

Comandos.

Actividad 1. Tarea individual.

15/01/2021

Raúl Montero Peregrina

ÍNDICE

1.- Enunciado.....	3
2.- Introducción	4
2.1.- Instalación servidor web APACHE	4
3.- ¿Cómo sabemos si tenemos conexión a internet? Pista: ifconfig, ping.....	5
4.- ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: ufw, netstat.	5
5.- ¿Cómo sabemos a quién pertenece una dirección web (URL)? Pista: dig, nslookup.	7
6.- ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget.....	7
7.- ¿Qué otros comandos te han hecho falta?	8
8.- Guía HOW TO en formato Markdown	11
8.1.- Subida de la guía a GitHub en formato texto o MarkDown.....	11

Actividad 1. Tarea Individual. Comandos.



1.- Enunciado

Requerimiento 1

La administración de un servidor web y/o un servidor de aplicaciones requiere unos conocimientos básicos de comandos de consola que permite visualizar qué está pasando en nuestro servidor. Se pide practicar y crear una guía de uso para las siguientes problemáticas que nos podemos encontrar:

1. ¿Cómo sabemos si tenemos conexión a internet? Pista: ifconfig, ping
2. ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: ufw, netstat
3. ¿Cómo sabemos a quién pertenece una dirección web (URL)? Pista: dig, nslookup
4. ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget
5. ¿Qué otros comandos te han hecho falta?

Valoración: 10 puntos sobre 10

Consideraciones

Para toda la actividad se valorará el orden y la claridad de la documentación, así como la facilidad de uso.

Para la entrega, es necesaria la creación y subida a la plataforma de un pequeño documento formal sobre la actividad (portada, explicación, etc.) y una guía "how-to" que describa y permita resolver las preguntas planteadas en la actividad*.

Nótese que más adelante se pedirá que se realicen tareas con un repositorio GIT que contenga la documentación de esta actividad.

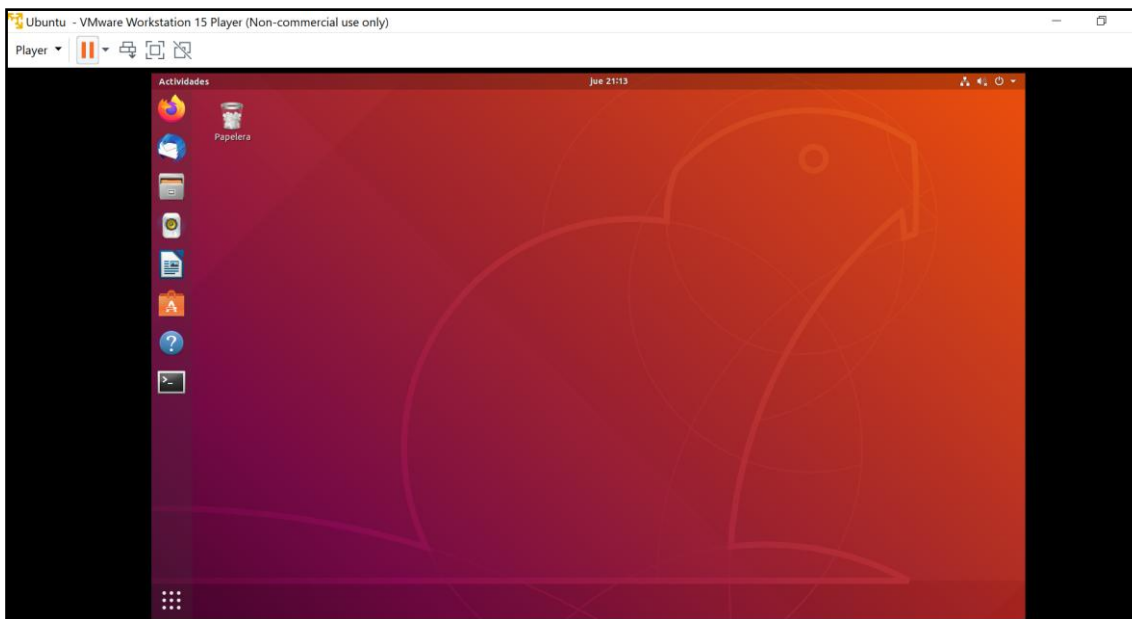
* Se recomienda crear un repositorio GitHub para almacenar la guía "how-to" en formato texto y/o Markdown.

Se considerará a la hora de evaluar la facilidad de uso de dicha guía que se explique de manera clara y sencilla:

- 1.-Qué hace el comando
- 2.-Cómo se usa
- 3.-Por qué responde a la pregunta
- 4.-Cómo se interpreta la salida

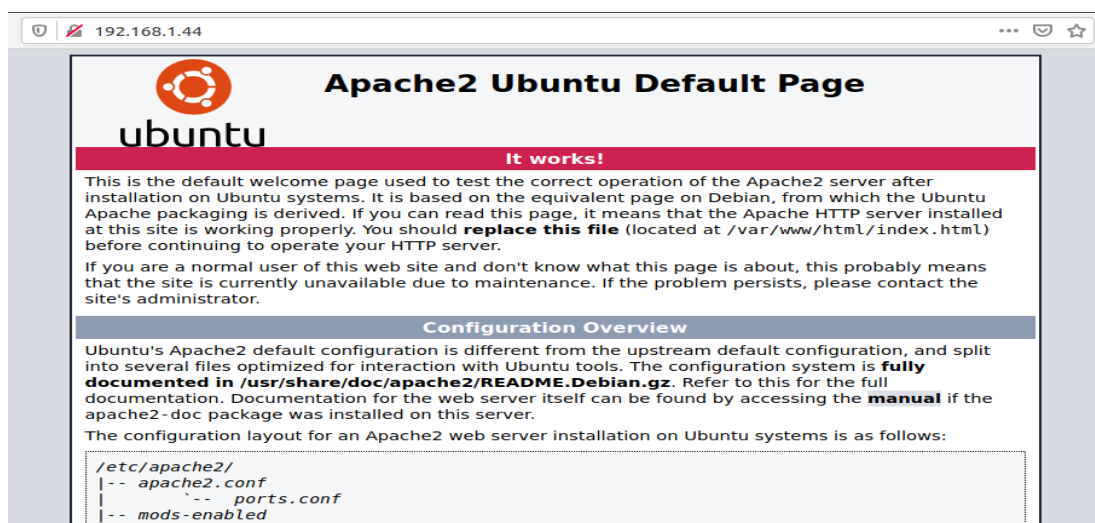
2.- Introducción

Para la realización de la práctica y dado que nuestro Sistema Operativo es Windows 10, usaremos el software VMware para crear una máquina virtual de Linux con distribución Ubuntu para así poder ejecutar y probar los comandos propuestos en su terminal:



2.1.- Instalación servidor web APACHE

Para poder realizar la práctica y poder manejar los comandos sobre un servidor, aunque este no será remoto pero lo simulará, vamos a instalar sobre nuestro SO un servidor web como Apache. Para ello usaremos simplemente el comando de instalación `sudo apt install apache2`. Esto instalará Apache en nuestro linux. Con los comandos que siguen a continuación daremos acceso a Apache en nuestro cortafuegos (ufw). Si todo ha ido bien y tecleamos nuestra ip en el navegador (ifconfig) deberíamos ver la pantalla de Apache:



Podemos observar que Apache se ha instalado correctamente en nuestro sistema operativo Linux-Ubuntu.

3.- ¿Cómo sabemos si tenemos conexión a internet? Pista: ifconfig, ping.

Con el comando **ifconfig** podemos averiguar nuestra ip y haciendo un **ping** a cualquier página de internet veremos si hay conexión a internet:

```
Archivo Editar Ver Buscar Terminal Ayuda
raul@raul-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.44 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8409:462:c865:9f1c prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:0c:a1:de txqueuelen 1000 (Ethernet)
    RX packets 195931 bytes 275093197 (275.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12029 bytes 1140283 (1.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Usando el comando **ping -c 4 hostinger.es** estamos diciendo que enviaremos 4 paquetes a la dirección de internet **www.hostinger.es**:

```
raul@raul-virtual-machine:~$ ping -c 4 hostinger.es
PING hostinger.es (104.18.21.174) 56(84) bytes of data.
64 bytes from 104.18.21.174 (104.18.21.174): icmp_seq=1 ttl=57 time=5.60 ms
64 bytes from 104.18.21.174 (104.18.21.174): icmp_seq=2 ttl=57 time=7.21 ms
64 bytes from 104.18.21.174 (104.18.21.174): icmp_seq=3 ttl=57 time=6.66 ms
64 bytes from 104.18.21.174 (104.18.21.174): icmp_seq=4 ttl=57 time=31.4 ms

--- hostinger.es ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 5.602/12.736/31.462/10.827 ms
raul@raul-virtual-machine:~$
```

La salida es que se han enviado 4 paquetes y recibido esos 4 paquetes de la web a la que se los hemos enviado sin pérdida de datos por lo que tenemos conexión a internet correcta desde nuestra Máquina Virtual de Linux.

4.- ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: ufw, netstat.

Con el comando **ufw** podemos interactuar con nuestro firewall, crear reglas y ver su estado. Con **ufw status** vemos el estado del firewall:

```
root@raul-virtual-machine:/home/raul# ufw status
Estado: activo

Hasta                Acción              Desde
-----
Apache               ALLOW               Anywhere
Apache (v6)          ALLOW               Anywhere (v6)
```

Como se puede observar en esta salida el firewall está activo y se permite el acceso a nuestro servidor web Apache.

Con **ufw** también será posible abrir un puerto específico con el objetivo de permitir las conexiones a un determinado servicio a través de él, por ejemplo, si se necesita configurar un servidor web para escuchar en los puertos 80 (HTTP) y 443 (HTTPS) de forma predeterminada, ejecutaremos lo siguiente según sea el caso:

sudo ufw allow 80/tcp (Por número de puerto)

sudo ufw allow 443/tcp (Por número de puerto)

```
root@raul-virtual-machine:/home/raul# sudo ufw allow 80/tcp
Regla añadida
Regla añadida (v6)
root@raul-virtual-machine:/home/raul# sudo ufw allow 443/tcp
sudo: uf: orden no encontrada
root@raul-virtual-machine:/home/raul# sudo ufw allow 443/tcp
Regla añadida
Regla añadida (v6)
root@raul-virtual-machine:/home/raul#
```

Como se ve en esta salida se han añadido 2 reglas y permitido (allow) el acceso al puerto 80(protocolo HTTP) y puerto 443(protocolo HTTPS) usados por nuestro servidor web Apache.

El comando **netstat** nos ofrece información acerca de nuestra red. Con el comando **netstat -ap | grep http** estamos preguntando a la red si el puerto asociado a ese protocolo está listo para escuchar o recibir peticiones y cuál es el servicio que lo usa:

```
root@raul-virtual-machine:/home/raul# netstat -ap | grep http
tcp6      0      0 [::]:http        [::]:*            ESCUCHAR      8101/apache2
root@raul-virtual-machine:/home/raul#
```

Se observa en la salida que el puerto 80(el usado por nuestro servidor web para envío de respuestas y recepción de consultas) está escuchando y listo y es usado por el servidor web apache2.

Otra opción es usar el comando **netstat -ano**, nos dará protocolos ips y puertos que están escuchando y listos para recibir respuestas:

```
root@raul-virtual-machine:/home/raul# netstat -ano
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Envíad Dirección local      Dirección remota      Estado      Temporizador
tcp      0      0 0.0.0.0:139        0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
tcp      0      0 127.0.0.53:53      0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
tcp      0      0 0.0.0.0:22         0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
tcp      0      0 127.0.0.1:631      0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
tcp      0      0 0.0.0.0:445        0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
tcp6     0      0 :::139             :::*                   ESCUCHAR    apagado (0.00/0/0)
tcp6     0      0 :::80              :::*                   ESCUCHAR    apagado (0.00/0/0)
tcp6     0      0 :::22              :::*                   ESCUCHAR    apagado (0.00/0/0)
tcp6     0      0 :::1:631           :::*                   ESCUCHAR    apagado (0.00/0/0)
tcp6     0      0 :::445             :::*                   ESCUCHAR    apagado (0.00/0/0)
udp      0      0 0.0.0.0:5353       0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
udp      0      0 127.0.0.53:53      0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
udp      0      0 0.0.0.0:68         0.0.0.0:*             ESCUCHAR    apagado (0.00/0/0)
```

De esta salida interpretamos que el puerto 80 utilizado por nuestro servidor está listo para interactuar.

5.- ¿Cómo sabemos a quién pertenece una dirección web (URL)?

Pista: dig, nslookup.

Tanto el comando **dig** como **nslookup** nos permiten averiguar la ip asociada a un dominio o el dominio asociado a una ip. El comando dig es algo más antiguo que nslookup el cual es más utilizado ya que ofrece una salida más limpia y es más intuitivo. Para ver a quien pertenece (ip) una url como hostinger.es usaremos el comando **nslookup hostinger.es** y para dig usaremos **dig hostinger.es +short** para acostar la salida. La salida nos la ip a la que pertenece la url hostinger.es:

```
root@raul-virtual-machine:/home/raul# dig hostinger.com +short
104.19.184.120
104.19.185.120
```

```
root@raul-virtual-machine:/home/raul# nslookup hostinger.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hostinger.com
Address: 104.19.184.120
Name:   hostinger.com
Address: 104.19.185.120
Name:   hostinger.com
Address: 2606:4700::6813:b878
Name:   hostinger.com
Address: 2606:4700::6813:b978
```

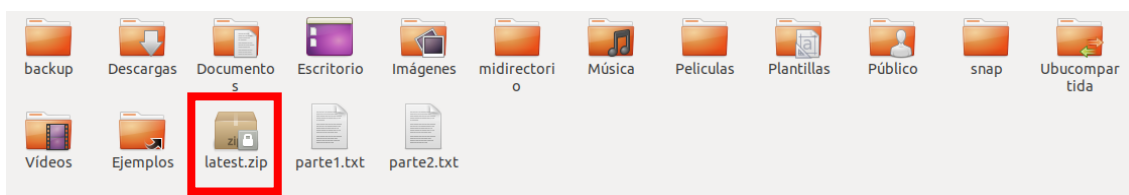
6.- ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget.

El comando **wget** nos permite acceder a un servidor y realizar una descarga de alguno de sus contenidos. Usando el comando **wget https://wordpress.org/latest.zip** accederemos al servidor y descargaremos en nuestro directorio de trabajo el archivo latest.zip:

```
root@raul-virtual-machine:/home/raul# wget https://wordpress.org/latest.zip
--2021-01-22 21:41:51-- https://wordpress.org/latest.zip
Resolviendo wordpress.org (wordpress.org)... 198.143.164.252
Conectando con wordpress.org (wordpress.org)[198.143.164.252]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 16528923 (16M) [application/zip]
Guardando como: "latest.zip"

latest.zip          100%[=====] 15,76M  1,29MB/s  en 12s
2021-01-22 21:42:04 (1,29 MB/s) - "latest.zip" guardado [16528923/16528923]
```

Se observa en la salida como se accede al servidor y el proceso de descarga del recurso. A continuación veremos en nuestro directorio de trabajo como efectivamente se ha descargado dicho recurso del servidor y concluimos que se ha accedido al servidor solicitado correctamente.



Con el comando **curl** también podemos comprobar la conectividad a una URL. El uso más simple de curl es mostrar el contenido de una página. El siguiente ejemplo mostrará la página de inicio de **testdomain.com**.

```
root@raul-virtual-machine:/home/raul# curl hostinger.com

No se ha encontrado la orden «curl», pero se puede instalar con:

apt install curl

root@raul-virtual-machine:/home/raul# sudo apt install curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  efibootmgr libfwup1
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libcurl4
Se instalarán los siguientes paquetes NUEVOS:
  curl libcurl4
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 96 no actualizados.
Se necesita descargar 373 kB de archivos.
Se utilizarán 1.038 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4 amd64 7.58.0-2ubuntu3.12 [214 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.12 [159 kB]
Descargados 373 kB en 0s (1.088 kB/s)
Seleccionando el paquete libcurl4:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 174181 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libcurl4_7.58.0-2ubuntu3.12_amd64.deb ...
Desempaquetando libcurl4:amd64 (7.58.0-2ubuntu3.12) ...
Seleccionando el paquete curl previamente no seleccionado.
Preparando para desempaquetar .../curl_7.58.0-2ubuntu3.12_amd64.deb ...
Desempaquetando curl (7.58.0-2ubuntu3.12) ...
Configurando libcurl4:amd64 (7.58.0-2ubuntu3.12) ...
Configurando curl (7.58.0-2ubuntu3.12) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1.2) ...
root@raul-virtual-machine:/home/raul#

root@raul-virtual-machine:/home/raul# curl testdomain.com
<html><head><title>testdomain.com</title></head><body><h1>testdomain.com</h1><p>Coming soon.</p></body></html>root@raul-virtual-machine:/home/raul#
```

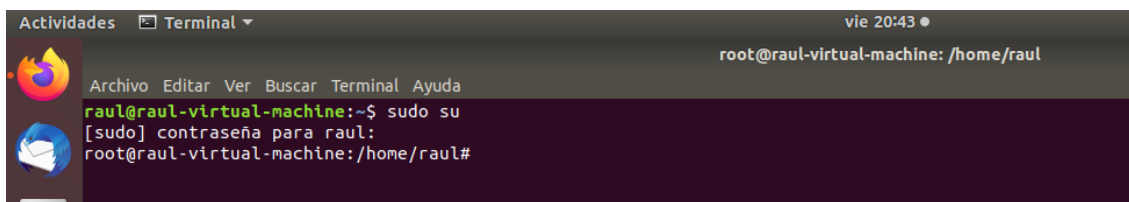
Esto generará el código fuente completo de la página de inicio del dominio. Si no se especifica ningún protocolo, curl lo interpretará a HTTP.

7.- ¿Qué otros comandos te han hecho falta?

- **sudo apt –get install gufw**: con permisos de superusuario seguido de apt –get install seguido de la app que queremos instalar para instalar apps. En este caso no teníamos instalado ningún firewall así que hemos instalado gufw:

```
raul@raul-virtual-machine:~$ sudo apt-get install gufw
[sudo] contraseña para raul:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  efibootmgr libfwup1
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  gufw
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 96 no actualizados.
Se necesita descargar 855 kB de archivos.
Se utilizarán 3.517 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 gufw all 18.04.0-0ubuntu1 [855 kB]
Descargados 855 kB en 1s (1.053 kB/s)
Seleccionando el paquete gufw previamente no seleccionado.
(Leyendo la base de datos ... 173755 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../gufw_18.04.0-0ubuntu1_all.deb ...
Desempaquetando gufw (18.04.0-0ubuntu1) ...
Configurando gufw (18.04.0-0ubuntu1) ...
Procesando disparadores para desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para gnome-menus (3.13.3-11ubuntu1.1) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para mime-support (3.60ubuntu1) ...
```


- **sudo su**: para ejecutar comandos como superusuario con plenos permisos. Se pide contraseña:



```

raul@raul-virtual-machine:~$ sudo su
[sudo] contraseña para raul:
root@raul-virtual-machine:/home/raul#

```

- **clear**: para limpiar la consola.
- **nc o Netcat**: netcat o nc nos permite escanear un puerto y ver si la conexión a él es satisfactoria

Netcat

Nc o *Netcat* se utiliza para una gran variedad de objetivos relacionados con redes y sus diferentes protocolos. Puede hacer cosas desde abrir conexiones TCP/UDP, enviar paquetes, escuchar en puertos de nuestra elección, realizar un escaneo de puertos...

Permite escribir la salida a un fichero local, con lo que perfectamente vale para traspasar ficheros y de hecho se utiliza en ciencia forense informática para exfiltrar archivos de sistemas cuando es preciso un análisis en vivo (con el sistema funcionando) sin provocar modificaciones que invaliden la investigación.

Escaneo de un puerto con Netcat

```
nc -zv 127.0.0.1 80
```

```

root@raul-virtual-machine:/home/raul# nc -zv 192.168.1.44 80
Connection to 192.168.1.44 80 port [tcp/http] succeeded!

```

- **sudo apt update**: con este comando actualizamos el índice de los paquetes locales. Esto, para garantizar que en él se reflejen las cargas más recientes de las nuevas versiones de los paquetes.

```

root@raul-virtual-machine:/home/raul# sudo apt update
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [49,0 kB]
Des:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [59,8 kB]
Des:7 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2.464 B]
Des:8 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [295 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [289 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe DEP-11 64x64 Icons [490 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2.468 B]
Des:12 http://es.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [9.288 B]
Descargados 1.449 kB en 1s (1.641 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 96 paquetes. Ejecute «apt list --upgradable» para verlos.

```

- **sudo apt upgrade**: para actualizar los paquetes e instalarlos:

```
root@raul-virtual-machine:/home/raul# sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
efibootmgr libfwup1 liblvm9
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
liblvm10 libnetplan0 linux-headers-5.4.0-64-generic linux-hwe-5.4-headers-5.4.0-64 linux-image-5.4.0-64-generic linux-modules-5.4.0-64-generic
linux-modules-extra-5.4.0-64-generic
Se actualizarán los siguientes paquetes:
apport apport-gtk base-files dmsetup gdb gdbserver gir1.2-mutter-2 grub-common grub-pc grub-pc-bin grub2-common initramfs-tools initramfs-tools-bin
initramfs-tools-core iproute2 kmod language-pack-es language-pack-gnome-es libappindicator3-1 libc-bin libc6 libc6-dbg libcephfs2 libcryptsetup12
libdevmapper1.02.1 libdjvulibre-text libdjvulibre21 libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libdrm2 libegl-mesa0
libevdev2 libgbm1 libgl1-mesa-dri libgl1-mesa-glx libglapi-mesa libglx-mesa0 libgnutls30 libinput-bin libinput10 libkmod2 libmutter-2-0
libnss-myhostname libnss-systemd libpam-modules libpam-modules-bin libpam-runtime libpam-systemd libpam0g libpcap0.8 librados2 libsane-common
libsane1 libsystemd0 libudev1 libwhoopsie0 libxatracker2 linux-firmware linux-generic-hwe-18.04 linux-headers-generic-hwe-18.04
linux-image-generic-hwe-18.04 locales multiarch-support mutter mutter-common netplan.io nplan python3-apport python3-distupgrade
python3-problem-report python3-update-manager sftp-utils shotwell shotwell-common snapd sudo systemd systemd-cryptsetup ubuntu-desktop ubuntu-minimal
```

```
96 actualizados, 7 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 237 MB de archivos.
Se utilizarán 456 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libc6-dbg amd64 2.27-3ubuntu1.4 [5.163 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libc6 amd64 2.27-3ubuntu1.4 [2.832 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 base-files amd64 10.1ubuntu2.10 [60,4 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 locales all 2.27-3ubuntu1.4 [3.611 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libc-bin amd64 2.27-3ubuntu1.4 [643 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpam0g amd64 1.1.8-3.6ubuntu2.18.04.2 [54,9 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpam-modules-bin amd64 1.1.8-3.6ubuntu2.18.04.2 [36,7 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpam-modules amd64 1.1.8-3.6ubuntu2.18.04.2 [242 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnss-systemd amd64 237-3ubuntu10.44 [105 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsystemd0 amd64 237-3ubuntu10.44 [209 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnss-myhostname amd64 237-3ubuntu10.44 [37,6 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpam-systemd amd64 237-3ubuntu10.44 [108 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 snapd amd64 2.48+18.04 [21,0 MB]
Des:14 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 systemd amd64 237-3ubuntu10.44 [2.913 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 udev amd64 237-3ubuntu10.44 [1.103 kB]
```

8.- Guía HOW TO en formato Markdown

Markdown nació como herramienta de conversión de texto plano a HTML. Aunque en realidad Markdown también se considera un lenguaje que tiene la finalidad de permitir crear contenido de una manera sencilla de escribir, y que en todo momento mantenga un diseño legible, así que para simplificar puedes considerar Markdown como un método de escritura.

-----SE ADJUNTA GÚÍA HOW TO EN FORMATO MARKDOWN-----

8.1.- Subida de la guía a GitHub en formato texto o MarkDown

Realizaremos la subida al repositorio creado en GitHub mediante comandos desde la consola. Una vez subido a GitHub, este se encarga de interpretar el fichero.md:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.18363.1316]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

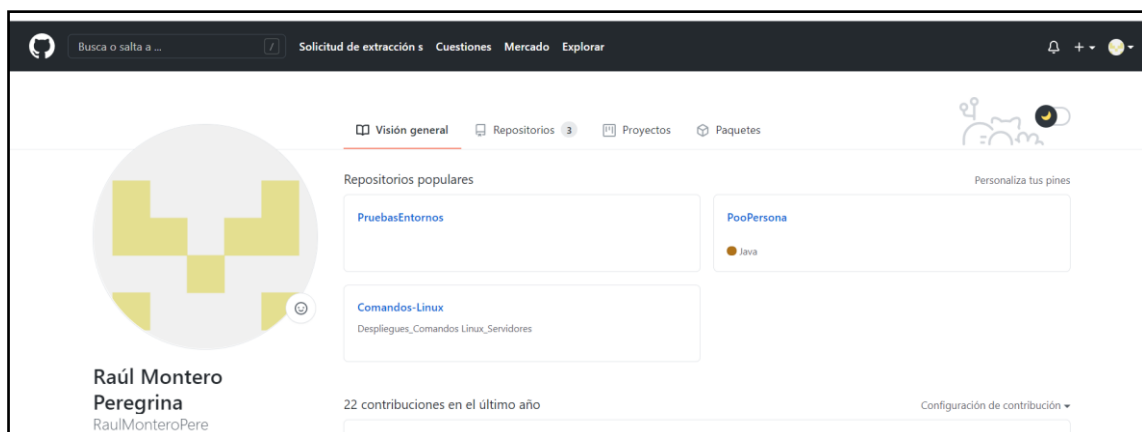
D:\Rauli\FP TELEFONICA DAM\2º CURSO\Despliegue Aplicaciones Web-2T-Oliver\UF3-Instalación y Admin. de Servidores de Transferencia de Archivo\HOW TO-COMANDO
git status
On branch main
Your branch is up to date with 'origin/main'.

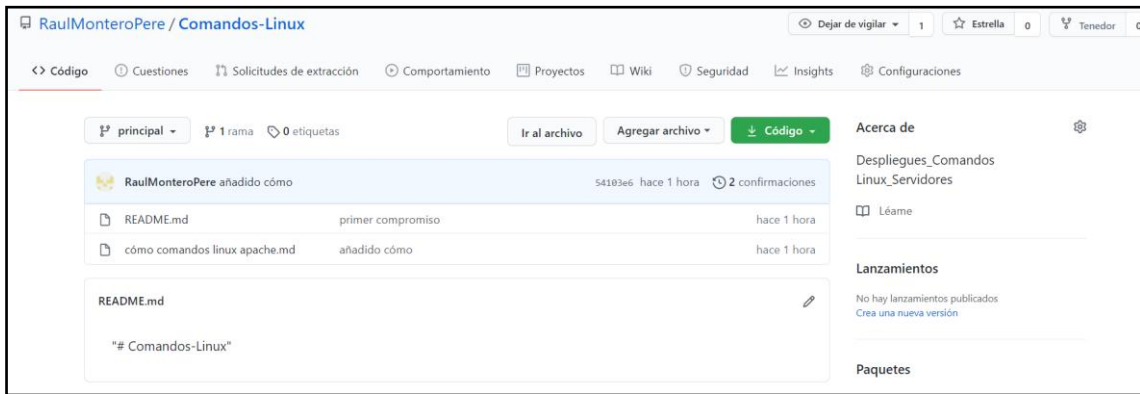
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   how to comandos linux apache.md

D:\Rauli\FP TELEFONICA DAM\2º CURSO\Despliegue Aplicaciones Web-2T-Oliver\UF3-Instalación y Admin. de Servidores de Transferencia de Archivo\HOW TO-COMANDO
git commit -m "añadido how to"
[main 54103e6] añadido how to
1 file changed, 50 insertions(+)
create mode 100644 how to comandos linux apache.md

D:\Rauli\FP TELEFONICA DAM\2º CURSO\Despliegue Aplicaciones Web-2T-Oliver\UF3-Instalación y Admin. de Servidores de Transferencia de Archivo\HOW TO-COMANDO
git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 672 bytes | 672.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RaulMonteroPere/Comandos-Linux.git
   a15b69d..54103e6  main -> main

D:\Rauli\FP TELEFONICA DAM\2º CURSO\Despliegue Aplicaciones Web-2T-Oliver\UF3-Instalación y Admin. de Servidores de Transferencia de Archivo\HOW TO-COMANDO
```





Enlace al fichero en nuestro repositorio de GitHub:

[https://github.com/RaulMonteroPere/Comandos-](https://github.com/RaulMonteroPere/Comandos-Linux/blob/main/how%20to%20comandos%20linux%20apache.md)

[Linux/blob/main/how%20to%20comandos%20linux%20apache.md](https://github.com/RaulMonteroPere/Comandos-Linux/blob/main/how%20to%20comandos%20linux%20apache.md)

