



Introducción

En este módulo vamos a simular escenarios reales donde apenas trabajaremos en local, en nuestro propio ordenador. Simularemos, mediante una máquina virtual que es en la que realmente trabajaremos, que todos nuestros despliegues ocurren en una máquina remota, tal y como ocurre en la realidad.

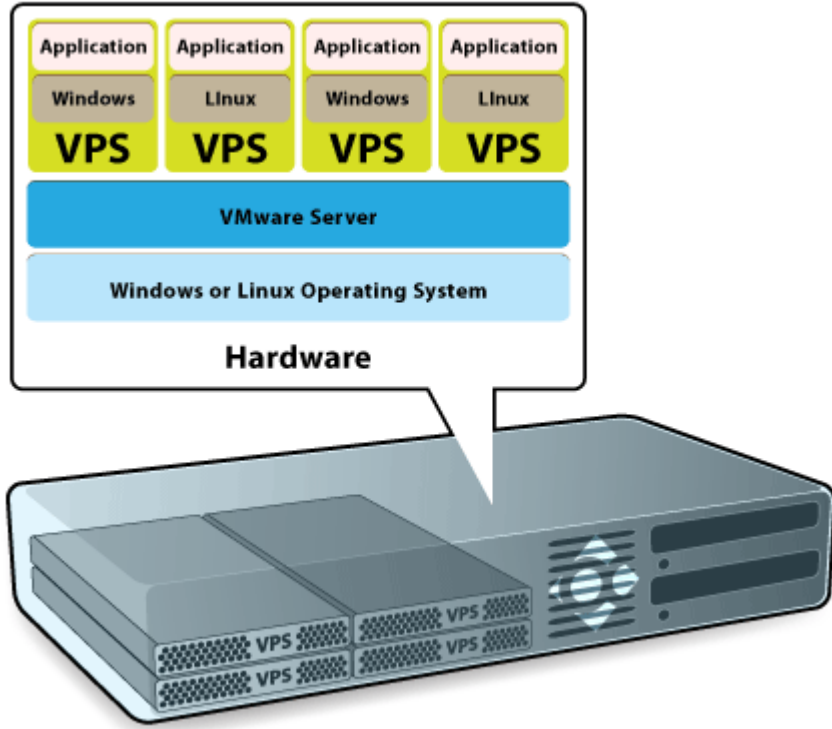
De hecho, se simulará un escenario donde tengamos contratado un VPS (Virtual Private Server) y debamos conectarnos de forma remota al mismo para poder trabajar. Un escenario muy común en el mundo real.

¿Qué es un VPS?

Un servidor es una computadora en la que tu proveedor de alojamiento web almacena los archivos y las bases de datos necesarios para tu sitio web. Cada vez que un visitante en línea quiere acceder a tu sitio web, su navegador le envía una solicitud a tu servidor y transfiere los archivos necesarios a través de Internet. El alojamiento VPS te proporciona un servidor en la nube que simula un servidor físico; sin embargo, en realidad, la máquina se comparte entre varios usuarios.

Al usar la tecnología de virtualización, tu proveedor de alojamiento web instala una capa virtual sobre el sistema operativo del servidor. Esta capa divide el servidor en particiones y le permite a cada usuario instalar su propio sistema operativo y software.

Por lo tanto, un servidor privado virtual (VPS) es tanto virtual como privado porque tienes control absoluto. Está separado de otros usuarios del servidor a nivel del sistema operativo. De hecho, la tecnología VPS es similar a la creación de particiones en tu computadora cuando quieres ejecutar más de un sistema operativo (por ejemplo, Windows y Linux) sin tener que reiniciar.



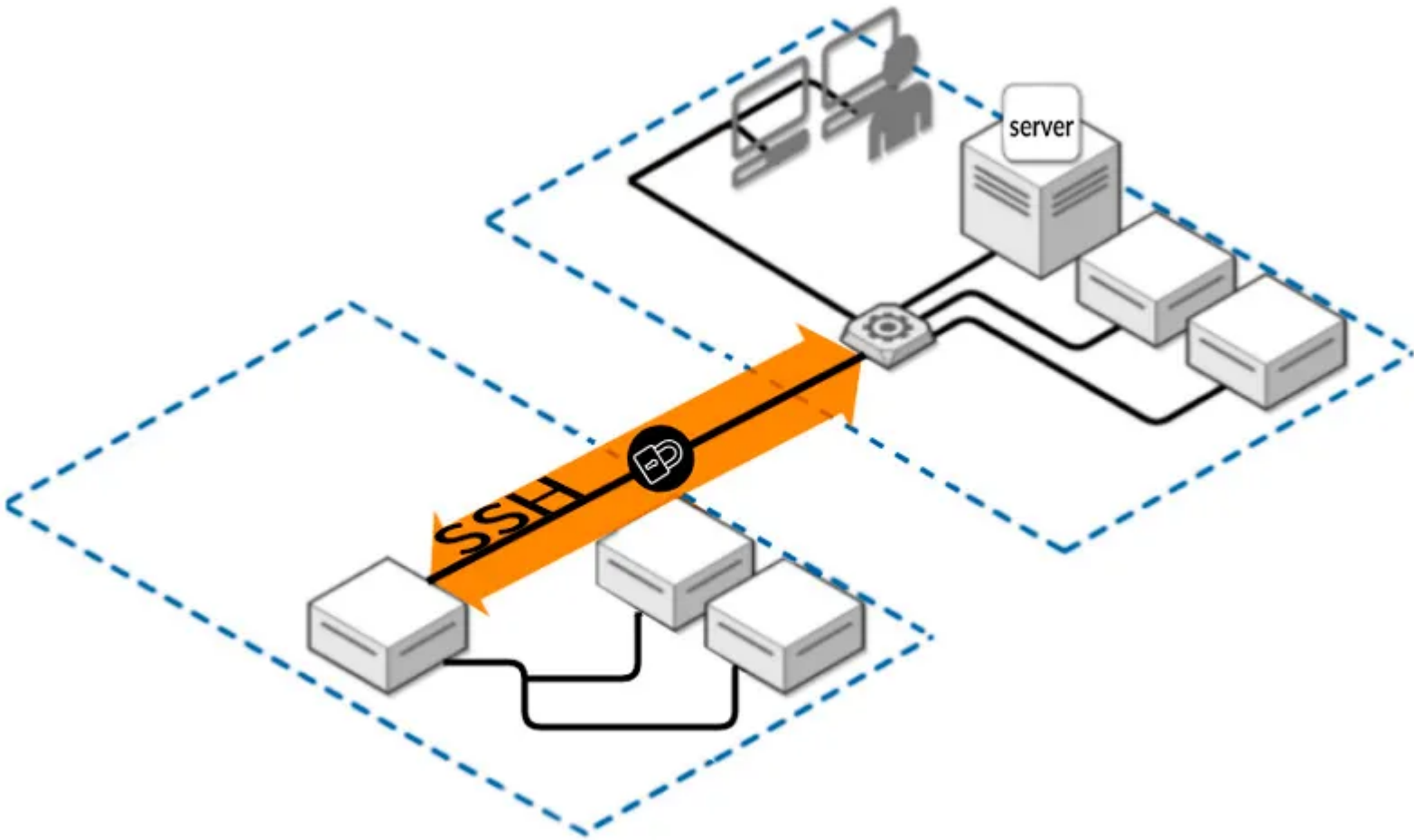
Un VPS te permite configurar tu sitio web dentro de un contenedor seguro con recursos garantizados (memoria, espacio en disco, núcleos de CPU, etc.) que no tienes que compartir con otros usuarios. Con el hosting VPS, tienes el mismo acceso de nivel raíz que si alquilaras un servidor dedicado, pero a un costo mucho más bajo.

El VPS es una solución más segura y estable que el hosting compartido, con el que no obtienes espacio de servidor dedicado. Sin embargo, es de menor escala y más barato que alquilar un servidor completo.

El hosting VPS generalmente es elegido por los propietarios de sitios web que tienen un tráfico de nivel medio que excede los límites de los planes de hosting compartido pero que aún no necesitan los recursos de un servidor dedicado.

Conexión mediante SSH

Aunque nuestra máquina virtual esté en nuestro ordenador, ya hemos dicho que estamos simulando un VPS remoto. Para conectarnos a una máquina de forma remota y segura, la opción más recomendable es SSH.



SSH o Secure Shell es un protocolo de red criptográfico para operar servicios de red de forma segura a través de una red no protegida. Las aplicaciones típicas incluyen línea de comandos remota, inicio de sesión y ejecución de comandos remota, pero cualquier servicio de red puede protegerse con SSH.

SSH proporciona un canal seguro a través de una red no segura mediante el uso de una arquitectura cliente-servidor, conectando una aplicación cliente SSH con un servidor SSH. El puerto TCP estándar para SSH es 22 y se usa generalmente para acceder a sistemas operativos similares a Unix, pero también se puede usar en Microsoft Windows.

Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente.

SSH tiene muchas aplicaciones diferentes:

- Gestión de servidores a los que no se puede acceder localmente
- Transferencia segura de archivos
- Creación de copias de seguridad
- Conexión entre dos ordenadores con encriptación de extremo a extremo
- Mantenimiento remoto desde otros ordenadores

Autenticación

Los dos métodos de autenticación de usuario SSH más comunes que se utilizan son las contraseñas (cifrado simétrico) y las claves SSH (cifrado asimétrico o de clave pública). Los clientes envían contraseñas cifradas al servidor de forma segura. Sin embargo, las contraseñas son un método de autenticación arriesgado porque su solidez depende de que el usuario sepa qué hace que una contraseña sea segura.

Los pares de claves pública-privada SSH encriptados asimétricamente son una mejor opción. Una vez que el cliente descifra el mensaje, el servidor le otorga acceso al sistema.

Es decir, SSH opta por el cifrado híbrido, donde se utiliza el cifrado asimétrico para intercambiar unas claves que serán las que se utilizarán posteriormente en el

intercambio de información.

Este tipo de cifrado utiliza la misma clave para cifrar y para descifrar la información. Por este motivo, la clave debe ser secreta y sólo conocida por el emisor y el receptor del mensaje.

Cifrados simétricos o de clave privada

Este tipo de cifrado utiliza la misma clave para cifrar y para descifrar la información. Por este motivo, la clave debe ser secreta y sólo conocida por el emisor y el receptor del mensaje.



Ventajas

- Muy rápidos → cifrar y descifrar un mensaje cada vez requiere un cierto tiempo, que si el algoritmo es complejo, puede ser elevado.

Inconvenientes

- Si alguien no autorizado consigue la clave, podrá espiar la comunicación sin problemas
- ¿Cómo hacemos para que emisor y receptor conozcan la clave en un primer momento? → no se puede transmitir por el canal inseguro → hay que transmitirla por otro canal seguro Ejemplos: PIN de la tarjeta del banco o archivo comprimido con contraseña

Cifrados asimétricos o de clave pública

En este tipo de cifrados cada usuario utiliza un par de claves: una clave pública y una clave privada. Un mensaje cifrado con la clave pública sólo se puede descifrar con su correspondiente clave privada y viceversa.



La *clave pública* es accesible a cualquier persona que quiera consultarla, no hace falta que sea transmitida por un canal seguro como en el caso anterior.

La *clave privada* sólo la debe conocer su dueño.

Funcionamiento:


1. El emisor cifra un mensaje con la clave pública del receptor
2. El receptor recibe el mensaje y es el único que podrá descifrarlo porque es el único que posee la clave cifrada asociada

Ventajas

- No se necesita un nuevo canal independiente y seguro para transmitir la clave

Inconvenientes

- Son más lentos que los cifrados simétricos
- Hay que proteger muy bien la clave privada y tenerla siempre disponible para poder descifrar los mensajes (no es una contraseña)
- Hay que asegurarse de que la clave pública es de quién dice ser y no de un impostor que se esté haciendo pasar por él

 **Nota**

Nosotros, para conectarnos por primera vez por SSH y comprobar la conectividad, utilizaremos el cifrado simétrico (una contraseña).

Tras ello, simulando un entorno real que aporte comodidad (no introducir contraseña cada vez que hagamos *login*) pero también y sobre todo, por seguridad, utilizaremos cifrado asimétrico. Esto es, un par de claves.

Referencias

¿Qué es un VPS? Todo lo que necesitas saber sobre servidores virtuales

Instalación y configuración de nuestra máquina virtual

Instalación de Debian

Como servidor, utilizaremos la distribución Linux Debian. Tal y como podemos leer en la [propia página de Debian](#):

Quote

Un CD de "instalación por red" o "netinst" es un único CD que posibilita que instale el sistema completo. Este único CD contiene sólo la mínima cantidad de software para instalar el sistema base y obtener el resto de paquetes a través de Internet.

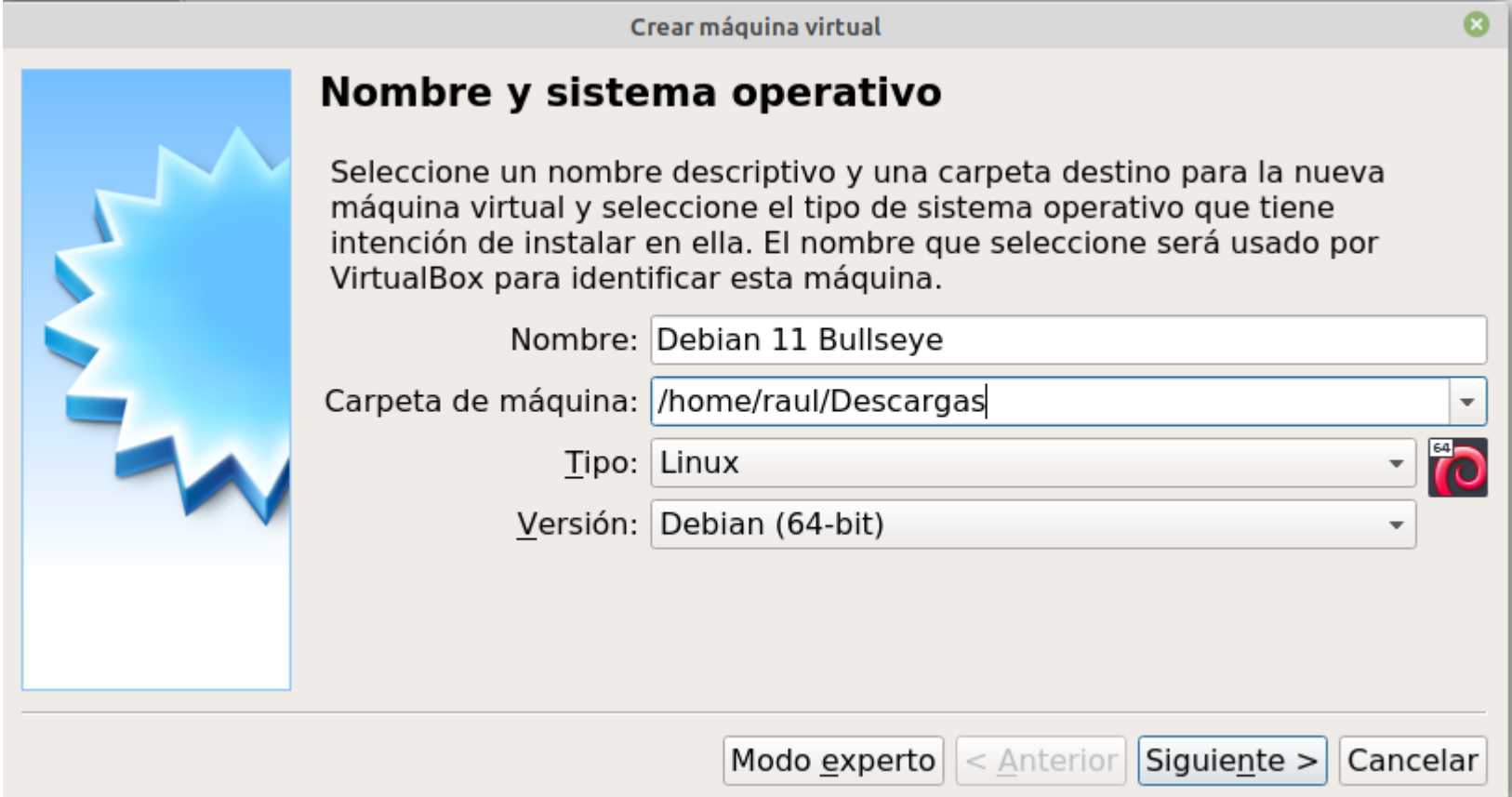
Así pues, procedamos a descargar la imagen de Debian netinstall [aquí](#)

Info

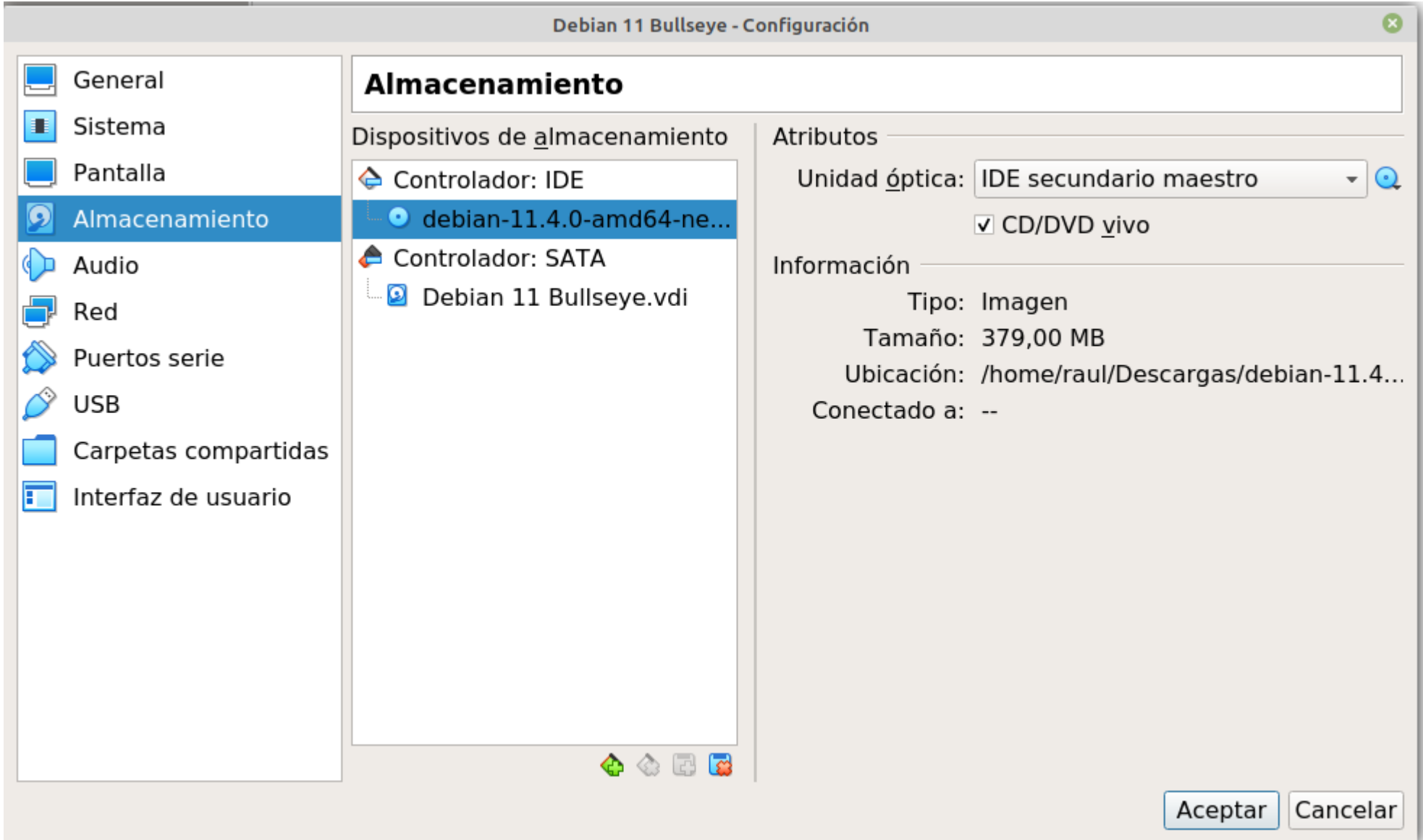
Queda muy lejos de la intención de este módulo explicar como instalar máquinas virtuales puesto que es algo que se supone aprendido del curso anterior y/u otros módulos. Así pues, se darán unas pautas generales para instalar la máquina correctamente.

La instalación de esta máquina virtual debería servir a lo largo del módulo utilizado cualquier hipervisor (VMWare, VirtualBox, KVM, HyperV...). No obstante, se utilizará Virtualbox para esta explicación.

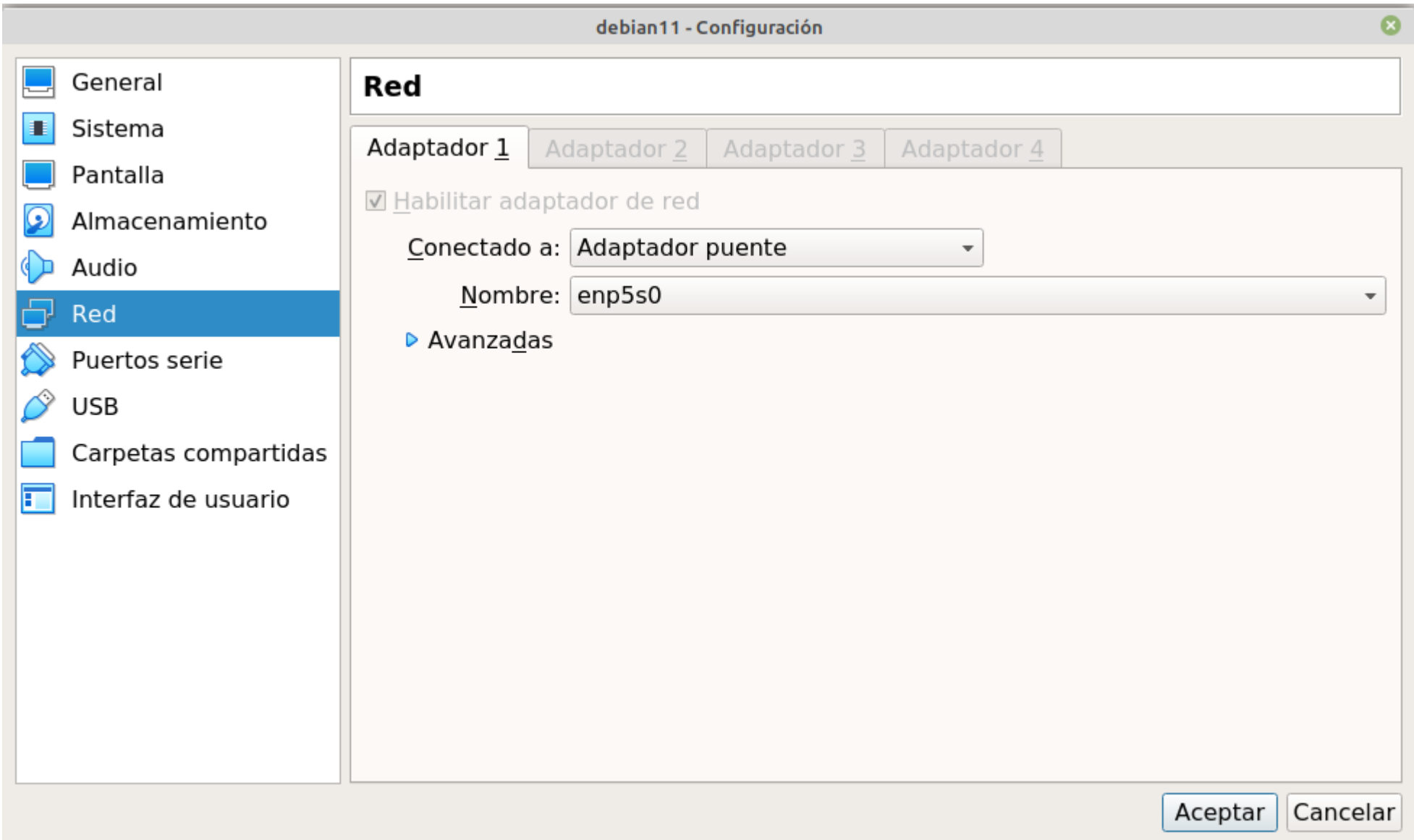
En primer lugar, debemos crear una máquina virtual nueva, indicando su ubicación, su nombre y el tipo de sistema operativo:



Le indicamos que monte como unidad de CD la iso de netinstall de Debian que hemos descargado previamente:

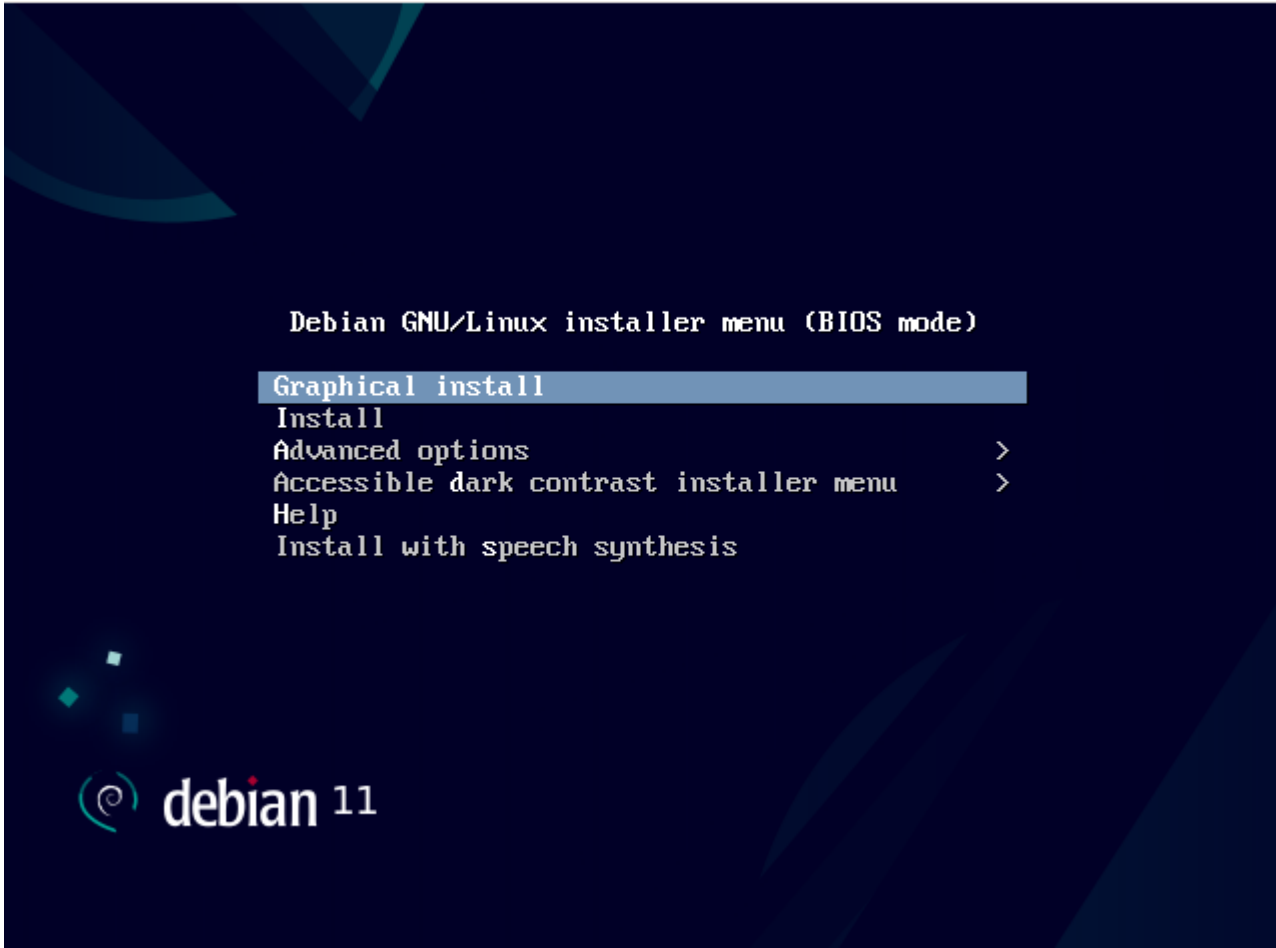


También estableceremos un único interfaz de red. Para ello, en el único adaptador de red que debe tener la máquina virtual, debemos configurarlo como tipo puente, de forma que obtenga una IP en el rango de la red local en la que nos encontremos conectados (casa, instituto...).




Sin entorno gráfico la máquina puede que funcione perfectamente con 1GB de RAM, no obstante se aconseja, si es posible, asignarle 2GB de RAM y, como mínimo, 2 procesadores.

Podéis instalar Debian tanto de forma gráfica como de forma clásica en terminal. La primera de ellas es la que os recomiendo:



Le dáis el nombre que queráis a vuestra máquina. Recomendable un nombre corto pues luego aparecerá en el *prompt* del terminal (`usuario@nombredemaquina`)

 **debian 11**

Configurar la red

Por favor, introduzca el nombre de la máquina.

El nombre de máquina es una sola palabra que identifica el sistema en la red. Consulte al administrador de red si no sabe qué nombre debería tener. Si está configurando una red doméstica puede inventarse este nombre.


Nombre de la máquina:

Capturar la pantalla

Retroceder

Continuar

Os pedirá también contraseña de superusuario (root), nombre de vuestro usuario y contraseña para este nuevo usuario:

 **debian 11**

Configurar usuarios y contraseñas

Se creará una cuenta de usuario para que la use en vez de la cuenta de superusuario en sus tareas que no sean administrativas.

Por favor, introduzca el nombre real de este usuario. Esta información se usará, por ejemplo, como el origen predeterminado para los correos enviados por el usuario o como fuente de información para los programas que muestren el nombre real del usuario. Su nombre completo es una elección razonable.


Nombre completo para el nuevo usuario:

Capturar la pantalla

Retroceder

Continuar

Tras ello, para simplificar nuestro proceso, le diremos que utilice todo el disco para la instalación:

 **debian 11**

Particionado de discos

Este instalador puede guiarle en el particionado del disco (utilizando distintos esquemas estándar) o, si lo desea, puede hacerlo de forma manual. Si escoge el sistema de particionado guiado tendrá la oportunidad más adelante de revisar y adaptar los resultados.

Se le preguntará qué disco a utilizar si elige particionado guiado para un disco completo.

Método de particionado:

Guiado - utilizar todo el disco

Guiado - utilizar el disco completo y configurar LVM

Guiado - utilizar todo el disco y configurar LVM cifrado

Manual

Capturar la pantalla

Retroceder

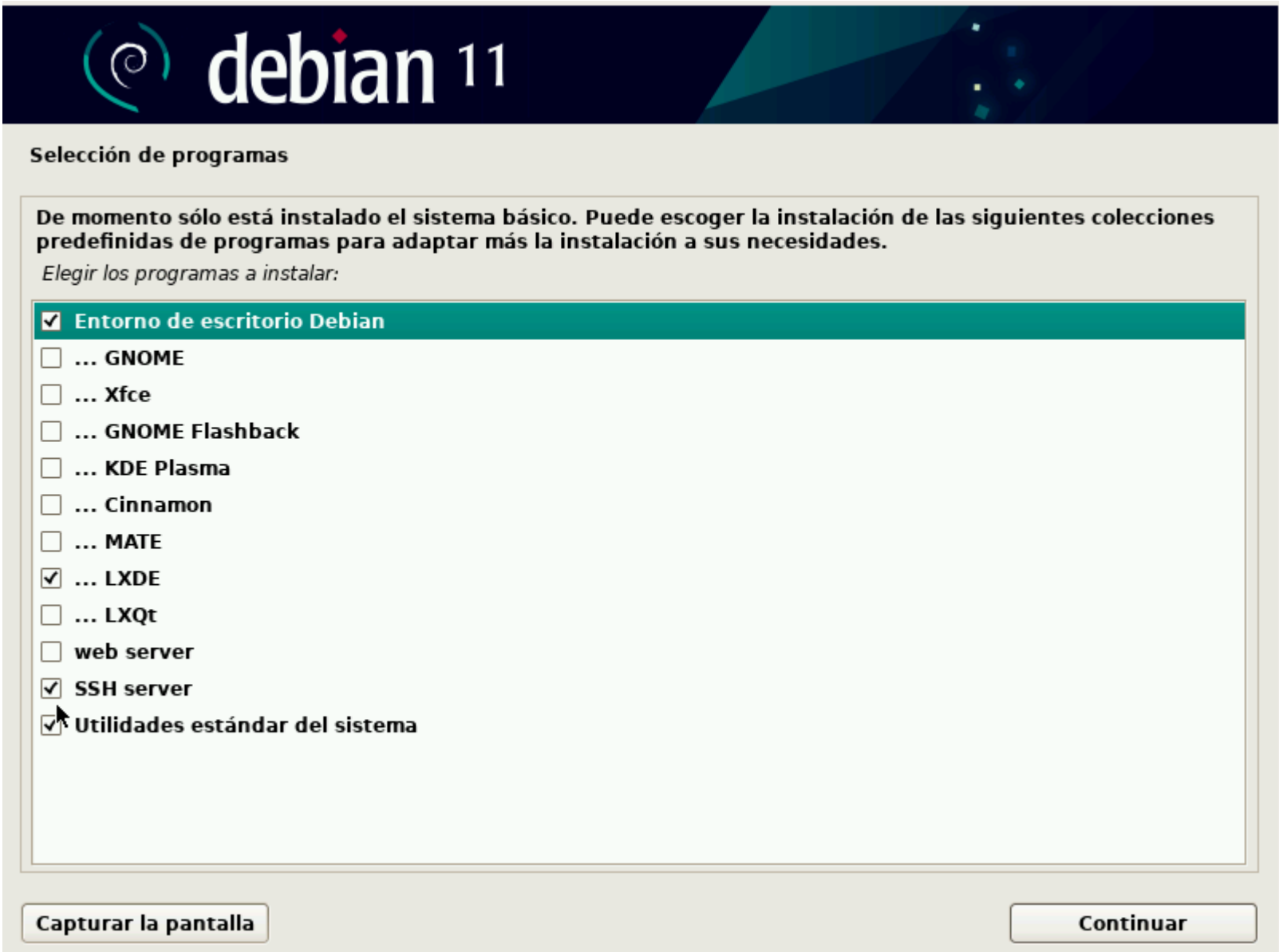
Continuar

E iremos dejando todas las opciones que nos vayan apareciendo por defecto y continuando la instalación.

Tras un rato, que puede ser más o menos largo, nos mostrará la opción de instalar un entorno gráfico. En principio no nos hace falta ninguno y esta es la opción recomendada por un tema de economización de los recursos.



Pero si por alguna razón queréis instalar alguno, os recomiendo LXDE puesto que es el que menos recursos consume:



También debéis marcar las opciones que aparecen en la imagen, **SSH Server** y **Utilidades estándar**.

Le indicamos que sí que instale el gestor de arranque GRUB y continuamos con todas las opciones por defecto:

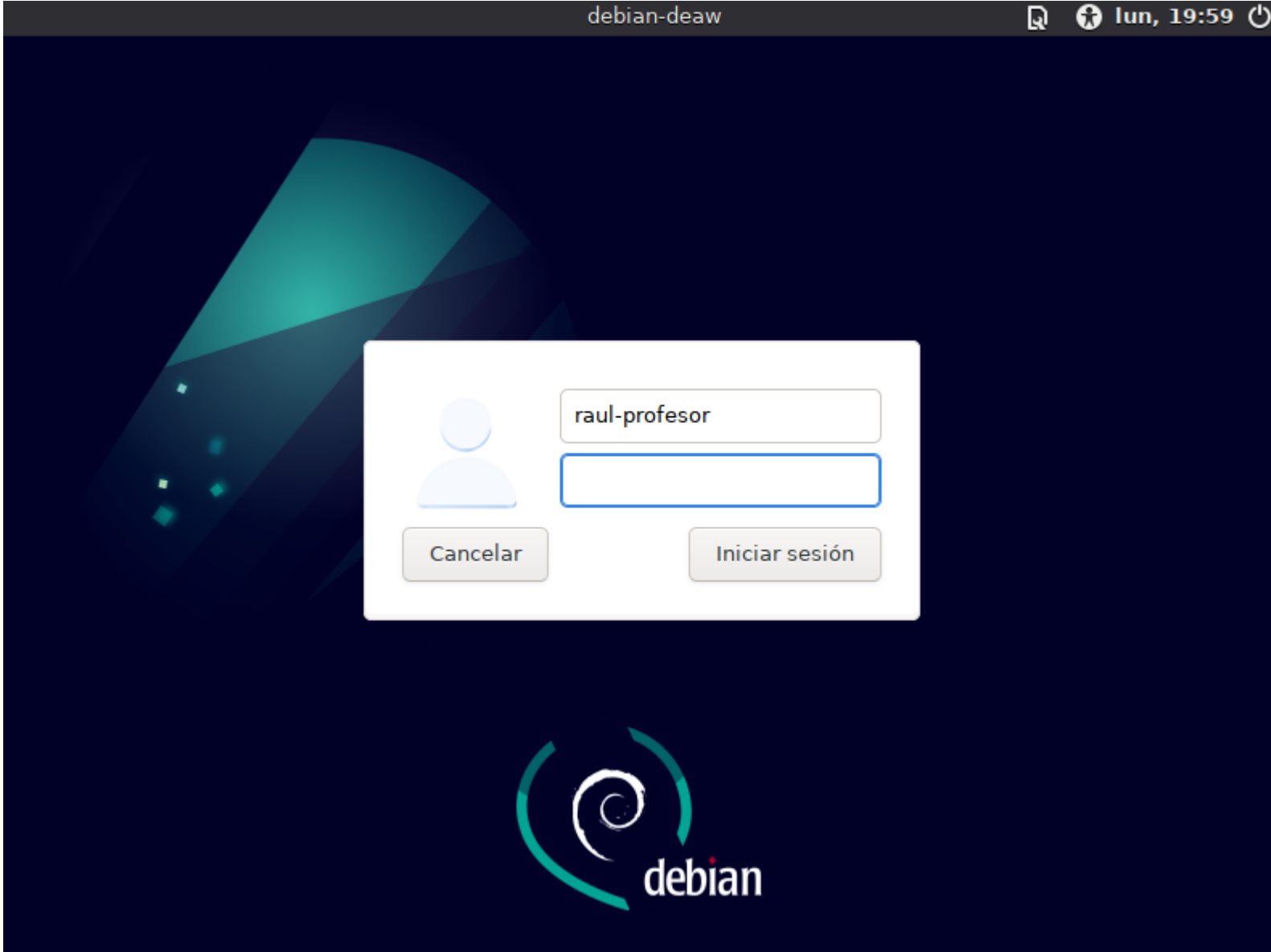


Y le indicamos que lo instale en el único disco que tenemos: `/dev/sda1` (pinchad en el nombre o no lo instalará ahí)



Completará el proceso v pedirá reiniciar, cosa que haréis. Tras ello, si no tenéis entorno gráfico aparecerá un terminal pidiendo login.

Si hubieráis instalado el entorno gráfico, os aparecerá algo así:



En ambos casos, introduciendo el nombre de usuario y contraseña podremos loguearnos en el sistema.

Dar permisos de sudo a nuestro usuario

Una vez instalada nuestra Debian, tendremos un usuario *raso* que es el que le dijimos que crease durante la instalación.

Puesto que a lo largo de este módulo realizaremos incontables tareas de administración, resulta un tanto incómodo, así como peligroso, el tener que cambiar de nuestro usuario a *root* cada vez que haya que instalar, configurar o modificar algo que así lo requiera.

Así pues, le daremos permisos de `sudo` a nuestro usuario. Estos permisos nos permitirán que cualquier comando que ejecutemos en el terminal precedido de la palabra `sudo` se ejecute como *root*. De la misma forma, cualquier comando que ejecutemos con nuestro usuario sin `sudo`, será ejecutado con los permisos de nuestro usuario, por lo que nos protegemos de *liarla* con un comando que no toca como *root*.

Dicho esto, hay varias formas de proceder, veamos la más típica y conocida. Se trata de modificar el archivo del sistema encargado de recoger estos permisos: `/etc/sudoers`.

En primer lugar debemos conectarnos por SSH a nuestra máquina Debian:

```
ssh -l nombre_de_usuario IP_MV_Debian
```

Donde:

- `nombre_de_usuario` es vuestro nombre de usuario (el que configurastéis durante la instalación)
- `IP_MV_Debian` es la IP de la máquina Debian

Info

Existen varias formas de conocer la IP de vuestra Debian pero quizás la más sencilla sea desde la propia máquina virtual, con el comando:

```
ip a
```

```
raul@debian-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:0c:29:2f:35:c5 brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   inet 192.168.18.205/24 brd 192.168.18.255 scope global ens33
       valid lft forever preferred_lft forever
   inet6 fe80::20c:29ff:fe2f:35c5/64 scope link
       valid lft forever preferred_lft forever
3: ens34: <NO-CARRIER,BROADCAST,MULTICAST,DYNAMIC,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
   link/ether 00:0c:29:2f:35:cf brd ff:ff:ff:ff:ff:ff
   altname enp2s2
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
   link/ether 02:42:18:82:e7:f3 brd ff:ff:ff:ff:ff:ff
raul@debian-server:~$
```

Ahí veo que esa IP está dentro del rango de mi red local. Además, puesto que la máquina sólo tiene una interfaz de red, no puede ser ninguna otra. Esa será la IP a la que conectarse.

Cambiando de nuestro usuario al usuario *root*:

```
su root
```

Ejecutamos la aplicación `visudo` que se encarga directamente de modificar el archivo de sudoers:

```
# /usr/sbin/visudo
```

Y dejamos el archivo así, claro está, con vuestro propio nombre de usuario:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
nombreusuario  ALL=(ALL:ALL) ALL
```

Pulsamos `CTRL+x` y guardamos los cambios.

Tras esto, debemos desloguearnos de nuestra sesión SSH y volver a loguearnos. Ahora podremos validar que ya podemos realizar acciones que requieran permisos de superusuario o *root*. Esta validación puede realizarse con el comando:

```
sudo -v
```

Que en caso de no tener permisos nos devolverá el siguiente mensaje:

```
Sorry, user [username] may not run sudo on [hostname].
```

Y en caso de tenerlos, no devolverá nada.

Si aún así no os quedase del todo claro, podéis utilizar este comando:

```
timeout 2 sudo id && echo Access granted || echo Access denied
```

Que, en caso de tener los permisos de sudo devuelve:

```
uid=0(root) gid=0(root) grupos=0(root)
Access granted
```

Y si no los tuviera, devuelve:

```
[username] is not in the sudoers file.  This incident will be reported.
Access denied
```

Configuración

En primer lugar nos crearemos nuestro par de claves, pública y privada, en el ordenador que se conectará a nuestra debian, con el comando (**sin sudo**):

```
ssh-keygen -b 4096
```

Si dejáis las opciones por defecto, creará una clave privada `id_rsa` y una clave pública `id_rsa.pub` en el directorio `/home/nombreusuario/.ssh`.

Os pedirá una contraseña para proteger el uso de la clave privada. Puesto que precisamente queremos agilizar el proceso de conexión por SSH para no introducir contraseñas, **debéis dejarla vacía**.

Una vez creado el par de claves, tal y como hemos visto en el apartado anterior, el servidor SSH (Debian) debe poseer nuestra clave pública para que podamos autenticarnos con nuestra clave privada, que como su nombre indica, sólo debemos poseer nosotros y por eso nos identifica unívocamente.

Este proceso de copia se puede realizar fácilmente con el comando:


```
ssh-copy-id usuario@ip_servidor
```

Para no tener ningún problema con los permisos sobre directorios y archivos, ejecutad en Debian:

```
chmod 700 .ssh/
chmod 600 .ssh/authorized_keys
```

Que no es más que una conexión SSH que además copia la clave, por lo que:

- usuario: nombre de vuestro usuario en Debian
- ip_servidor: ip de la máquina Debian

 **Para Windows**

Este módulo está diseñado desde un cliente Linux conectándose al servidor Linux, por lo que el cliente SSH está integrado en el propio terminal. Para Windows existen multitud de alternativas como cliente SSH, desde utilizar el propio WSL2 (Windows Subsystem Linux) de forma similar a lo que aquí se describe, hasta utilizar cualquier otro de los [varios clientes disponibles](#)

Por ejemplo, si utilizáis [Putty](#), deberéis seguir los pasos que detallan en [este tutorial](#) para configurar las claves.

En caso de utilizar otro cliente, buscad la forma de hacerlo pues diferirá en cada caso.

Referencias

[¿Qué es un VPS? Todo lo que necesitas saber sobre servidores virtuales](#)