

DÍA 7: INSTALACIÓN Y CONFIGURACIÓN DE LINUX SERVER

Por Daniel Ariza.

18/06/2025

Fase 1: Instalación del sistema base.

- **Instalar Ubuntu Server o Debian desde ISO en una máquina virtual.**

Configurar durante la instalación:

Zona horaria correcta

Nombre del host: srv-base-[nombreAlumno]

Usuario administrador personalizado con contraseña segura

Estos pasos se realizaron en el Reto del día 4 adjunto enlace donde puede ser consultado

<https://github.com/daniariza64/Pr-cticas>

- **Verificar que el sistema arranca sin errores y actualiza sus paquetes** (`apt update && apt upgrade`).



```
ubuntu3 (ip ubuntu) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Ubuntu 16.04.7 LTS ubuntu3 tty1

ubuntu3 login: daniel
Password:
Last login: Mon Jun 16 20:13:12 CEST 2025 on tty1
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

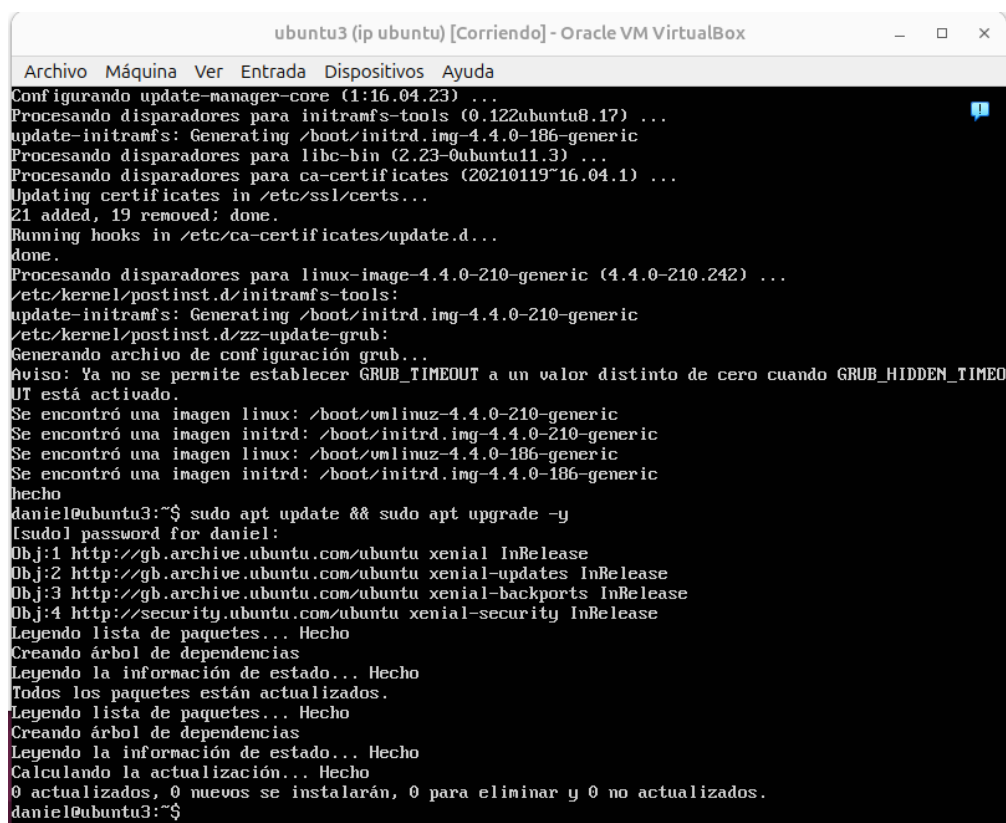
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

daniel@ubuntu3:~$ _
```

Nuestro Ubuntu server arranca bien, nos ha pedido usuario y contraseña y está funcionando correctamente.

Ahora ejecutaremos `sudo apt update` → Actualiza la lista de paquetes disponibles desde los repositorios.

`sudo apt upgrade -y` → Instala todas las actualizaciones disponibles sin pedir confirmación.



```
ubuntu3 (ip ubuntu) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Configurando update-manager-core (1:16.04.23) ...
Procesando disparadores para initramfs-tools (0.122ubuntu8.17) ...
update-initramfs: Generating /boot/initrd.img-4.4.0-186-generic
Procesando disparadores para libc-bin (2.23-0ubuntu11.3) ...
Procesando disparadores para ca-certificates (20210119~16.04.1) ...
Updating certificates in /etc/ssl/certs...
21 added, 19 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Procesando disparadores para linux-image-4.4.0-210-generic (4.4.0-210.242) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-4.4.0-210-generic
/etc/kernel/postinst.d/zz-update-grub:
Generando archivo de configuración grub...
Aviso: Ya no se permite establecer GRUB_TIMEOUT a un valor distinto de cero cuando GRUB_HIDDEN_TIMEOUT
UT está activado.
Se encontró una imagen linux: /boot/vmlinuz-4.4.0-210-generic
Se encontró una imagen initrd: /boot/initrd.img-4.4.0-210-generic
Se encontró una imagen linux: /boot/vmlinuz-4.4.0-186-generic
Se encontró una imagen initrd: /boot/initrd.img-4.4.0-186-generic
hecho
daniel@ubuntu3:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for daniel:
Obj:1 http://gb.archive.ubuntu.com/ubuntu xenial InRelease
Obj:2 http://gb.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:3 http://gb.archive.ubuntu.com/ubuntu xenial-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
daniel@ubuntu3:~$
```

El proceso se ha realizado correctamente.

Fase 2: Configuración de red y acceso remoto.

- **Asignar una IP estática válida en la red local.**

Este punto también lo realizamos en el reto del día 4 adjunto enlace <https://github.com/daniariza64/Pr-cticas>

- **Configurar el archivo `/etc/hosts` correctamente con el nombre del servidor.**

Primero miramos como se llama nuestro servidor con el comando `hostname`.

```
127.0.0.1    localhost
127.0.1.1    ubuntu3.myguest.virtualbox.org  ubuntu3

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

Aquí vemos su nombre que es ubuntu3.

Pasamos a editar `/etc/hosts`.

```
daniel@ubuntu3:~$ getent hosts ubuntu
127.0.1.1    ubuntu server practicas codearts
daniel@ubuntu3:~$ _
```

Hemos cambiado la línea de 127.0.1.1 cambiando el nombre inicial por el de ubuntu server prácticas codearts y con esto queda editado el archivo.

● Instalar y habilitar el servicio SSH.

Primero instalamos el servicio SSH con los comandos

```
sudo apt update
```

```
sudo apt install openssh-server -y
```

```
daniel@ubuntu3:~$ sudo apt install openssh-server -y
sudo: imposible resolver el anfitrión ubuntu3
[sudo] password for daniel:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-server ya está en su versión más reciente (1:7.2p2-4ubuntu2.10).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
daniel@ubuntu3:~$
```

Ahora verificamos que el servicio esté activo con `sudo systemctl status ssh`

```
Archivo Maquina ver Entrada Dispositivos Ayuda
daniel@ubuntu3:~$ sudo systemctl status ssh
sudo: imposible resolver el anfitrión ubuntu3
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since mié 2025-06-18 15:05:56 CEST; 28min ago
 Main PID: 1080 (sshd)
   CGroup: /system.slice/ssh.service
           └─1080 /usr/sbin/sshd -D

jun 18 15:05:56 ubuntu3 systemd[1]: Starting OpenBSD Secure Shell server...
jun 18 15:05:56 ubuntu3 sshd[1080]: Server listening on 0.0.0.0 port 22.
jun 18 15:05:56 ubuntu3 sshd[1080]: Server listening on :: port 22.
jun 18 15:05:56 ubuntu3 systemd[1]: Started OpenBSD Secure Shell server.
daniel@ubuntu3:~$
```

Vemos que efectivamente está activo.

● Verificar la conexión remota desde otro sistema con ssh.

Con este comando `ssh daniel@192.168.1.40` que se compone de el nombre de usuario y la ip de Ubuntu server verificaremos si existe conexión ssh desde mi pc real (que es desde donde ejecutamos el comando) hacia el Ubuntu server virtualizado.

```
daniel-ariza@Personal:~$ ssh daniel@192.168.1.40
The authenticity of host '192.168.1.40 (192.168.1.40)' can't be established.
ED25519 key fingerprint is SHA256:W37pkg534UFBY3oPQR1xd00LG5lxOBtPEoNFnxvL5yA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.40' (ED25519) to the list of known hosts.
daniel@192.168.1.40's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
Last login: Wed Jun 18 13:24:27 2025
daniel@ubuntu3:~$
```

Podemos ver que ha ido todo correctamente.

Fase 3: Seguridad mínima obligatoria

- **Instalar y configurar UFW para que:**
 - Solo permita tráfico por puerto 22 (SSH) y puerto 80 (HTTP)

```
daniel@ubuntu3:~$ sudo apt update
sudo: imposible resolver el anfitrión ubuntu3
[sudo] password for daniel:
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [106 kB]
Obj:2 http://gb.archive.ubuntu.com/ubuntu xenial InRelease
Des:3 http://gb.archive.ubuntu.com/ubuntu xenial-updates InRelease [106 kB]
Des:4 http://gb.archive.ubuntu.com/ubuntu xenial-backports InRelease [106 kB]
Descargados 317 kB en 0s (320 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
daniel@ubuntu3:~$ sudo apt install ufw
sudo: imposible resolver el anfitrión ubuntu3
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.35-0ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
daniel@ubuntu3:~$
```

Con los comandos `sudo apt update` `sudo apt install ufw` comprobamos que ya lo tenemos instalado, en caso de no tenerlo se realizaría la instalación.

Con los comandos:

```
sudo ufw allow 22 # Permitir SSH
```

```
sudo ufw allow 80 # Permitir HTTP
```

Permitiremos la conexión en los puertos necesarios antes de activar el firewall, esto es importante para no perder la conexión SSH.

```
daniel@ubuntu3:~$ sudo ufw allow 22
sudo: imposible resolver el anfitrión ubuntu3
Reglas actualizadas
Reglas actualizadas (v6)
daniel@ubuntu3:~$ sudo ufw allow 80
sudo: imposible resolver el anfitrión ubuntu3
Reglas actualizadas
Reglas actualizadas (v6)
daniel@ubuntu3:~$
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

Con estos comandos negaremos cualquier conexión entrante no autorizada previamente.

Permitimos cualquier conexión saliente.

```
daniel@ubuntu3:~$ sudo ufw default deny incoming
sudo: imposible resolver el anfitrión ubuntu3
La política incoming predeterminada cambiÃ³ a «deny»
(asegÃºrese de actualizar sus reglas consecuentemente)
daniel@ubuntu3:~$ sudo ufw default allow outgoing
sudo: imposible resolver el anfitrión ubuntu3
La política outgoing predeterminada cambiÃ³ a «allow»
(asegÃºrese de actualizar sus reglas consecuentemente)
daniel@ubuntu3:~$
```

Ahora activaremos UFW (Uncomplicated Firewall) con

```
sudo ufw enable
```

```
daniel@ubuntu3:~$ sudo ufw enable
sudo: imposible resolver el anfitrión ubuntu3
El cortafuegos estÃ¡ activo y habilitado en el arranque del sistema
daniel@ubuntu3:~$
```

Con `sudo ufw status verbose` verificaremos las reglas activas.

```
daniel@ubuntu3:~$ sudo ufw status verbose
sudo: imposible resolver el anfitrión ubuntu3
Estado: activo
Acceso: on (low)
Predeterminado: deny (entrantes), allow (salientes), disabled (enrutados)
Perfiles nuevos: skip

Hasta                Acción          Desde
-----
22                   ALLOW IN       Anywhere
80                   ALLOW IN       Anywhere
22 (v6)              ALLOW IN       Anywhere (v6)
80 (v6)              ALLOW IN       Anywhere (v6)

daniel@ubuntu3:~$
```

Como podemos ver está todo correcto.

- **Crear un nuevo usuario llamado desarrollador, con acceso limitado y sin permisos de superusuario.**

`sudo adduser desarrollador` con este comando crearemos el usuario.

```
daniel@ubuntucodearts:~$ sudo adduser desarrollador
sudo: imposible resolver el anfitrión ubuntucodearts
[sudo] password for daniel:
Añadiendo el usuario 'desarrollador' ...
Añadiendo el nuevo grupo 'desarrollador' (1001) ...
Añadiendo el nuevo usuario 'desarrollador' (1001) con grupo 'desarrollador' ..
Creando el directorio personal '/home/desarrollador' ...
Copiando los ficheros desde '/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para desarrollador
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []:
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
daniel@ubuntucodearts:~$
```

Nos pedirá una nueva contraseña y ciertos datos.
Ahora comprobaremos que no está en el grupo `sudo`.


```
¿Es correcta la información? [S/n] s
daniel@ubuntucodearts:~$ groups desarrollador
desarrollador : desarrollador
daniel@ubuntucodearts:~$ _
```

`groups desarrollador` Este comando servirá para ello, sabemos que no está en el grupo `sudo` por que no aparece dicha palabra.

- **Cambiar el puerto por defecto de SSH a 2222 y reforzar la configuración (/etc/ssh/sshd_config).** Primero editaremos el archivo de configuración SSH con el comando `sudo nano /etc/ssh/sshd_config`

```
GNU nano 2.5.3 Archivo: /etc/ssh/sshd_config

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 2222
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no_
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
```

`Port 2222`

`# Cambia de 22 a 2222`

`PermitRootLogin no` # Desactiva login como root
`PasswordAuthentication yes` # (O 'no' si solo quieres clave SSH)

`AllowUsers desarrollador` # (Opcional) Solo este usuario puede usar SSH

Ahora permitiremos el nuevo puerto en el firewall con:

`sudo ufw allow 2222`

`sudo ufw delete allow 22.`

```
daniel@ubuntucodearts:~$ sudo ufw allow 2222
sudo: imposible resolver el anfitrión ubuntucodearts
[sudo] password for daniel:
Regla añadida
Regla añadida (v6)
daniel@ubuntucodearts:~$ sudo ufw delete allow 22
sudo: imposible resolver el anfitrión ubuntucodearts
Regla eliminada
Regla eliminada (v6)
daniel@ubuntucodearts:~$
```

```
daniel@ubuntucodearts:~$ sudo ss -tulpn | grep ssh
sudo: imposible resolver el anfitrión ubuntucodearts
tcp    LISTEN    0      128      *:2222      *:*        users:(("sshd",pid=
924,fd=3))
tcp    LISTEN    0      128      :::2222     :::*       users:(("sshd",pid=
924,fd=4))
daniel@ubuntucodearts:~$ _
```

`sudo ss -tulpn | grep ssh` con este comando vemos que el puerto de escucha se ha cambiado correctamente.

```

daniel-ariza@Personal:~$ ssh desarrollador@192.168.1.40 -p 2222
desarrollador@192.168.1.40's password:
Permission denied, please try again.
desarrollador@192.168.1.40's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

desarrollador@ubuntucodearts:~$

```

Y también podemos comprobar que nos podemos conectar desde otro equipo sin problemas.

● Desactivar el acceso SSH del usuario root.

Primero debemos editar la configuración SSH, para ello ejecutaremos el comando:

`sudo nano /etc/ssh/sshd_config`

```

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes

[ 88 líneas leídas ]
^G Ver ayuda  ^O Guardar  ^U Buscar  ^K Cortar Text  ^J Justificar  ^C Posición  ^Y Pág. ant.
^X Salir      ^R Leer fich  ^_ Reemplazar  ^M Pegar txt  ^T Ortografía  ^L Ir a línea  ^U Pág. sig

```

Buscamos esta línea `PermitRootLogin no` si no está la agregamos nosotros.

```
daniel-ariza@Personal:~$ ssh root@192.168.1.40 -p 2222
root@192.168.1.40's password:
Permission denied, please try again.
root@192.168.1.40's password: █
```

Al intentar establecer conexión desde otro terminal como usuario root la petición es denegada lo que quiere decir que el proceso se realizó correctamente.

Fase 4: Estructura de carpetas y servicios iniciales.

- **Crear una estructura de carpetas en /srv/ con los siguientes directorios:**
 - /srv/www → para proyectos web
 - /srv/repositorios → para guardar código fuente
 - /srv/docs → para documentación técnica interna

Vamos a crear los directorios con el siguiente comando:

`sudo mkdir -p /srv/www /srv/repositorios /srv/docs`

```
daniel@ubuntucodearts:~$ sudo mkdir -p /srv/www /srv/repositorios /srv/docs
sudo: imposible resolver el anfitrión ubuntucodearts
daniel@ubuntucodearts:~$ ls -l /sv
ls: no se puede acceder a '/sv': No existe el archivo o el directorio
daniel@ubuntucodearts:~$ ls -l /srv
total 12
drwxr-xr-x 2 root root 4096 jun 19 09:58 docs
drwxr-xr-x 2 root root 4096 jun 19 09:59 repositorios
drwxr-xr-x 3 root root 4096 jun 19 09:58 www
daniel@ubuntucodearts:~$
```

Vemos que se crearon correctamente.

- **Establecer permisos específicos:**

- El usuario desarrollador puede escribir solo en /srv/www
- Solo el usuario administrador puede acceder a /srv/repositorios.

En primer lugar crearemos el usuario administrador con:

sudo adduser administrador

```
daniel@ubuntucodearts:~$ sudo adduser administrador
sudo: imposible resolver el anfitrión ubuntucodearts
[sudo] password for daniel:
Añadiendo el usuario 'administrador' ...
Añadiendo el nuevo grupo 'administrador' (1002) ...
Añadiendo el nuevo usuario 'administrador' (1002) con grupo 'administrador' ...
Creando el directorio personal '/home/administrador' ...
Copiando los ficheros desde '/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para administrador
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []:
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
daniel@ubuntucodearts:~$
```

En segundo lugar asignaremos propietarios y permisos.

/srv/www → propietario: desarrollador.

sudo chown desarrollador:desarrollador /srv/www

sudo chmod 755 /srv/www

con estos comandos.

Esto permite:

- **Desarrollador:** lectura, escritura, ejecución.
- **Otros usuarios:** solo lectura y ejecución.

/srv/repositorios → propietario: administrador,
nadie más puede acceder.

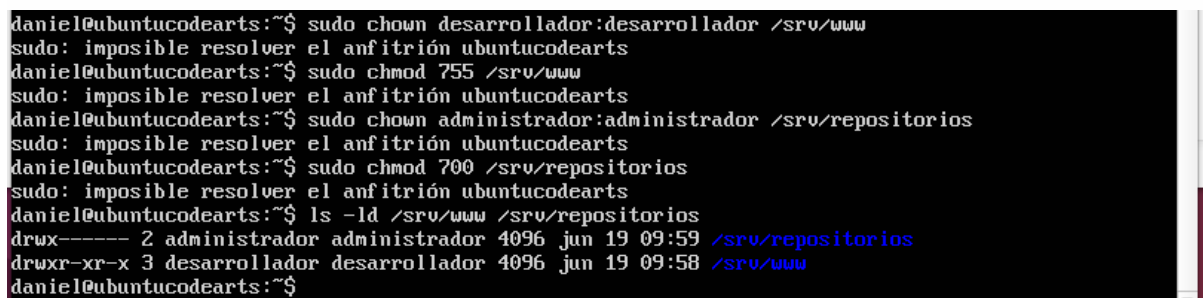
`sudo chown administrador:administrador /srv/repositorios`

`sudo chmod 700 /srv/repositorios`

Esto permite:

- Solo **administrador**: acceso completo.
- Nadie más (ni siquiera **root** en modo usuario normal).

`ls -ld /srv/www /srv/repositorios` y con este comando verificamos que se ha realizado correctamente.



```
daniel@ubuntucodearts:~$ sudo chown desarrollador:desarrollador /srv/www
sudo: imposible resolver el anfitrión ubuntucodearts
daniel@ubuntucodearts:~$ sudo chmod 755 /srv/www
sudo: imposible resolver el anfitrión ubuntucodearts
daniel@ubuntucodearts:~$ sudo chown administrador:administrador /srv/repositorios
sudo: imposible resolver el anfitrión ubuntucodearts
daniel@ubuntucodearts:~$ sudo chmod 700 /srv/repositorios
sudo: imposible resolver el anfitrión ubuntucodearts
daniel@ubuntucodearts:~$ ls -ld /srv/www /srv/repositorios
drwx----- 2 administrador administrador 4096 jun 19 09:59 /srv/repositorios
drwxr-xr-x 3 desarrollador desarrollador 4096 jun 19 09:58 /srv/www
daniel@ubuntucodearts:~$
```

Confirmamos que se ha realizado de forma correcta.

- Instalar el servidor web Apache2 o NGINX (a elegir)
y colocar una página de prueba en /srv/www.

Procedemos a instalar Apache2 con:

`sudo apt update`

`sudo apt install apache2`

```

daniel@ubuntucodearts:~$ sudo apt update
sudo: imposible resolver el anfitrión ubuntucodearts
[sudo] password for daniel:
Obj:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Obj:2 http://gb.archive.ubuntu.com/ubuntu xenial InRelease
Obj:3 http://gb.archive.ubuntu.com/ubuntu xenial-updates InRelease
Obj:4 http://gb.archive.ubuntu.com/ubuntu xenial-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
daniel@ubuntucodearts:~$ sudo apt install apache2
sudo: imposible resolver el anfitrión ubuntucodearts
sudo: aptinstall: orden no encontrada
daniel@ubuntucodearts:~$ sudo apt install apache2
sudo: imposible resolver el anfitrión ubuntucodearts
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 libperl5.22 perl perl-modules-5.22 rename ssl-cert
Paquetes sugeridos:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 libperl5.22 perl perl-modules-5.22 rename ssl-cert
0 actualizados, 14 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 7.802 kB de archivos.
Se utilizarán 45,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

Ahora cambiaremos la raíz del sitio web a `/srv/www`.
 Apache, por defecto, usa `/var/www/html`. Vamos a cambiar eso por `/srv/www`.

Editaremos el archivo del sitio con `sudo nano`
`/etc/apache2/sites-available/000-default.conf`.

```

ServerAdmin webmaster@localhost
DocumentRoot /srv/www_

```

Y cambiemos la línea antes mencionada.

Crearemos una página de prueba.

echo "<h1>Servidor Apache funcionando correctamente</h1>" | sudo tee /srv/www/index.html

```
daniel@ubuntucodearts:~$ echo "<h1>Servidor Apache funcionando correctamente</h1>" | sudo tee /srv/www/index.html
sudo: imposible resolver el anfitrión ubuntucodearts
[sudo] password for daniel:
<h1>Servidor Apache funcionando correctamente</h1>
daniel@ubuntucodearts:~$
```

Reiniciamos el servidor apache con `sudo systemctl restart apache2`

Y ahora desde el navegador de internet del equipo real introducimos la ip de nuestro Ubuntu server y si nos da el resultado que hay en la siguiente captura nuestro servidor Apache estará instalado y funcionando perfectamente.



Servidor Apache funcionando correctamente