



Actividad 1. Tarea Individual. Comandos.

Pilar Bermejo 2 DAW

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?

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

GUÍA DE USO:

1. ¿CÓMO SABEMOS SI TENEMOS CONEXIÓN A INTERNET?

PING: Este comando ping es una de las herramientas básicas de diagnóstico de redes.

1.1 ¿Qué hace?

Permite comprobar si es posible establecer una conexión IP hacia un determinado host mediante el protocolo **ICMP (Protocolo de Mensajes de Control de Internet)**. De esta forma se puede determinar **si el ruteo de paquetes IP desde y hacia el host destino funciona correctamente.**

1.2 ¿Cómo se usa?

`$ ping dominioejemplo.com` (podría ser también una IP si la conocemos.

- Especificando el número ECHOREQUEST: *especificar el número de paquetes o solicitudes que el usuario desea realizar.*

`$ ping -c dominioejemplo.com` (es el número de pings que deseas realizar.)

- Con ping audible: emite un pitido para verificar si el host está activo o no.

`$ ping -a dominioejemplo.com`

- Estableciendo intervalos: La opción `-i` permite al usuario establecer intervalos en segundos entre cada paquete.

\$ `ping -i 2 -c 7 dominioejemplo.com`. Los números 2 y 7 los puedes cambiar por el tiempo que desees.

- Recibiendo solo el resumen del comando Ping de Linux:

\$ `ping -c 7 -q dominioejemplo.com`. Usando `-c 7` para realizar siete solicitudes, pero solo recibimos el resumen debido a que se agregó `-q`.

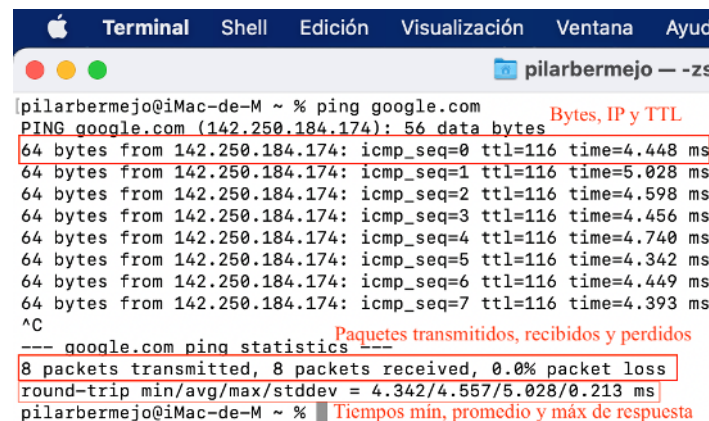
- Inundando la red con el comando Ping: permite a los usuarios enviar 100 o más paquetes por segundo, con la ayuda del siguiente comando:

\$ `ping -f dominioejemplo.com`

1.3 ¿Por qué responde a la pregunta?

Porque si no tenemos ningún problema al ejecutarlo y se enviaron los paquetes y devolvieron es que tenemos conexión a internet y funciona perfectamente.

1.4 ¿Cómo se interpreta la salida?



```
Terminal  Shell  Edición  Visualización  Ventana  Ayud
pilarbermejo@iMac-de-M ~ % ping google.com
PING google.com (142.250.184.174): 56 data bytes
64 bytes from 142.250.184.174: icmp_seq=0 ttl=116 time=4.448 ms
64 bytes from 142.250.184.174: icmp_seq=1 ttl=116 time=5.028 ms
64 bytes from 142.250.184.174: icmp_seq=2 ttl=116 time=4.598 ms
64 bytes from 142.250.184.174: icmp_seq=3 ttl=116 time=4.456 ms
64 bytes from 142.250.184.174: icmp_seq=4 ttl=116 time=4.740 ms
64 bytes from 142.250.184.174: icmp_seq=5 ttl=116 time=4.342 ms
64 bytes from 142.250.184.174: icmp_seq=6 ttl=116 time=4.449 ms
64 bytes from 142.250.184.174: icmp_seq=7 ttl=116 time=4.393 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 4.342/4.557/5.028/0.213 ms
pilarbermejo@iMac-de-M ~ %
```

El resultado indica que se enviaron ocho paquetes de prueba de 64 bytes desde el host 216.58.211.238 y se devolvieron. TTL son las siglas de tiempo de vida, la latencia, que es el tiempo de conexión.

TRACEROUTE: Es otro comando que puede ser bastante útil en caso de que la **conexión con el host haya sido perdida**, porque podemos entender a qué punto la conexión ha sido perdida.

1.1 ¿Qué hace?

Determina la ruta para alcanzar un host mediante el envío de paquetes "echo" de **protocolo ICMP** a la dirección de destino con un tiempo de vida TTL. Si el TTL antes de reenviarlo llega a 0 antes de alcanzar el destino nos devuelve el mensaje "Tiempo de espera agotado..."

1.2 ¿Cómo se usa?

`$ traceroute dominioejemplo.com` (o como en el ping, la IP si la conocemos).

1.3 ¿Por qué responde a la pregunta?

Porque podemos ver los diferentes **saltos en los diferentes nodos porque se establece la conexión**, si no se establece y no tuviéramos conexión a internet nos devolvería el mensaje mencionado anteriormente.

1.4 ¿Cómo se interpreta la salida?

```
pilarbermejo@iMac-de-M ~ % traceroute google.com
traceroute to google.com (216.58.209.78), 64 hops max, 52 byte packets
 1  192.168.1.1 (192.168.1.1)  2.841 ms  12.683 ms  9.579 ms
 2  * * *
 3  * * *
 4  10.red-81-46-66.customer.static.ccg.telefonica.net (81.46.66.10)  4.680 ms  4.440 ms  4.145 ms
 5  * 17.red-81-46-0.customer.static.ccg.telefonica.net (81.46.0.17)  5.101 ms *
 6  * 176.52.253.93 (176.52.253.93)  4.542 ms
 7  209.85.149.88 (209.85.149.88)  4.636 ms
   176.52.253.102 (176.52.253.102)  4.749 ms
   5.53.0.110 (5.53.0.110)  5.104 ms
 8  74.125.242.177 (74.125.242.177)  5.696 ms  5.782 ms
   74.125.242.161 (74.125.242.161)  4.225 ms
 9  142.250.46.167 (142.250.46.167)  5.200 ms
   mad07s22-in-f14.1e100.net (216.58.209.78)  4.746 ms
  108.170.253.241 (108.170.253.241)  5.423 ms
pilarbermejo@iMac-de-M ~ %
```

Lo primero que hace el comando al ser ejecutado es mostrar la dirección **IP real del dominio de la web que hemos elegido**. Luego nos dice todos los nodos y routers por los que pasa el mensaje de prueba que hemos enviado, sus direcciones IP y la **latencia** de cada uno de ellos hasta llegar a su destino.

IFCONFIG:

1.1 ¿Qué hace?

Muestra el estado de las **interfaces actualmente activas**. Información de la IP de cada interface y la dirección MAC, IPV6, IPV4...

1.2 ¿Cómo se usa?

Si ponemos solo el **comando ifconfig** muestra el estado de todas las interfaces, incluso las que están inactivas.

Si ponemos **ifconfig seguido del nombre de la interface**, solo muestra el estado de la interface que hemos indicado.

`$ ifconfig` Muestra el estado de todas las interfaces, incluso las que están inactivas.

`$ ifconfig lo0` (Accedemos a loopback 0, dirección especial que los hosts utilizan para dirigir el tráfico hacia ellos mismos)

1.3 ¿Por qué responde a la pregunta?

Porque sabemos si nos ha asignado una IP de Red y sabemos si el **puerto está activo y transmitiendo**.

1.4 ¿Cómo se interpreta la salida?

```
pilarbermejo@iMac-de-M ~ % ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000 IP
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=50b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV,CHANNEL_IO>
    ether a8:60:b6:2a:cb:cb
    inet6 fe80::46d:87a:8b83:ea4a%en0 prefixlen 64 secured scopeid 0x4
    inet 192.168.1.51 netmask 0xfffff000 broadcast 192.168.1.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect (1000baseT <full-duplex,flow-control>)
    status: active Estado del puerto
en1: flags=8823<UP,BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether 48:bf:6b:f3:f6:c4
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect (<unknown type>)
    status: inactive
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=460<TSO4,TSO6,CHANNEL_IO>
    ether 82:19:0c:f0:ce:80 Máscara de Red
    media: autoselect <full-duplex>
    status: inactive
en3: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=460<TSO4,TSO6,CHANNEL_IO>
    ether 82:19:0c:f0:ce:81
    media: autoselect <full-duplex>
    status: inactive
```

Puedo ver si el puerto está activo, la dirección IP que tiene y la MAC.

2. ¿CÓMO SABEMOS SI NUESTRO SERVIDOR ES ACCESIBLE DESDE INTERNET?

NETSTAT:

1.1 ¿Qué hace?

Nos permite **supervisar las conexiones de red** tanto entrantes como salientes, así como ver tablas de enrutamiento, estadísticas de la interfaz, etc.

1.2 ¿Cómo se usa?

\$ *netstat -a* (lista todos los puertos TCP y UDP en estado de escucha)

\$ *netstat -at* (solo puertos TCP)

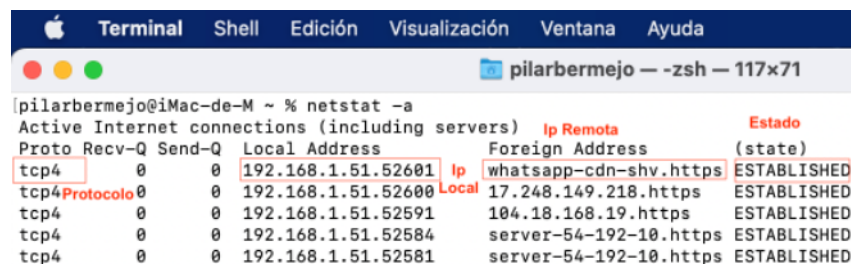
\$ *netstat -au* (solo puertos UDP)

\$ `netstat -l` (Todas las conexiones en estado de escucha, con `-lt` todos los TCP y con `-lu` todos los UDP)

1.3 ¿Por qué responde a la pregunta?

Nos indica qué **puertos están abiertos y si los programas están escuchando** en los puertos permitiéndonos llevar un mejor control sobre estos. Nos ofrece una vía fácil para supervisar y solucionar problemas relacionados con la red y determinar el rendimiento del tráfico de red

1.4 ¿Cómo se interpreta la salida?



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	Estado
tcp4	0	0	192.168.1.51.52601	whatsapp-cdn-shv.https	ESTABLISHED
tcp4	0	0	192.168.1.51.52600	17.248.149.218.https	ESTABLISHED
tcp4	0	0	192.168.1.51.52591	104.18.168.19.https	ESTABLISHED
tcp4	0	0	192.168.1.51.52584	server-54-192-10.https	ESTABLISHED
tcp4	0	0	192.168.1.51.52581	server-54-192-10.https	ESTABLISHED

Tipo de protocolo, dirección local y remota y el Estado.

UFW:

Comando que nos permite configurar reglas y administrar el firewall. Pero no hemos podido ejecutarlo en el Terminal de MAC.

3. ¿CÓMO SABEMOS A QUIÉN PERTENECE UNA DIRECCIÓN WEB (URL)?

HOST:

1.1 ¿Qué hace?

Devuelve los registros DNS (habilita un enlace entre nombres de dominio y direcciones IP con la que están asociados) **configurados para un dominio**. No incluye los registros configurados para los subdominios.

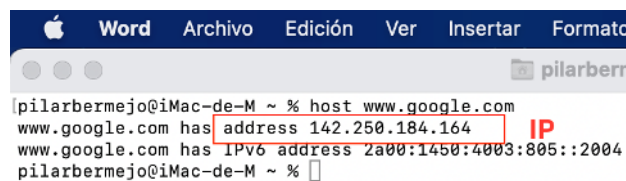
1.2 ¿Cómo se usa?

`$ host nombrededominio.com`

1.3 ¿Por qué responde a la pregunta?

Porque con la información que devuelve **obtenemos la dirección IP** a la que está asociado ese dominio.

1.4 ¿Cómo se interpreta la salida?



```
pilarbermejo@iMac-de-M ~ % host www.google.com
www.google.com has address 142.250.184.164 IP
www.google.com has IPv6 address 2a00:1450:4003:805::2004
pilarbermejo@iMac-de-M ~ %
```

Nos devuelve la dirección IP .

DIG:

1.1 ¿Qué hace?

Ofrece la misma consulta que HOST pero con más información

1.2 ¿Cómo se usa?

`$ dig nombrededominio.com`

1.3 ¿Por qué responde a la pregunta?

A devolvernos los registros DNS podemos **ver a la IP a la que está asociado el dominio**.

1.4 ¿Cómo se interpreta la salida?

```
pilarbermejo@iMac-de-M ~ % dig www.google.com

;<<> DiG 9.10.6 <<> www.google.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 33399
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1460
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                81      IN      A      172.217.17.4

;; Query time: 77 msec
;; SERVER: 80.58.61.250#53(80.58.61.250)
;; WHEN: Sat Jan 23 20:38:43 CET 2021
;; MSG SIZE rcvd: 59
```

NSLOOKUP:

1.1 ¿Qué hace?

Nos permite averiguar si el **servidor DNS que tenemos configurado está traduciendo correctamente las URL** a sus correspondientes direcciones IP.

1.2 ¿Cómo se usa?

Primero ejecutamos el comando y a continuación las URL que queramos conocer las IP's.

```
$ nslookup
```

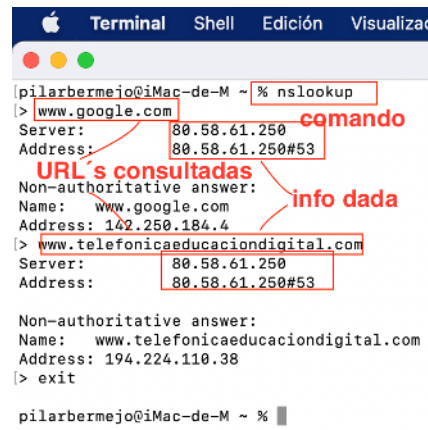
```
www.google.com
```

```
www.telefonicaeducaciondigital.com
```

1.3 ¿Por qué responde a la pregunta?

Nos da información sobre un dominio determinado.

1.4 ¿Cómo se interpreta la salida?



```
pilarbermejo@iMac-de-M ~ % nslookup
> www.google.com
Server:      80.58.61.250
Address:     80.58.61.250#53
Non-authoritative answer:
Name:   www.google.com
Address: 142.250.184.4
> www.telefonicaeducaciondigital.com
Server:      80.58.61.250
Address:     80.58.61.250#53
Non-authoritative answer:
Name:   www.telefonicaeducaciondigital.com
Address: 194.224.110.38
> exit

pilarbermejo@iMac-de-M ~ %
```

Annotations in the image:

- comando**: points to the `nslookup` command.
- URL's consultadas**: points to the domain names `www.google.com` and `www.telefonicaeducaciondigital.com`.
- info dada**: points to the IP addresses returned in the output.

WHOIS:

1.1 ¿Qué hace?

Sirve para consultar la información sobre un dominio de Internet, **quién es el dueño del dominio**, cuándo expira el dominio, quién es el registrar o registrador del dominio, sus DNS-s, etc...

1.2 ¿Cómo se usa?

`$ whois nombredeldominio.com`

1.3 ¿Por qué responde a la pregunta?

Porque nos ofrece datos de quién es el dueño del dominio o donde está registrado, DNS, etc...

1.4 ¿Cómo se interpreta la salida?

```
Terminal Shell Edición Visualización Ventana Ayuda
pilarbermejo@iMac-de-M ~ % whois www.telefonicaeducaciondigital.com
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:      whois.verisign-grs.com
domain:     COM

organisation: VeriSign Global Registry Services
address:    12061 Bluemont Way
address:    Reston Virginia 20190
address:    United States

contact:    administrative
name:       Registry Customer Service
organisation: VeriSign Global Registry Services
address:    12061 Bluemont Way
address:    Reston Virginia 20190
address:    United States
phone:      +1 703 925-6999
fax-no:     +1 703 948 3978
e-mail:     info@verisign-grs.com

contact:    technical
name:       Registry Customer Service
organisation: VeriSign Global Registry Services
address:    12061 Bluemont Way
address:    Reston Virginia 20190
address:    United States
phone:      +1 703 925-6999
fax-no:     +1 703 948 3978
e-mail:     info@verisign-grs.com

nslookup:    A.GTLD-SERVERS.NET 192.5.6.30 2001:503:a83e:0:0:0:2:30
nslookup:    B.GTLD-SERVERS.NET 192.33.14.30 2001:503:231d:0:0:0:2:30
nslookup:    C.GTLD-SERVERS.NET 192.26.92.30 2001:503:83eb:0:0:0:0:30
nslookup:    D.GTLD-SERVERS.NET 192.31.80.30 2001:500:856e:0:0:0:0:30
nslookup:    E.GTLD-SERVERS.NET 192.12.94.30 2001:502:1ca1:0:0:0:0:30
nslookup:    F.GTLD-SERVERS.NET 192.35.51.30 2001:503:d414:0:0:0:0:30
nslookup:    G.GTLD-SERVERS.NET 192.42.93.30 2001:503:eea3:0:0:0:0:30
nslookup:    H.GTLD-SERVERS.NET 192.54.112.30 2001:502:8cc:0:0:0:0:30
nslookup:    I.GTLD-SERVERS.NET 192.43.172.30 2001:503:39c1:0:0:0:0:30
nslookup:    J.GTLD-SERVERS.NET 192.48.79.30 2001:502:7094:0:0:0:0:30
nslookup:    K.GTLD-SERVERS.NET 192.52.178.30 2001:503:d2d:0:0:0:0:30
nslookup:    L.GTLD-SERVERS.NET 192.41.162.30 2001:500:d937:0:0:0:0:30
nslookup:    M.GTLD-SERVERS.NET 192.55.83.30 2001:501:b1f9:0:0:0:0:30
ds-rdata:    30909 8 2 E2D3C916F6DEEAC73294E8268FB5885044A833FC5459588F4A9184CFC41A5766

whois:      whois.verisign-grs.com

status:     ACTIVE
remarks:    Registration information: http://www.verisigninc.com
created:    1985-01-01
changed:    2017-10-05
source:     IANA

# whois.verisign-grs.com

No match for domain "WWW.TELEFONICAEDUCACIONDIGITAL.COM".
>>> Last update of whois database: 2021-01-23T20:59:09Z <<<
```

Contactos

DNS's

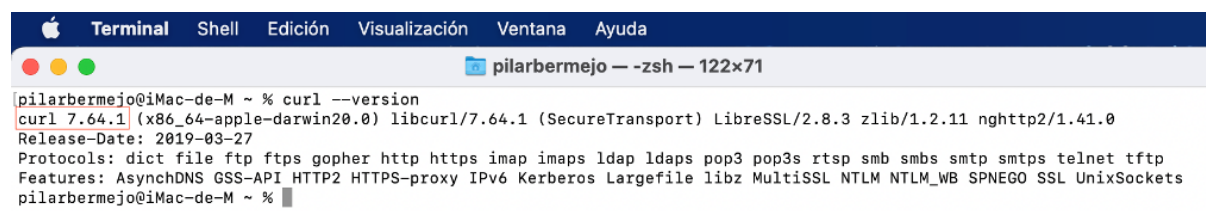
Estado

Nos da la información del registrador con contactos administrativo y técnico, en esta ocasión nos devuelve Verisign que opera una gran variedad de infraestructuras de red que incluye dos de los trece servidores de internet.

4. ¿CÓMO PROBAMOS QUE PODEMOS ACCEDER A UN SERVIDOR?

CURL:

Lo primero es comprobar que Curl está instalado con el comando `$ curl --version`



```
pilarbermejo@iMac-de-M ~ % curl --version
curl 7.64.1 (x86_64-apple-darwin20.0) libcurl/7.64.1 (SecureTransport) LibreSSL/2.8.3 zlib/1.2.11 nghttp2/1.41.0
Release-Date: 2019-03-27
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtsp smb smbs smtp smtps telnet tftp
Features: AsynchDNS GSS-API HTTP2 HTTPS-proxy IPv6 Kerberos Largefile libz MultiSSL NTLM NTLM_WB SPNEGO SSL UnixSockets
pilarbermejo@iMac-de-M ~ %
```

1.1 ¿Qué hace?

La función de Curl es similar a la de Wget que veremos más adelante. Curl no es un software especializado en descargas, sino que está **diseñado para la comunicación general a través de redes**, por lo que ofrece aún más ventajas.

Con este comando podremos **comprobar si un servidor es accesible**, el funcionamiento es similar al del comando ping visto anteriormente, sin embargo, debido a los protocolos y las opciones disponibles, Curl se puede utilizar con más flexibilidad. Además, ping trabaja en la capa de Internet, mientras que Curl funciona sobre la capa de aplicación. Eso significa que ping verifica si la máquina está en red. En cambio, el Curl siguiente comprueba si un servidor reacciona y cómo lo hace.

1.2 ¿Cómo se usa?

`$ curl nombrededominio.com` (es el uso más simple) Y obtenemos si podemos acceder a la URL.

Si el servidor no se puede conectar obtendremos un error del tipo (Could not resolve host)

Opciones de archivo de comandos CURL:

-O guarda el archivo en el directorio de trabajo actual con el mismo nombre de archivo que el remoto.

-o permite especificar un nombre de archivo o ubicación diferente.

Tiene más usos cuando tienes un servidor Proxy para hacer una solicitud GET o POST. También se puede usar para verificar que cookie se descargan en cualquier URL. También es compatible con FTP y se pueden descargar archivos de un servidor remoto.

`$ curl -v nombredominio.com` Si deseamos validar.

1.3 ¿Por qué responde a la pregunta?

El comando nos dice si podemos acceder a la URL o si no podemos conectar al servidor.

1.4 ¿Cómo se interpreta la salida?

```
pilarbermejo@iMac-de-M ~ % curl -v google.com
* Trying 142.250.184.174...
* TCP_NODELAY set
* Connected to google.com (142.250.184.174) port 80 (#0)
> GET / HTTP/1.1
> Host: google.com
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Location: http://www.google.com/
< Content-Type: text/html; charset=UTF-8
< Date: Sun, 24 Jan 2021 09:48:00 GMT
< Expires: Tue, 23 Feb 2021 09:48:00 GMT
< Cache-Control: public, max-age=2592000
< Server: gws
< Content-Length: 219
< 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.com/">here</A>.
</BODY></HTML>
* Connection #0 to host google.com left intact
* Closing connection 0
pilarbermejo@iMac-de-M ~ %
```

Conectado a Google
Opción GET
Host Google.com
Mediante Curl
Petición Aceptada

Cabecera

WGNET:

1.1 ¿Qué hace?

Podemos usarlo para **recuperar contenido** y archivos de varios servidores web.

1.2 ¿Cómo se usa?

Uno de los ejemplos más básicos de este comando es **descargar una página completa** y almacenarla en nuestro directorio de trabajo actual.

Antes de eso hemos tenido que instalarlo mediante el comando `$ brew install wget`

```
$ wget www.nombrededominio.com
```

`$ wget -p www.nombrededominio.com` (descargamos una página completa con hojas de estilo imágenes de línea)

`$ wget -r www.nombrededominio.com` (se descarga recursivamente hasta 5 niveles del sitio)

1.3 ¿Porqué responde a la pregunta?

Porque nos muestra cómo podemos acceder a servidores a **descargar la información que necesitamos**. Aunque algunos servidores pueden haber bloqueado el acceso a wget.

1.4 ¿Cómo se interpreta la salida?

```
pilarbermejo@iMac-de-M ~ % wget www.globosrosas.com Comando wget para descargar www.globosrosas.com
--2021-01-24 11:35:43-- http://www.globosrosas.com/
Resolviendo www.globosrosas.com (www.globosrosas.com)... 37.152.88.46
Conectando con www.globosrosas.com (www.globosrosas.com)[37.152.88.46]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
Grabando a: «index.html» Establece la conexión y descarga archivo index.html

index.html [ <=> ] 78,93K --.-KB/s en 0,03s
2021-01-24 11:35:45 (2,95 MB/s) - «index.html» guardado [80824] Archivo guardado en el directorio actual

pilarbermejo@iMac-de-M ~ %
```

index.html hoy 11:35

5. ¿QUÉ OTROS COMANDOS TE HAN HECHO FALTA?

Los comandos que me han ido haciendo falta los he ido indicando a lo largo de la actividad.

Únicamente he utilizado el comando *clear* para limpiar la pantalla y realizar las capturas.