

2º
DAW

Actividad 01

COMANDOS

Francisco José de Vicente Escolante



ÍNDICE

Pregunta	Página
1.- Saber si tenemos conexión a internet	02
2.- Saber si nuestro servidor es accesible desde internet	03
3.- Saber a quién pertenece una dirección web (url)	05
4.- Cómo comprobar acceso a un servidor	06
5.- Comandos extras	07
Repositorio GitHub	https://github.com/FranDeVicente/Actividad_01_DAW

Actividad 1. Tarea Individual. Comandos

Actividad de evaluación



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?

01.- ¿Cómo sabemos si tenemos acceso a internet?

Para saber si tenemos acceso a internet, lo primero que debemos conocer es nuestra dirección IP. Para ello, vamos a abrir la consola de nuestro Sistema Operativo que en mi caso, al utilizar un S.O. Windows7, abriré la **consola de comandos powerShell**. Una vez abierta, empezamos a trabajar con ella.

¿Cómo se usa? Lo usaremos escribiendo el comando "**ipconfig**" para conocer nuestra dirección IP.

¿Qué hace el comando? Observamos que nos aparece un listado con los datos de nuestra conexión red. En dicho listado, debemos localizar la tarjeta de red que estamos utilizando y nos fijaremos en la entrada **IPv4**, que será la que nos indique la dirección IP asignada a nuestro equipo.

```
Windows PowerShell
PS C:\Users\Chesco> ipconfig

Configuración IP de Windows

Adaptador de LAN inalámbrica Conexión de red inalámbrica 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de red inalámbrica:

    Sufijo DNS específico para la conexión. . : technicolor.net
    Dirección IPv6 local . . . . . : fe80::28c5:16c6:25a:ed25%14
    Dirección IPv4. . . . . : 192.168.0.121
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : 192.168.0.1
```

→ Esta es nuestra dirección IP

Ahora que ya conocemos nuestra dirección IP, vamos a comprobar si tenemos acceso a internet. Podemos comprobar nuestra conexión, utilizando el comando **ping**. Para ello, vamos a escribir **ping nuestraDirecciónIp**.

```
Windows PowerShell
PS C:\Users\Chesco> ping 192.168.0.121

Haciendo ping a 192.168.0.121 con 32 bytes de datos:
Respuesta desde 192.168.0.121: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.121: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.121: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.121: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.121:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

→ escribimos el comando ping junto a nuestra dirección IP

→ paquetes que envía, en éste caso 4, con los bytes que ocupan y el tiempo de tardanza

→ Nos indica el N° de paquetes enviados, que como vemos han sido los 4, y nos dice que se han recibido todos y no se ha perdido ninguno. Por tanto, estamos conectados correctamente a internet

¿Por qué responde a la pregunta? Si nos fijamos, vemos que nos indica el número de paquetes que envía, el tamaño en bytes que ocupa cada uno de ellos y el tiempo que tarda en enviarlo. Después, nos manda la estadística del **comando ping** para nuestra dirección IP y nos informa de los paquetes que ha enviado, recibido y perdido.

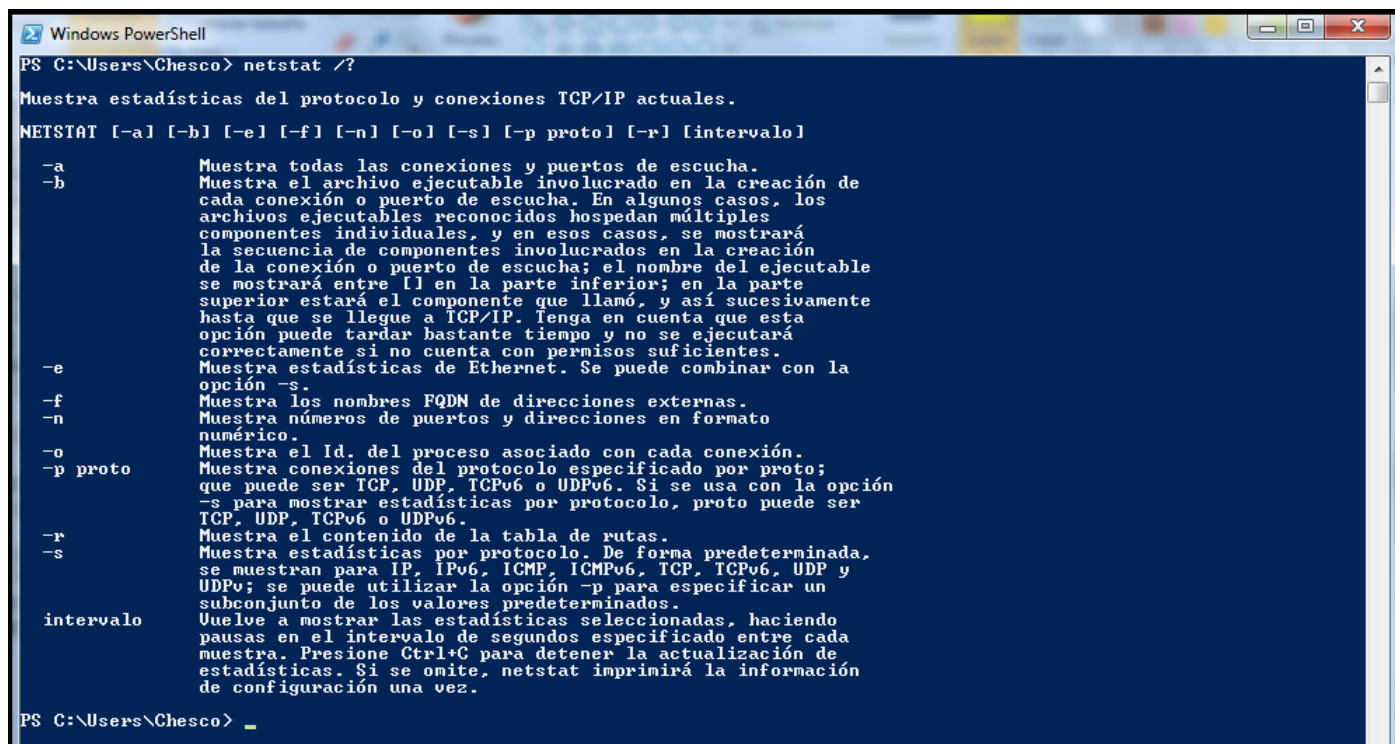
¿Cómo se interpreta la salida? Si recibimos el mensaje de que no se ha perdido ningún paquete "<0% perdidos>", significará que nuestra conexión a internet es correcta. Por el contrario, nos saldrá un mensaje diciendo "Tiempo de espera agotado para esta solicitud" lo que significará que habrá existido algún tipo de error de conexión, a causa del router por ejemplo, o lo que sea.

2.- ¿Cómo sabemos si nuestro servidor es accesible desde internet?

Para saber si nuestro servidor es accesible desde internet, debemos ver los puertos abiertos en nuestro Sistema Operativo.

¿Qué hace el comando? Al utilizar un S.O. Windows, necesitaremos el comando **netstat** ("ufw" para Linux).

Ahora bien, nos surge la duda de **¿Cómo se usa o qué hace éste comando?** para ello, podemos pedir ayuda a nuestra consola, escribiendo **netstat /?** que nos mostrará un menú de opciones como nuestro en la siguiente imagen:



```
Windows PowerShell
PS C:\Users\Chesco> netstat /?

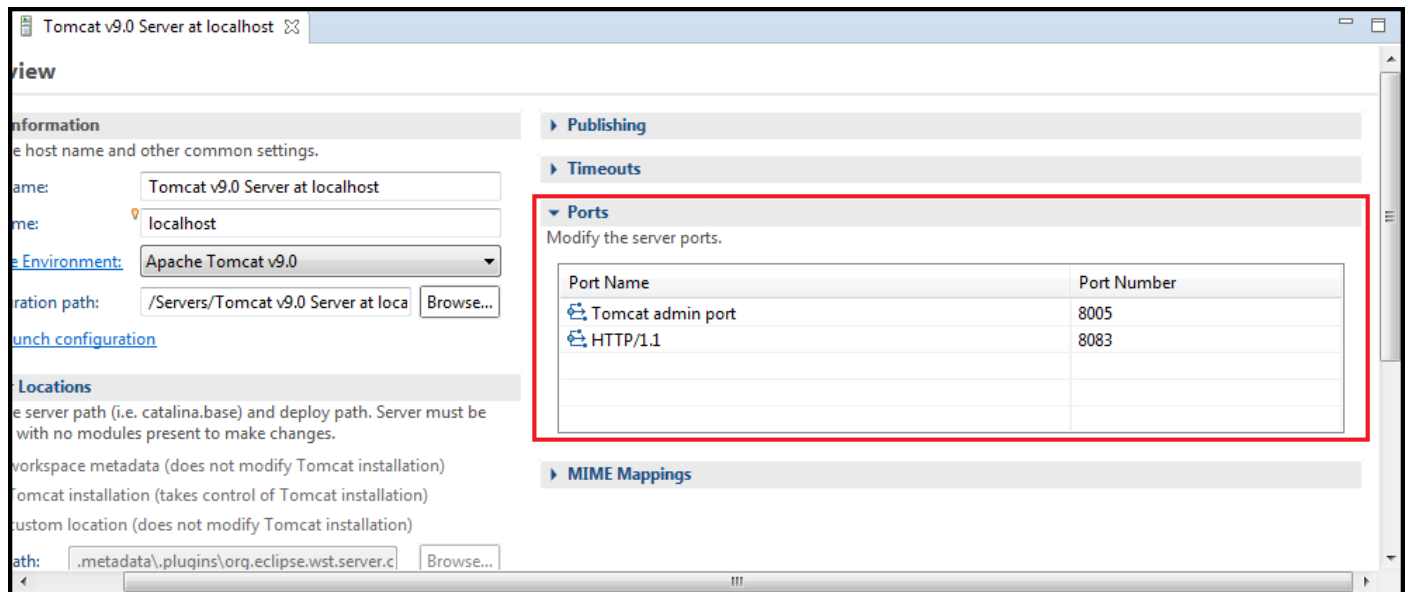
Muestra estadísticas del protocolo y conexiones TCP/IP actuales.
NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-s] [-p proto] [-r] [intervalo]

-a          Muestra todas las conexiones y puertos de escucha.
-b          Muestra el archivo ejecutable involucrado en la creación de
             cada conexión o puerto de escucha. En algunos casos, los
             archivos ejecutables reconocidos hospedan múltiples
             componentes individuales, y en esos casos, se mostrará
             la secuencia de componentes involucrados en la creación
             de la conexión o puerto de escucha; el nombre del ejecutable
             se mostrará entre [] en la parte inferior; en la parte
             superior estará el componente que llamó, y así sucesivamente
             hasta que se llegue a TCP/IP. Tenga en cuenta que esta
             opción puede tardar bastante tiempo y no se ejecutará
             correctamente si no cuenta con permisos suficientes.
-e          Muestra estadísticas de Ethernet. Se puede combinar con la
             opción -s.
-f          Muestra los nombres FQDN de direcciones externas.
-n          Muestra números de puertos y direcciones en formato
             numérico.
-o          Muestra el Id. del proceso asociado con cada conexión.
-p proto    Muestra conexiones del protocolo especificado por proto;
             que puede ser TCP, UDP, TCPv6 o UDPv6. Si se usa con la opción
             -s para mostrar estadísticas por protocolo, proto puede ser
             TCP, UDP, TCPv6 o UDPv6.
-r          Muestra el contenido de la tabla de rutas.
-s          Muestra estadísticas por protocolo. De forma predeterminada,
             se muestran para IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP y
             UDPv6; se puede utilizar la opción -p para especificar un
             subconjunto de los valores predeterminados.
intervalo   Vuelve a mostrar las estadísticas seleccionadas, haciendo
             pausas en el intervalo de segundos especificado entre cada
             muestra. Presione Ctrl+C para detener la actualización de
             estadísticas. Si se omite, netstat imprimirá la información
             de configuración una vez.

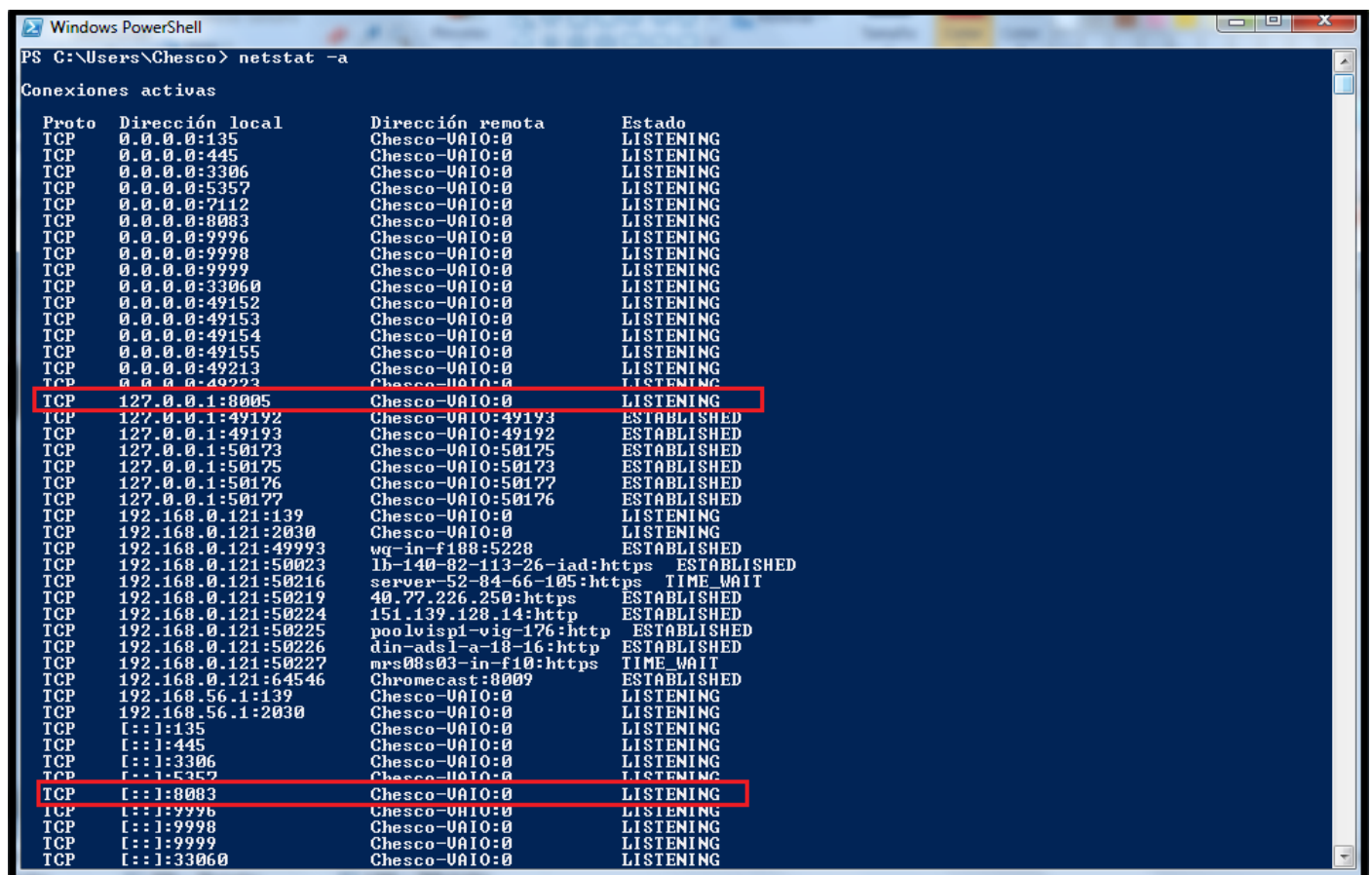
PS C:\Users\Chesco>
```

¿Por qué responde a la pregunta? Observamos que el comando **netstat -a** lo que hace es mostrar las estadísticas del protocolo y conexiones TCP/IP actuales. Como lo que queremos es saber si nuestro servidor es accesible desde internet, vamos a buscar nuestro servidor y ver qué ocurre.

En mi caso concreto, estoy usando un servidor Apache Tomcat que está utilizando el puerto **8083**



A continuación, utilizando `netstat -a` vemos que nuestro servidor sí que está escuchando, aunque en éste caso no es accesible desde internet si que es accesible de manera local. Por lo que, podemos saber si nuestro servidor es accesible desde internet mirando las conexiones activas y viendo si se encuentra en la columna "local" o "remota". Muestro aquí **cómo se comporta la salida**.



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

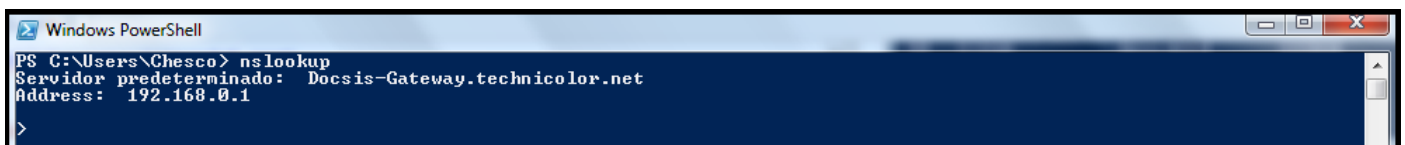
Para poder saber a quién pertenece una dirección web (URL), podemos utilizar tanto el comando **nslookup** como también el comando **dig** en windows, a partir de la versión 10. En mi caso como ya mencioné, mi S.O. es Windows 7 por tanto lo realizaré con **nslookup**. Pero... **¿qué hacen estos comandos?**

Ambos comandos son una herramienta muy útil que podemos utilizar para consultar servidores y permite a los usuarios realizar consultas a los distintos registros DNS.

Con **nslookup**, podemos obtener la dirección IP asociada a un nombre DNS y viceversa, además, podemos preguntar a los servidores de nombres información relativa a los registros de recursos (RR) de las zonas de las que son autorizados.

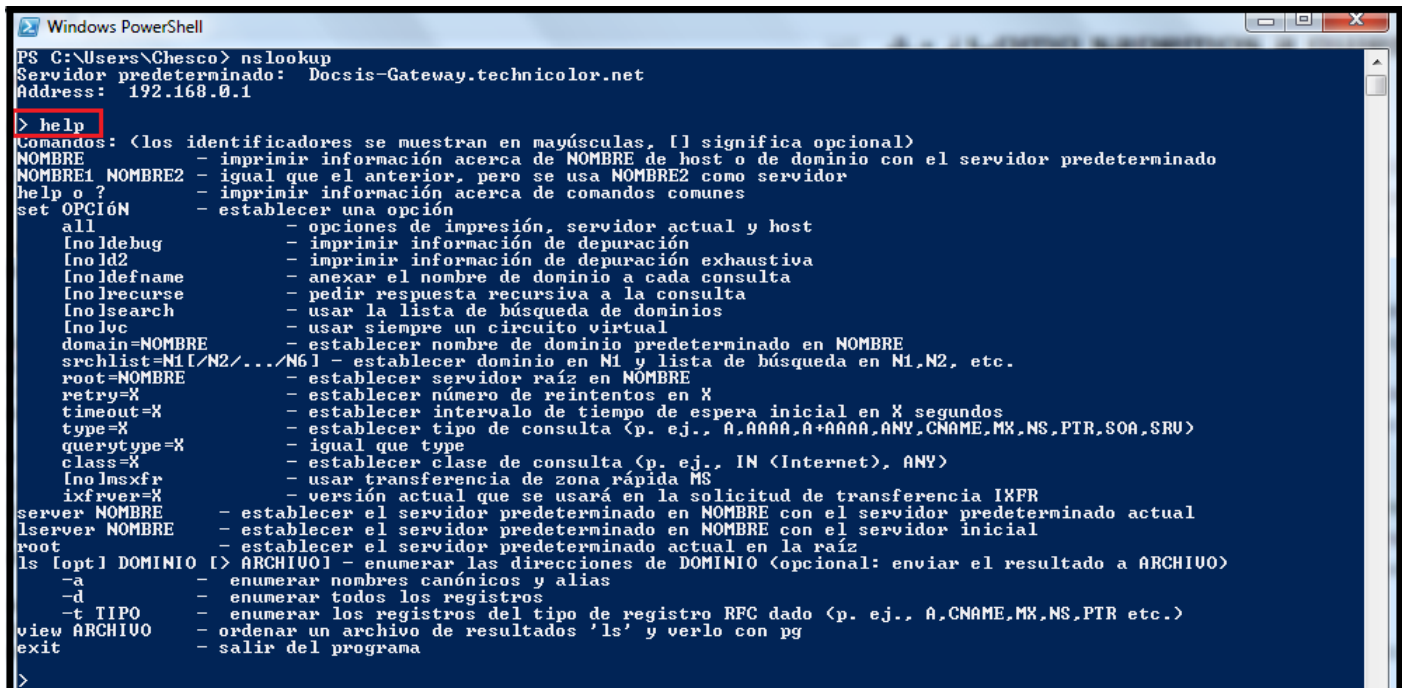
Nota: nslookup se usa de dos modos: interactivo y no interactivo. El modo interactivo permite al usuario consultar los servidores DNS para obtener información sobre varios hosts y dominios o para listar los hosts de un dominios. El modo no interactivo se usa para presentar solo el nombre y la información solicitada para un host o nombre DNS.

¿Cómo se usa? Primero tenemos que iniciar el comando en nuestro terminal. Para ello, simplemente lo escribimos y pulsamos "enter"



```
Windows PowerShell
PS C:\Users\Chesco> nslookup
Servidor predeterminado: Docsis-Gateway.technicolor.net
Address: 192.168.0.1
>
```

Podemos obtener ayuda escribiendo *help* para ver qué necesitamos saber en concreto y solicitarlo



```
Windows PowerShell
PS C:\Users\Chesco> nslookup
Servidor predeterminado: Docsis-Gateway.technicolor.net
Address: 192.168.0.1
> help
Comandos: (los identificadores se muestran en mayúsculas, [!] significa opcional)
NOMBRE - imprimir información acerca de NOMBRE de host o de dominio con el servidor predeterminado
NOMBRE1 NOMBRE2 - igual que el anterior, pero se usa NOMBRE2 como servidor
help o ? - imprimir información acerca de comandos comunes
set OPCION - establecer una opción
all - opciones de impresión, servidor actual y host
no debug - imprimir información de depuración
no ld2 - imprimir información de depuración exhaustiva
no ldname - anexar el nombre de dominio a cada consulta
no recurse - pedir respuesta recursiva a la consulta
no search - usar la lista de búsqueda de dominios
no loc - usar siempre un circuito virtual
domain=NOMBRE - establecer nombre de dominio predeterminado en NOMBRE
srchlist=N1[/N2/.../N6] - establecer dominio en N1 y lista de búsqueda en N1,N2, etc.
root=NOMBRE - establecer servidor raíz en NOMBRE
retry=X - establecer número de reintentos en X
timeout=X - establecer intervalo de tiempo de espera inicial en X segundos
type=X - establecer tipo de consulta (p. ej., A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRU)
querytype=X - igual que type
class=X - establecer clase de consulta (p. ej., IN (Internet), ANY)
no lmsxfr - usar transferencia de zona rápida MS
ixfrver=X - versión actual que se usará en la solicitud de transferencia IXFR
server NOMBRE - establecer el servidor predeterminado en NOMBRE con el servidor predeterminado actual
lserver NOMBRE - establecer el servidor predeterminado en NOMBRE con el servidor inicial
root - establecer el servidor predeterminado actual en la raíz
ls [opt] DOMINIO [-] ARCHIVO1 - enumerar las direcciones de DOMINIO (opcional: enviar el resultado a ARCHIVO)
-a - enumerar nombres canónicos y alias
-d - enumerar todos los registros
-t TIPO - enumerar los registros del tipo de registro RFC dado (p. ej., A,CNAME,MX,NS,PTR etc.)
view ARCHIVO - ordenar un archivo de resultados 'ls' y verlo con pg
exit - salir del programa
>
```

¿Por qué responde a mi pregunta? Porque si indicamos una dirección web, la que sea, nos dice a quién pertenece. Por poner un ejemplo, vamos a ver a quién pertenece la web (URL) del instituto (ITT). Yo le voy a decir que me de toda la información, por lo que usaré "set query=any" tal como nos indica en la ayuda del comando.

Así interpreta la salida:

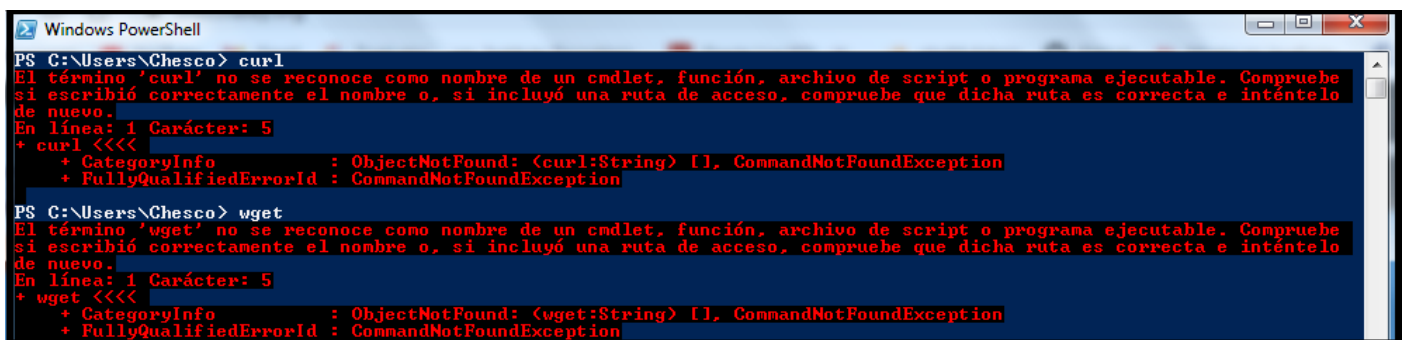
```
Respuesta no autoritativa:
institutotecnologico.telefonica.com      internet address = 195.76.34.244
> set query=any
> institutotecnologico.telefonica.com      ← dirección web (URL)
Servidor: Docsis-Gateway.technicolor.net
Address: 192.168.0.1

Respuesta no autoritativa:
institutotecnologico.telefonica.com      internet address = 195.76.34.244 ← A quién pertenece, su IP
>
```

4.- ¿Cómo probamos que podemos acceder a un servidor?

Para acceder a un servidor, podemos utilizar el comando **wget** ó **curl** ya que es una opción disponible para el usuario. Se trata de un programa de línea de comandos con la biblioteca correspondiente incluida, y que permite la transferencia de datos entre ordenadores interconectados a través del acceso al servidor.

Si probamos en Windows7 ambos comandos, sucede lo siguiente:

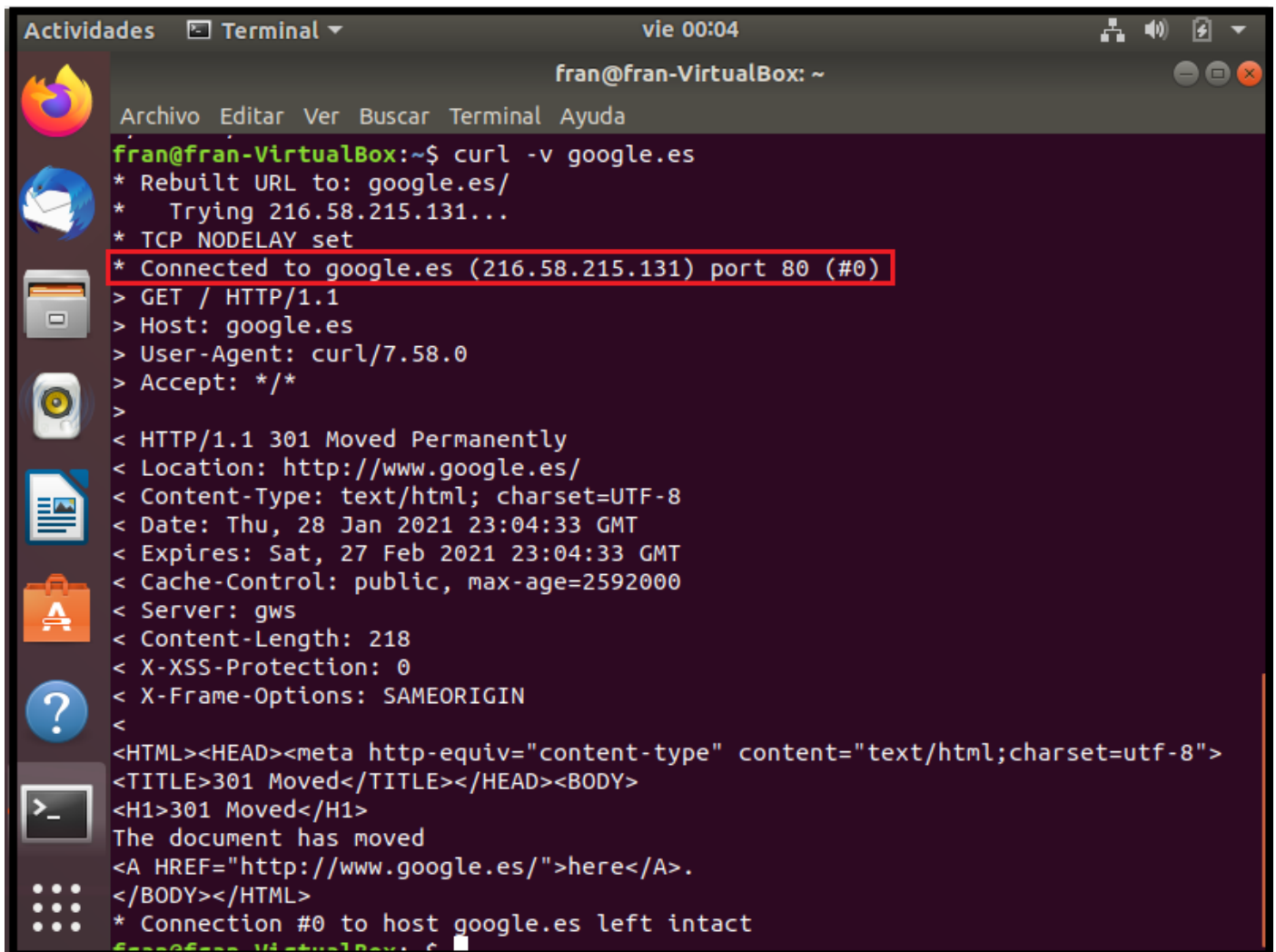


```
Windows PowerShell
PS C:\Users\Chesco> curl
El término 'curl' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 5
+ curl <<<<
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (curl:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Chesco> wget
El término 'wget' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 5
+ wget <<<<
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (wget:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

Nota Importante (Personal): Tras horas de intentos y búsquedas, no he conseguido avanzar en éste comando. Creo que viene dado por mi sistema operativo y la versión del powerShell que por lo que he investigado, tiene que ser a partir de la versión 5. Por lo que opto finalmente ha realizar dicho comando mediante una máquina virtual de ubuntu.

¿Cómo se usa? Si usamos el comando "curl servidor" (suponiendo que el servidor que usamos sea el más conocido, google) sería "curl google.es" y **el su compartamiento de su salida** sería sacarnos sólo el "head" de la cabecera del html de google y sin embargo, si utilizamos la sentencia "curl -v google.es", el comportamiento de salida, sería la conexión con el servidor. Podemos ver que conecta por el puerto 80



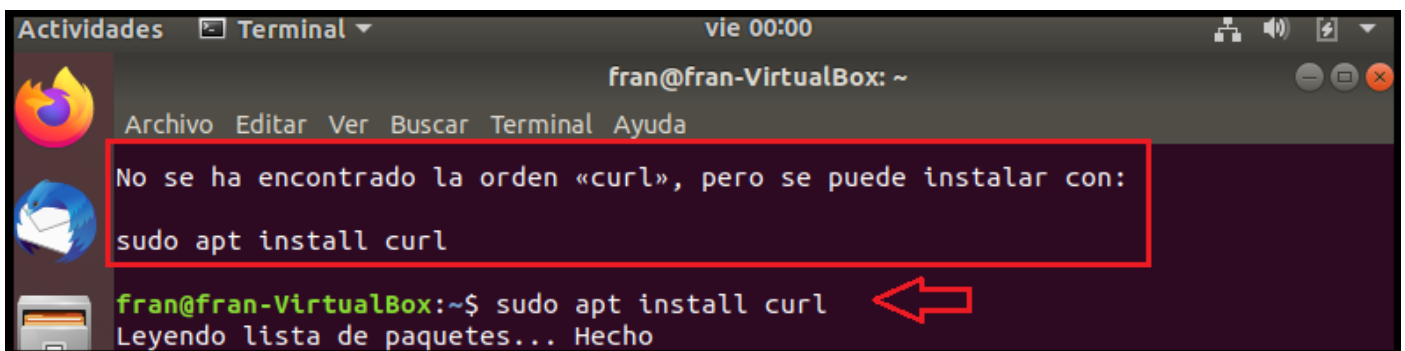
A terminal window titled 'Terminal' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda) and a status bar (vie 00:04). The prompt is 'fran@fran-VirtualBox: ~'. The command 'curl -v google.es' has been executed. The output shows the connection details and the HTTP response from google.es, which is a 301 Moved Permanently redirect to http://www.google.es/. A red box highlights the line '* Connected to google.es (216.58.215.131) port 80 (#0)'.

```
fran@fran-VirtualBox:~$ curl -v google.es
* Rebuilt URL to: google.es/
* Trying 216.58.215.131...
* TCP_NODELAY set
* Connected to google.es (216.58.215.131) port 80 (#0)
> GET / HTTP/1.1
> Host: google.es
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Location: http://www.google.es/
< Content-Type: text/html; charset=UTF-8
< Date: Thu, 28 Jan 2021 23:04:33 GMT
< Expires: Sat, 27 Feb 2021 23:04:33 GMT
< Cache-Control: public, max-age=2592000
< Server: gws
< Content-Length: 218
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
<
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.es/">here</A>.
</BODY></HTML>
* Connection #0 to host google.es left intact
```

Responde a la pregunta porque utilizando la sentencia anterior, podemos probar que hemos accedido al servidor, en éste caso el de google.es

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

En este caso, he tenido que utilizar un comando para poder instalar el último comando **curl**. El comando usado, te lo facilita el propio terminal de ubuntu y es "sudo apt install curl".



A terminal window titled 'Terminal' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda) and a status bar (vie 00:00). The prompt is 'fran@fran-VirtualBox: ~'. The message 'No se ha encontrado la orden «curl», pero se puede instalar con:' is displayed. Below it, the command 'sudo apt install curl' is shown. A red box highlights this message and command. The prompt then changes to 'fran@fran-VirtualBox:~\$ sudo apt install curl' and the output 'Leyendo lista de paquetes... Hecho' is shown. A red arrow points to the command line.

```
fran@fran-VirtualBox:~$ sudo apt install curl
Leyendo lista de paquetes... Hecho
```