qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| Comandos.  Actividad 1. Tarea individual.  15/01/2021  Raúl Montero Peregrina |

ÍNDICE

[1.- Enunciado 3](#_Toc62475629)

[2.- Introducción 4](#_Toc62475630)

[2.1.- Instalación servidor web APACHE 4](#_Toc62475631)

[3.- ¿Cómo sabemos si tenemos conexión a internet? Pista: ifconfig, ping. 5](#_Toc62475632)

[4.- ¿Cómo sabemos si nuestro servidor es accesible desde Internet? Pista: ufw, netstat. 5](#_Toc62475633)

[5.- ¿Cómo sabemos a quién pertenece una dirección web (URL)? Pista: dig, nslookup. 7](#_Toc62475634)

[6.- ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget. 7](#_Toc62475635)

[7.- ¿Qué otros comandos te han hecho falta? 8](#_Toc62475636)

[8.- Guía HOW TO en formato Markdown 11](#_Toc62475637)

[8.1.- Subida de la guía a GitHub en formato texto o MarkDown 11](#_Toc62475638)

Actividad 1. Tarea Individual. Comandos.



# 1.- Enunciado

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

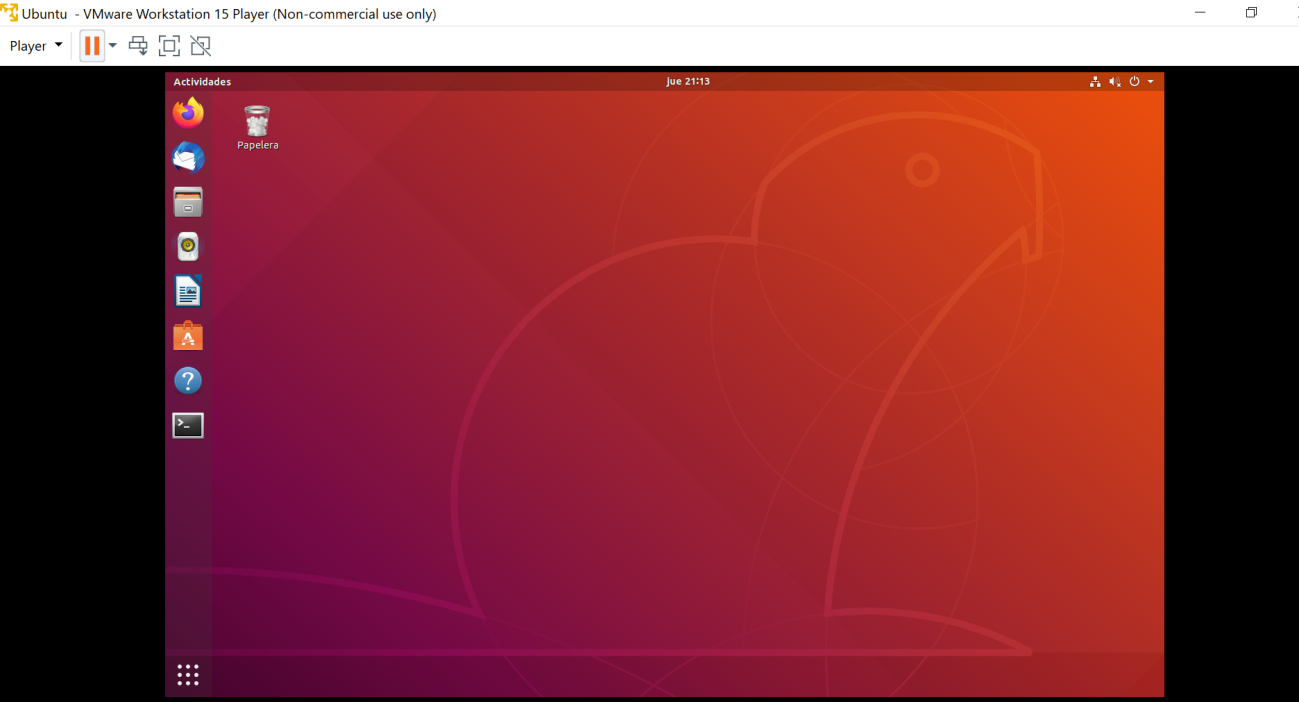
2.-Cómo se usa

3.-Por qué responde a la pregunta

4.-Cómo se interpreta la salida

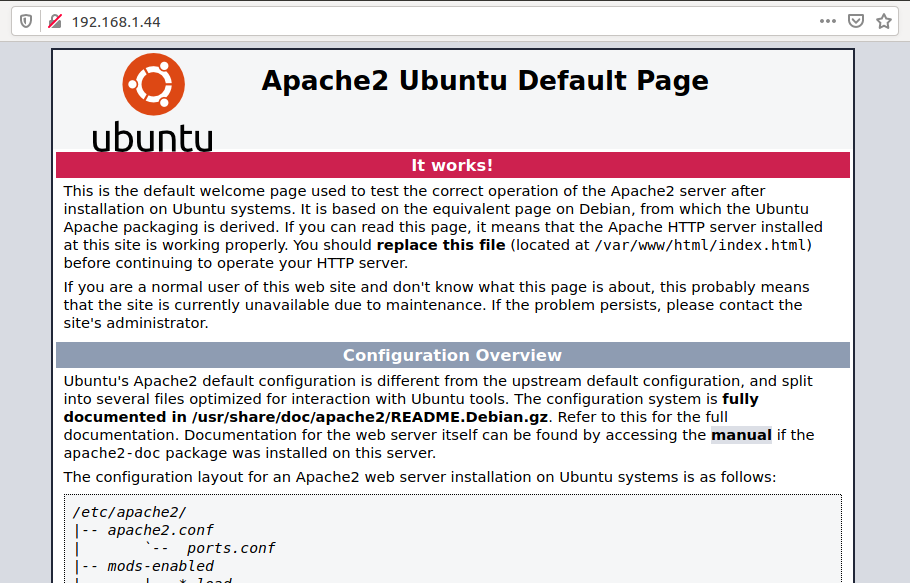
# 2.- Introducción

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

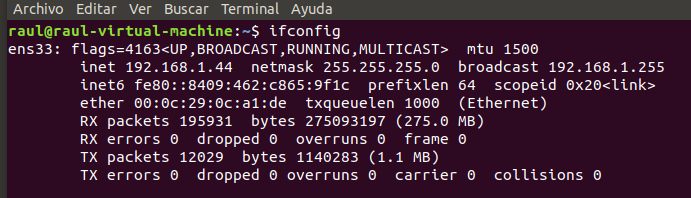
Para poder realizar la práctica y poder manejar los comandos sobre un servidor, aunque este no será remoto pero lo simulará, vamos a instalar sobre nuestro SO un servidor web como Apache. Para ello usaremos simplemente el comando de instalación sudo apt install apache2. Esto instalará Apache en nuestro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



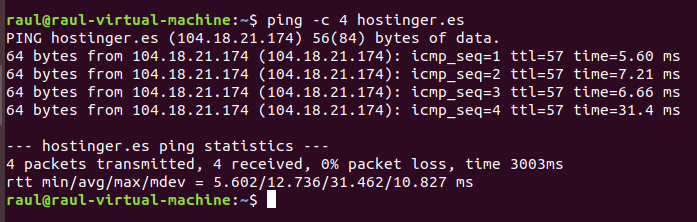
Podemos observar que Apache se ha instalado correctamente en nuestro sistema operativo Linux-Ubuntu.

# 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:



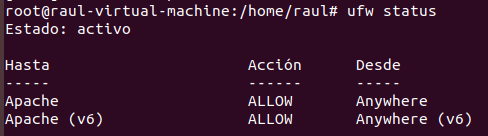
Usando el comando ping –c 4 hostinger.es estamos diciendo que enviaremos 4 paquetes a la dirección de internet www.hostinger.es:



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:

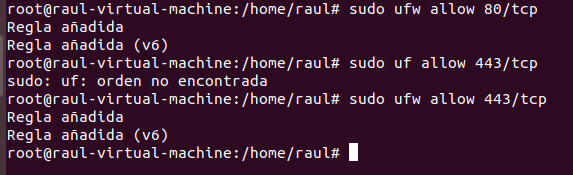


Como se puede observar en esta salida el firewall está activo y se permite el acceso a nuestro servidor web Apache.

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)



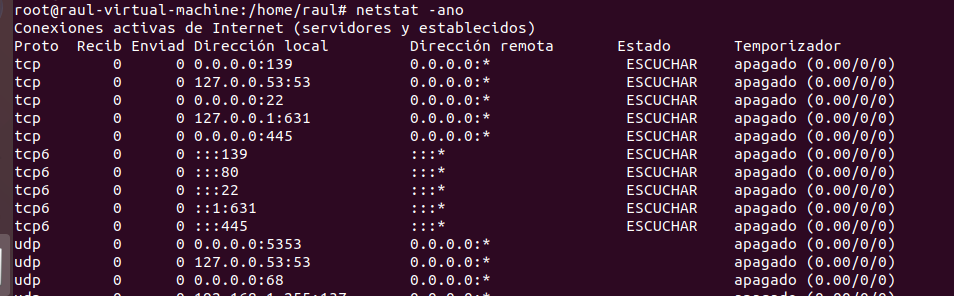
Como se ve en esta salida se han añadido 2 reglas y permitido (allow) el acceso al puerto 80(protocolo HTTP) y puerto 443(protocolo HTTPS) usados por nuestro servido web Apache.

El comando **netsat** 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:



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:

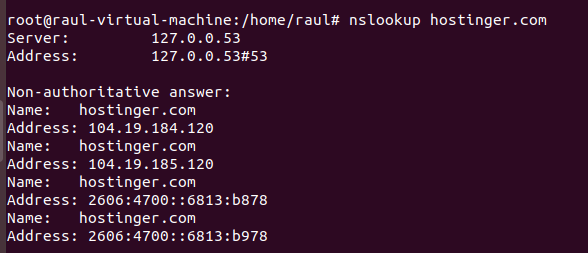


De esta salida interpretamos que el puerto 80 utilizado por nuestro servidor está listo para interactuar.

# 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:



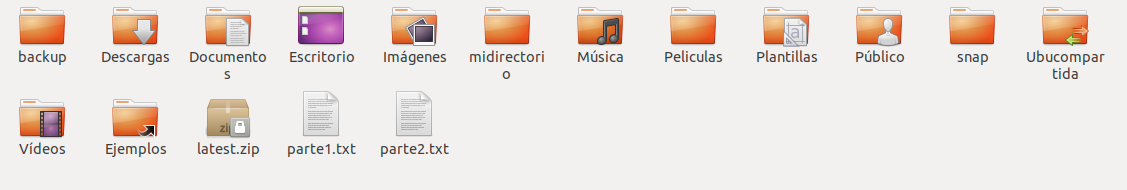


# 6.- ¿Cómo probamos que podemos acceder a un servidor? Pista: curl, wget.

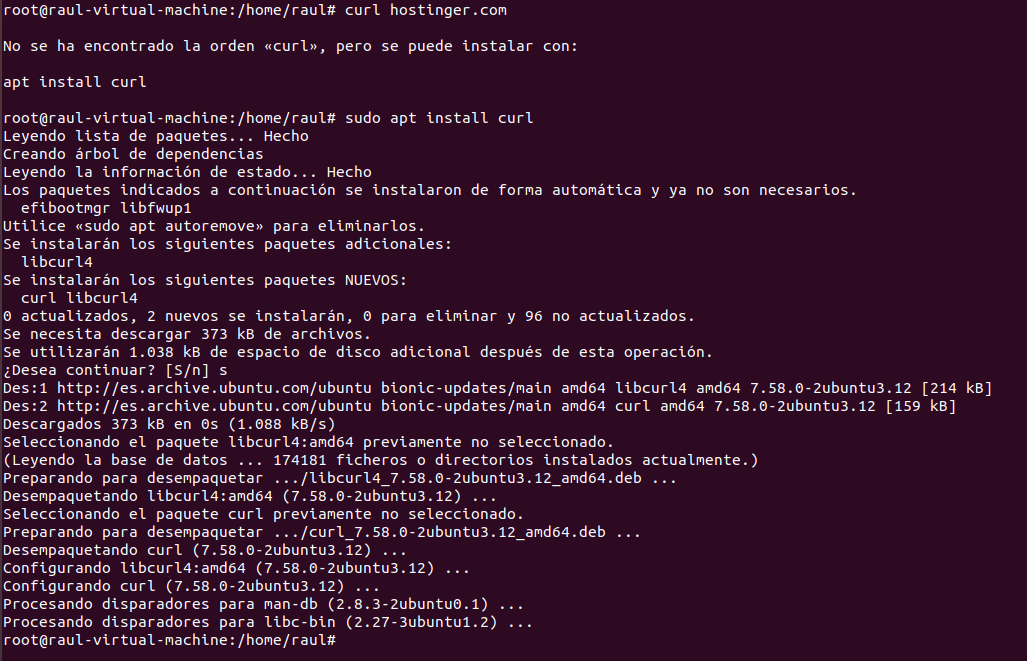
El comando **wget** nos permite acceder a un servidor y realizar una descarga de alguno de sus contenidos. Usando el comando **wget** [**https://wordpress.org/latest.zip**](https://wordpress.org/latest.zip) accederemos al servidor y descargaremos en nuestro directorio de trabajo el archivo latest.zip:



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**.

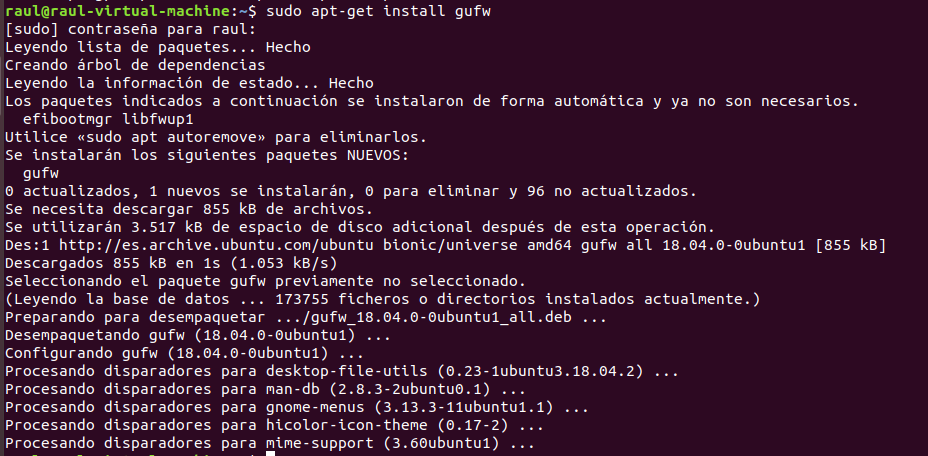




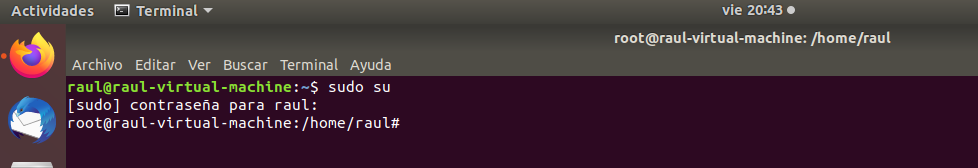
Esto generará el código fuente completo de la página de inicio del dominio. Si no se especifica ningún protocolo, curl lo interpretará a HTTP.

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

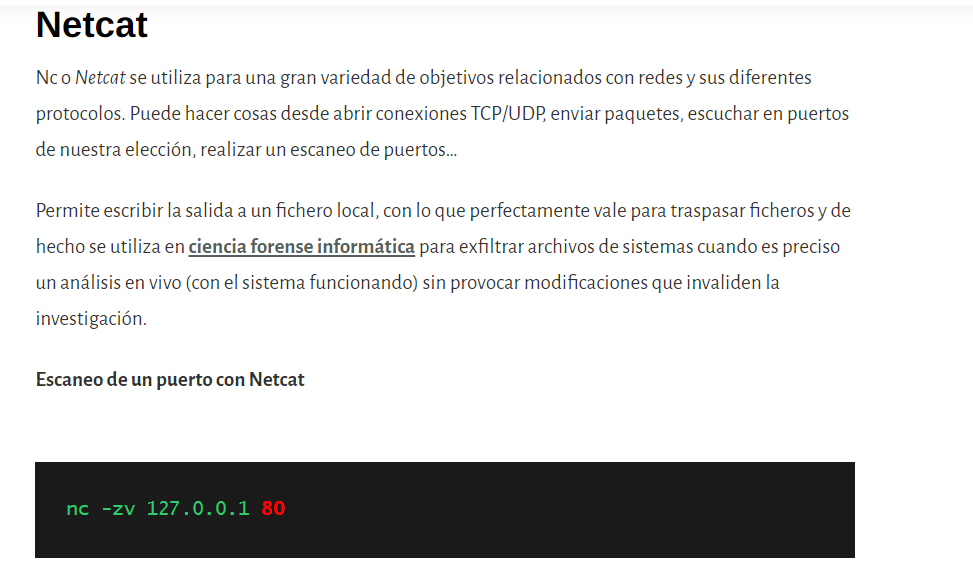
* **sudo apt –get install gufw**: con permisos de superusuario seguido de apt –get install seguido de la app que queremos instalar para instalar apps. En este caso no teniamos instalado ningún firewall asi que hemos instalado gufw:



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

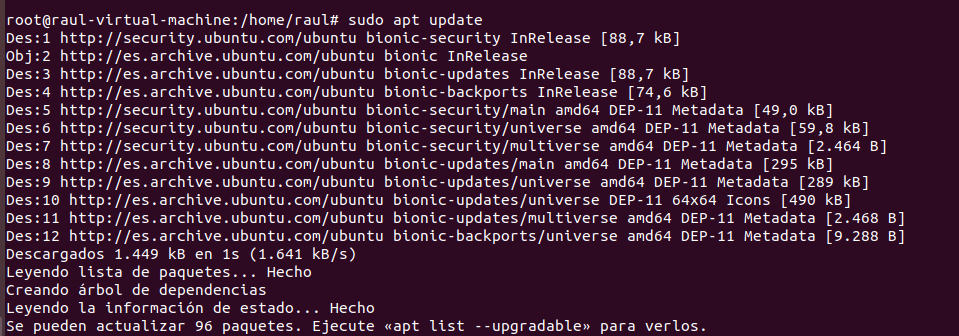


* **clear**: para limpiar la consola.
* **nc o Netcat**: netcat o nc nos permite escanear un puerto y ver si la conexión a él es satisfactoria

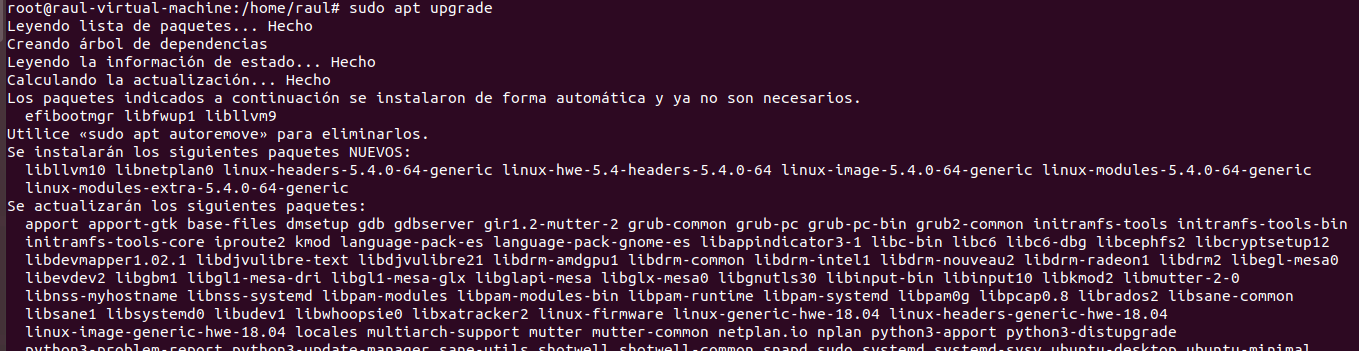


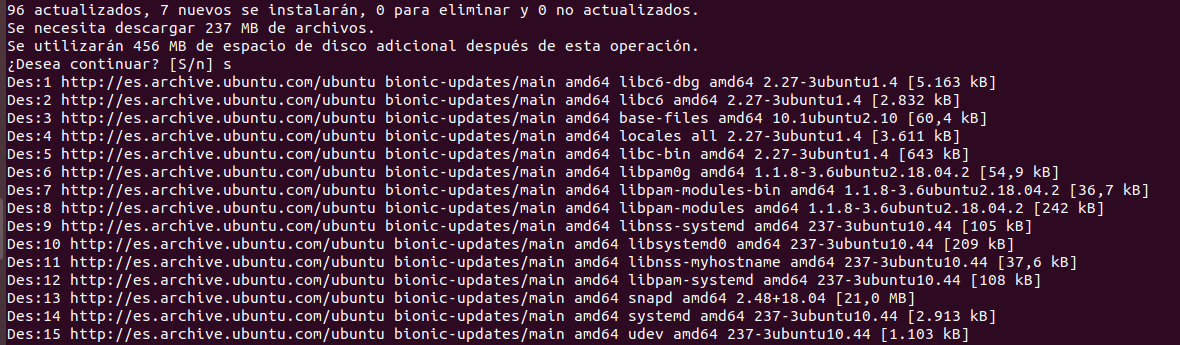


* **sudo apt update**: con este comando actualizamos el índice de los paquetes locales. Esto, para garantizar que en él se reflejen las cargas más recientes de las nuevas versiones de los paquetes.



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





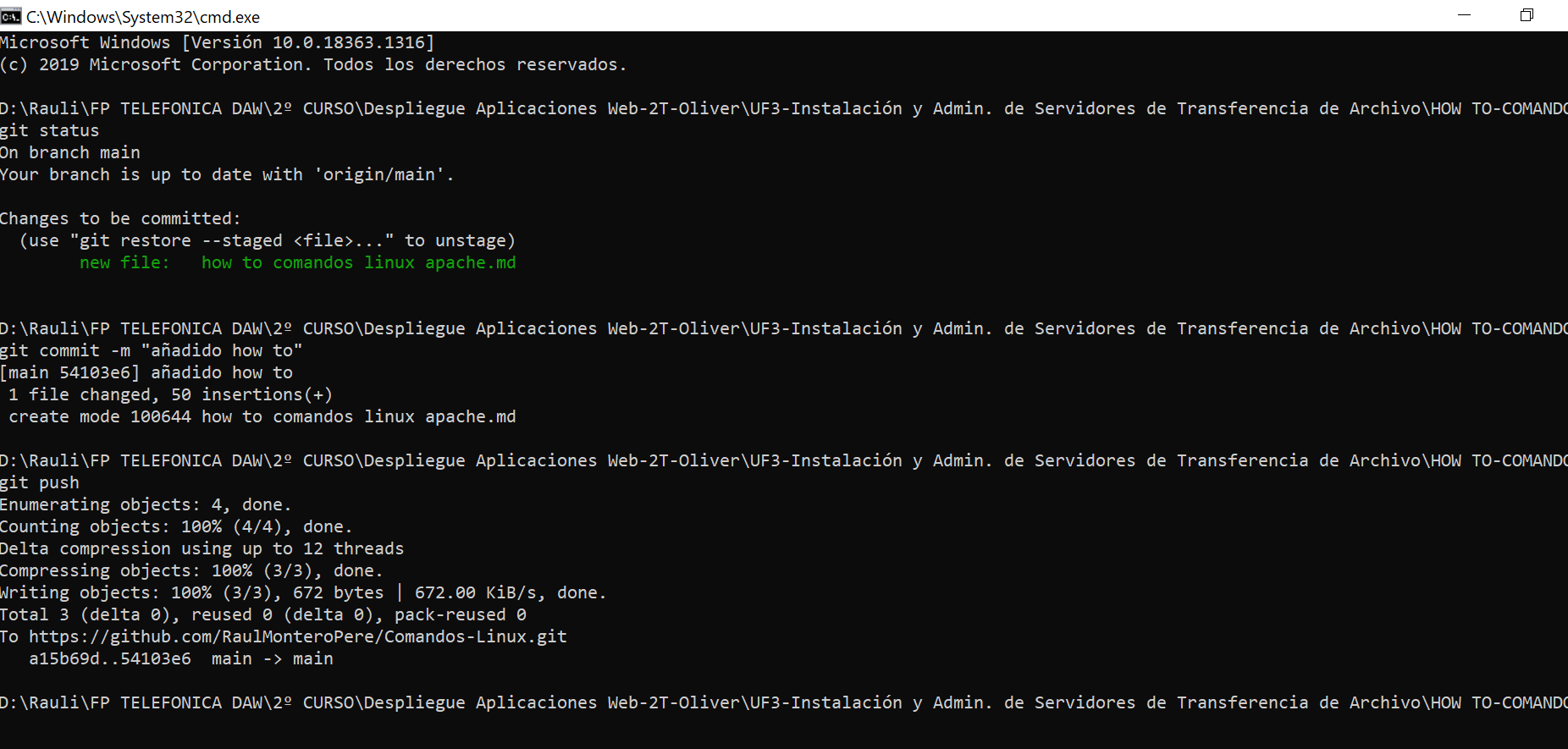
# 8.- Guía HOW TO en formato Markdown

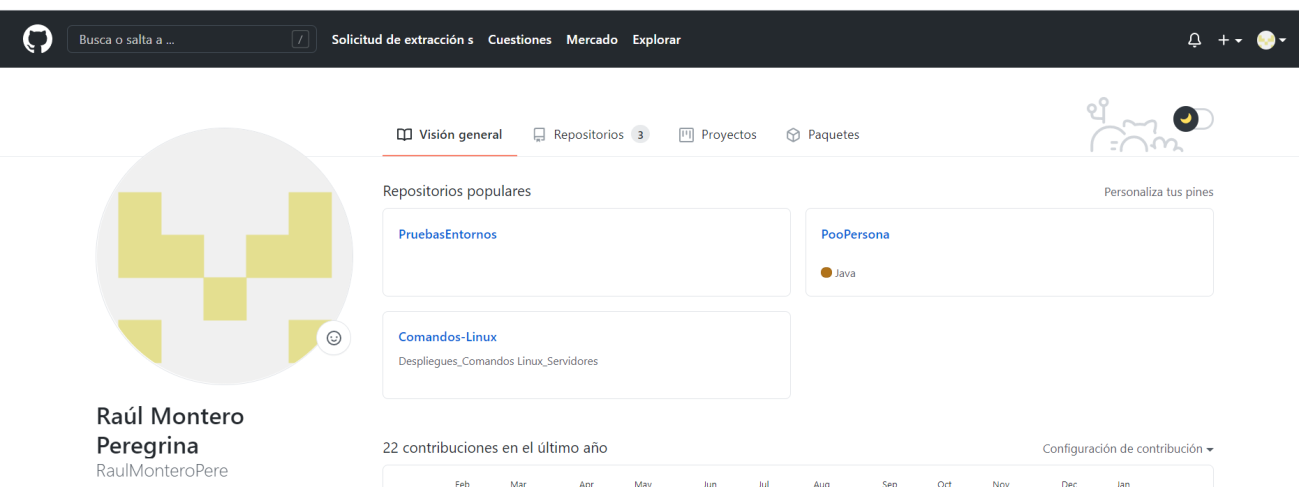
**Markdown** nació como herramienta de conversión de texto plano a HTML. Aunque en realidad Markdown también se considera un lenguaje que tiene la finalidad de permitir crear contenido de una manera sencilla de escribir, y que en todo momento mantenga un diseño legible, así que para simplificar puedes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

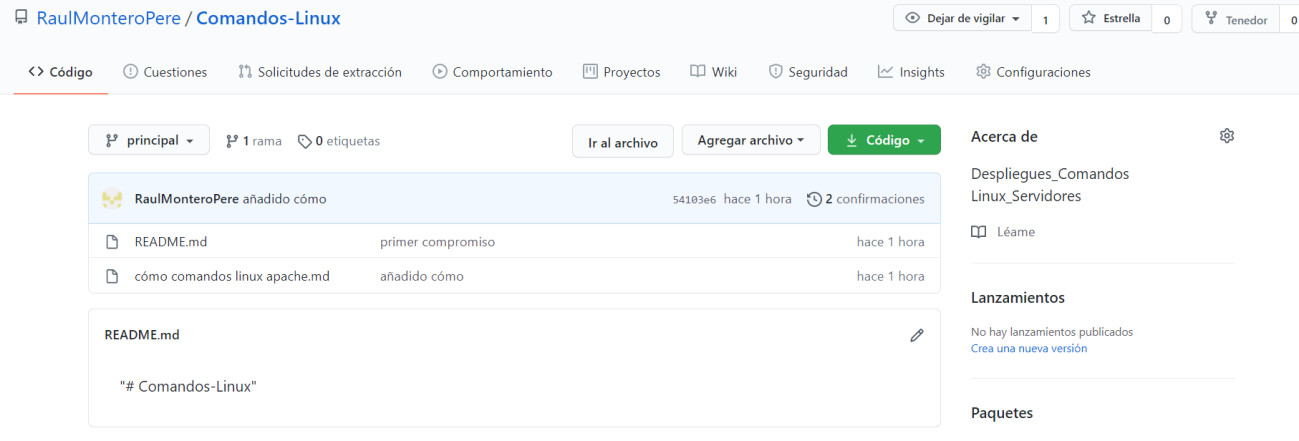
------------SE ADJUNTA GÚIA HOW TO EN FORMATO MARKDOWN----------

## 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:







Enlace al fichero en nuestro repositorio de GitHub: https://github.com/RaulMonteroPere/Comandos-Linux/blob/main/how%20to%20comandos%20linux%20apache.md

