb7ce76fc46f161e1ead1a22ce6
 /etc/rsyslog.d/20-ufw.conf 98e2f72c9c65ca8d6299886b524e80d1
 /etc/ufw/applications.d/ufw-bittorrent d9451245a3fb2aa85ed91533ce530f27
 /etc/ufw/applications.d/ufw-chat 73204a7a2819499d7802bc83b7e63ee9
 /etc/ufw/applications.d/ufw-directoryserver 28888bb4f7fa81ea2ca23bb86995df5b
 /etc/ufw/applications.d/ufw-dnsserver 7a2634d40515a5baab2d5b355873e1e6
 /etc/ufw/applications.d/ufw-fileserver d43adc11063000fc3cia824071382047
 /etc/ufw/applications.d/ufw-loginservice 366b3845c4360ea626f78875a400446b
 /etc/ufw/applications.d/ufw-mailserver 37e7910a1da915bcf60dac1c2d157377
 /etc/ufw/applications.d/ufw-printserver 47e009dc96a9eac7b3f2c2483a889756
 /etc/ufw/applications.d/ufw-proxyserver 6e035b6921d41aeee89c3d5867c593c5
 /etc/ufw/applications.d/ufw-webserver 07a41595f0b2c9865b7220bea998f8cf
 /etc/ufw/sysctl.conf 7723079fc108eda8f57eddab3079c70a
Description: program for managing a Netfilter firewall
The Uncomplicated Firewall is a front-end for iptables, to make managing a
Netfilter firewall easier. It provides a command line interface with syntax
similar to OpenBSD's Packet Filter. It is particularly well-suited as a
host-based firewall.
Homepage: https://launchpad.net/ufw
root@gemartin42:~#
```

```
sudo service ufw status
```

```
root@gemartin42:~# sudo service ufw status
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
   Active: active (exited) since Thu 2022-11-24 03:49:57 CET; 1h 35min ago
     Docs: man:ufw(8)
   Process: 315 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
 Main PID: 315 (code=exited, status=0/SUCCESS)
    CPU: 43ms

Nov 24 03:49:57 gemartin42 systemd[1]: Finished Uncomplicated firewall.
Warning: journal has been rotated since unit was started, output may be incomplete.
root@gemartin42:~#
```

17. Lista las reglas activas en UFW si no esta hecha la parte bonus solo debe aparecer la regla para el puerto 4242.

```
sudo ufw status numbered
```

```
root@gemartin42:~# sudo ufw status numbered
Status: active

      To          Action    From
      --          ----
[ 1] 4242        ALLOW IN  Anywhere
[ 2] 80          ALLOW IN  Anywhere
[ 3] 4242 (v6)  ALLOW IN  Anywhere (v6)
[ 4] 80 (v6)    ALLOW IN  Anywhere (v6)

root@gemartin42:~#
```

18 ° Crea una nueva regla para el puerto 8080. Comprueba que se ha añadido a las reglas activas y acto seguido puedes borrarla.

`sudo ufw allow 8080` para crearla

```
root@gemartin42:~# sudo ufw allow 8080
Rule added
Rule added (v6)
root@gemartin42:~# _
```

`sudo ufw status numbered`

```
root@gemartin42:~# sudo ufw status numbered
Status: active

      To          Action    From
      --          ----
[ 1] 4242        ALLOW IN  Anywhere
[ 2] 80          ALLOW IN  Anywhere
[ 3] 8080        ALLOW IN  Anywhere
[ 4] 4242 (v6)  ALLOW IN  Anywhere (v6)
[ 5] 80 (v6)    ALLOW IN  Anywhere (v6)
[ 6] 8080 (v6)  ALLOW IN  Anywhere (v6)

root@gemartin42:~# _
```

Para borrar la regla debemos utilizar el comando `sudo ufw delete num_rule`

```
root@gemartin42:~# sudo ufw delete 3
Deleting:
allow 8080
Proceed with operation (y|n)? y
Rule deleted
root@gemartin42:~# _
```

Comprobamos que se ha eliminado y vemos el numero de la siguiente regla que hay que borrar.

```
root@gemartin42:~# sudo ufw status numbered
Status: active

      To          Action    From
      --          ----
[ 1] 4242        ALLOW IN  Anywhere
[ 2] 80          ALLOW IN  Anywhere
[ 3] 4242 (v6)  ALLOW IN  Anywhere (v6)
[ 4] 80 (v6)    ALLOW IN  Anywhere (v6)
[ 5] 8080 (v6)  ALLOW IN  Anywhere (v6)

root@gemartin42:~# _
```

Borramos de nuevo la regla.

```
root@gemartin42:~# sudo ufw delete 5
Deleting:
allow 8080
Proceed with operation (y|n)? y
Rule deleted (v6)
root@gemartin42:~# _
```

Comprobamos que solo nos quedan las reglas requeridas en el subject.

```
root@gemartin42:~# sudo ufw status numbered
Status: active

      To          Action    From
      --          ----
[ 1] 4242        ALLOW IN  Anywhere
[ 2] 80          ALLOW IN  Anywhere
[ 3] 4242 (v6)  ALLOW IN  Anywhere (v6)
[ 4] 80 (v6)    ALLOW IN  Anywhere (v6)

root@gemartin42:~# _
```

19 ° Comprueba que el servicio ssh esta instalado en la maquina virtual, que funciona correctamente y que solo funciona por el puerto 4242.

```
which ssh
```

```
root@gemartin42:~# which ssh  
/usr/bin/ssh
```

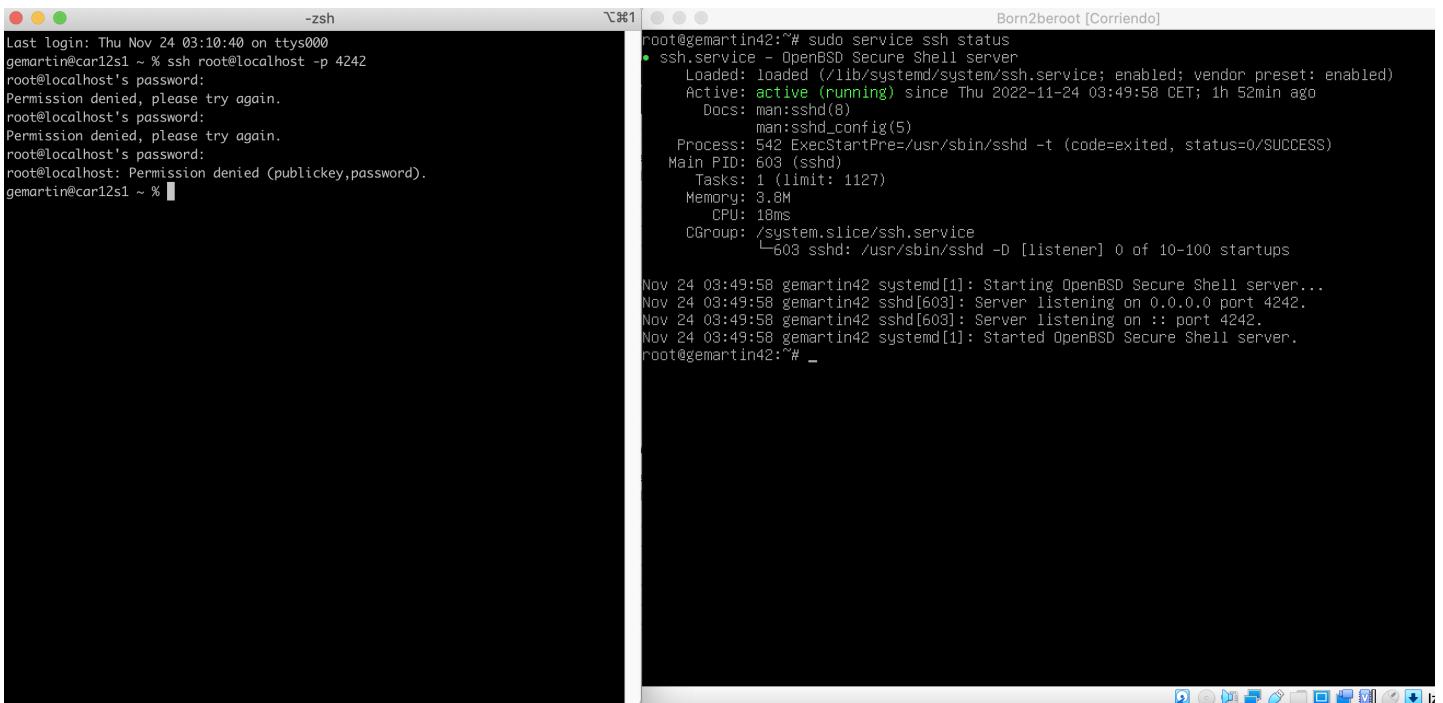
```
sudo service ssh status
```

```
root@gemartin42:~# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-11-24 03:49:58 CET; 1h 48min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 542 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 603 (sshd)
   Tasks: 1 (limit: 1127)
  Memory: 3.8M
     CPU: 18ms
    CGroup: /system.slice/ssh.service
            └─603 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Nov 24 03:49:58 gemartin42 systemd[1]: Starting OpenBSD Secure Shell server...
Nov 24 03:49:58 gemartin42 sshd[603]: Server listening on 0.0.0.0 port 4242.
Nov 24 03:49:58 gemartin42 sshd[603]: Server listening on :: port 4242.
Nov 24 03:49:58 gemartin42 systemd[1]: Started OpenBSD Secure Shell server.
root@gemartin42:~#
```

20. Usa ssh para iniciar sesión con el usuario recién creado. Asegurate de que no puede usar ssh con el usuario root.

Intentamos conectarnos por ssh con el usuario root pero no tenemos permisos.



Nos conectamos por ssh con el nuevo usuario con el comando `ssh newuser@localhost -p 4242`

```

newuser@gemartin42: ~
gemartin@car12s1 ~ % ssh newuser@localhost -p 4242
newuser@localhost's password:
Linux gemartin42 5.10.0-19- amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
newuser@gemartin42:~$ [REDACTED]

```

```

Born2beroot [Corriendo]
root@gemartin42:~# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-11-24 03:49:58 CET; 1h 57min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 542 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 603 (sshd)
   Tasks: 1 (limit: 1127)
     Memory: 5.3M
        CPU: 238ms
      CGroup: /system.slice/ssh.service
              └─603 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Nov 24 05:49:52 gemartin42 sshd[1169]: PAM 2 more authentication failures; logname= uid=0 e
Nov 24 05:49:52 gemartin42 sshd[1172]: Accepted password for gemartin from 10.0.2.2 port 54
Nov 24 05:49:56 gemartin42 sshd[1172]: pam_unix(sshd:session): session opened for user gema
Nov 24 05:49:58 gemartin42 sshd[1183]: pam_unix(sshd:auth): authentication failure: logname
Nov 24 05:49:58 gemartin42 sshd[1183]: Failed password for newuser from 10.0.2.2 port 54641
Nov 24 05:46:03 gemartin42 sshd[1183]: Failed password for newuser from 10.0.2.2 port 54641
Nov 24 05:46:04 gemartin42 sshd[1183]: Connection closed by authenticating user newuser 10.
Nov 24 05:46:04 gemartin42 sshd[1183]: PAM 1 more authentication failure; logname= uid=0 eu
Nov 24 05:47:05 gemartin42 sshd[1187]: Accepted password for newuser from 10.0.2.2 port 546
Nov 24 05:47:05 gemartin42 sshd[1187]: pam_unix(sshd:session): session opened for user newu
lines 1-23/23 (END)

```

21 ° Modifica el tiempo de ejecución del script de 10 minutos a 1.

Ejecutamos el siguiente comando para así modificar el fichero crontab `sudo crontab -u root -e`

```
root@gemartin42:/home/gemartin# sudo crontab -u root -e
```

Modificamos el primer parámetro , en vez de 10 lo cambiamos a 1.

```

GNU nano 5.4                               /tmp/crontab.xTOGMU/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/1 * * * * sh /home/gemartin/monitoring.sh

```

22 ° Finalmente haz que el script deje de ejecutarse cuando el servidor se haya iniciado, pero sin modificar el script.

```
sudo /etc/init.d/cron stop
```

```
root@gemartin42:/home/gemartin# sudo /etc/init.d/cron stop
Stopping cron (via systemctl): cron.service.
root@gemartin42:/home/gemartin# _
```

Si queremos que vuelva a ejecutarse:

```
sudo /etc/init.d/cron start
```

```
root@gemartin42:/home/gemartin# sudo /etc/init.d/cron start
Starting cron (via systemctl): cron.service.
root@gemartin42:/home/gemartin# _
```

10- Tester OK

Comprueba que no te hayas dejado nada! Tester propio para checkear que la instalación y configuración se ha realizado exitosamente.

[AQUÍ](#)

```
TEST CREATED BY: GEMARTIN

-----
Graphical environment
[OK]

Disk partitions
[OK]
[OK]
[OK]
[OK]

SSH
[OK]
[OK]

UFW
[OK]
[OK]

Hostname
[OK]

Password policy
1.[OK] minlen
2.[OK] uppercase
3.[OK] lowercase
4.[OK] digit
5.[OK] consecutive char
6.[OK] difok
7.[OK] enforce for root
8.[OK] reject username
9.[OK] passwd expire days
10.[OK] days allowed before the modification
11.[OK] warning message
12.[OK] folder /var/log/sudo exist

Crontab
[OK]
```

>Contacto ↴

>Contacta conmigo si crees que puedo mejorar el tutorial! Puede ayudar a futuros estudiantes! 😊

- Email: gemartin@student.42barcelona.com
- Linkedin: <https://www.linkedin.com/in/gemartin99/>

Quizás pueda interesarte!

- Para ver mi progresion en el common core 42 ↵

[AQUÍ](#)

- Mi perfil en la intranet de 42 ↵

[AQUÍ](#)