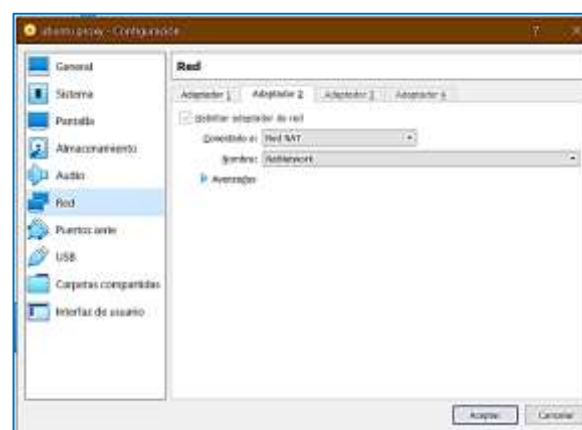
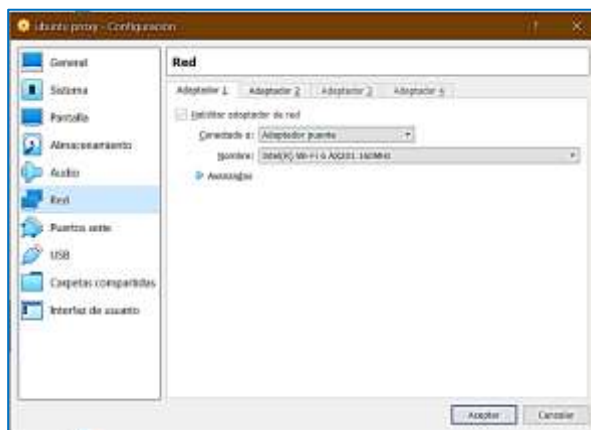


## Anexo. Instalar servidor proxy

Para el instalar el servidor proxy **Squid** utilizaremos una máquina virtual en VirtualBox con Linux Ubuntu y dos adaptadores de red:

- Adaptador puente, para conectar la máquina con la red externa
- Red Nat, para configurar una red interna



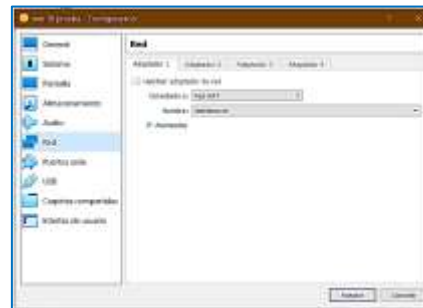
La máquina tendrá 2 direcciones IP, una para cada adaptador:

```
pru@pru-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.43 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::adfc:f033:9000:d819 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:78:27:6e txqueuelen 1000 (Ethernet)
    RX packets 613 bytes 280447 (280.4 KB)
    RX errors 0 dropped 11 overruns 0 frame 0
    TX packets 225 bytes 24278 (24.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.5 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::438e:3684:ebd4:6165 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:8b:a1:c8 txqueuelen 1000 (Ethernet)
    RX packets 53 bytes 15132 (15.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 113 bytes 13693 (13.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 200 bytes 17772 (17.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 200 bytes 17772 (17.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Usaremos como cliente una máquina virtual en VirtualBox con Windows 10 y 1 adaptador de red en la red NAT



La máquina tendrá 1 direcciones IP dentro del rango de la red interna:

```
C:\Users\pru>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . . . : fe80::8006:3903:33b0:f30e%14
    Dirección IPv4. . . . . : 10.0.3.4
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 10.0.3.1
```

## Instalación de Squid en Debian/Ubuntu Linux

La instalación del proxy **Squid** en Ubuntu y otras distribuciones de Debian Linux es un proceso sencillo. Puede instalar el proxy **Squid** desde el repositorio oficial de Linux utilizando el administrador de paquetes aptitude.

Ejecute las siguientes líneas de comandos que se indican a continuación en el shell de su terminal con privilegios de root para instalar el proxy **Squid** en su sistema.

```
sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install squid
```

```
pru@pru-VirtualBox:~$ sudo apt-get install squid  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.  
libflashrom1 libftdi1-2 liblvm1  
Utilice «sudo apt autoremove» para eliminarlos.  
Se instalarán los siguientes paquetes adicionales:  
libdbi-perl libecap3 squid-common squid-langpack  
Paquetes sugeridos:  
libnbd-perl libnet-daemon-perl libsql-statement-perl squidclient squid-cgi squid-purge resolvconf snbclient winbind  
Se instalarán los siguientes paquetes NUEVOS:  
libdbi-perl libecap3 squid squid-common squid-langpack  
o actualizados, 5 nuevos se instalarán, 0 para eliminar y 10 no actualizados.  
Se necesita descargar 3.913 kB de archivos.  
Se utilizarán 15,2 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s
```

Cuando finalice la instalación, se puede comprobar que el servicio está funcionando:

```
$ sudo netstat -apn | grep squid
```

```
pru@pru-VirtualBox:~$ sudo netstat -apn | grep squid  
tcp6      0      0  :::3128          :::*              ESCUCHAR    5128/(squid-1)  
udp       0      0  0.0.0.0:38106    0.0.0.0:*         5128/(squid-1)  
udp6      0      0  :::48597        :::*              5128/(squid-1)  
udp6      0      0  :::1:42478      :::1:39362        ESTABLECIDO 5128/(squid-1)  
unix  2      [ ]          DGRAM    CONECTADO    39482    5126/squid  
unix  2      [ ]          DGRAM    CONECTADO    40016    5128/(squid-1)  
unix  3      [ ]          FLUJO    CONECTADO    40023    5128/(squid-1)
```

En la máquina cliente, configuramos el proxy:

Configuración

Inicio

Buscar una configuración

Red e Internet

Estado

Ethernet

Acceso telefónico

VPN

Modo avión

Proxy

### Proxy

Detectar la configuración automáticamente

☐ Desactivado

Usar script de configuración

☐ Desactivado

Dirección de script

Guardar

### Configuración manual del proxy

Usa un servidor proxy para conexiones Ethernet o Wi-Fi. Esta configuración no se aplica a conexiones VPN.

Usar servidor proxy

☐ Desactivado

Dirección

10.0.3.5

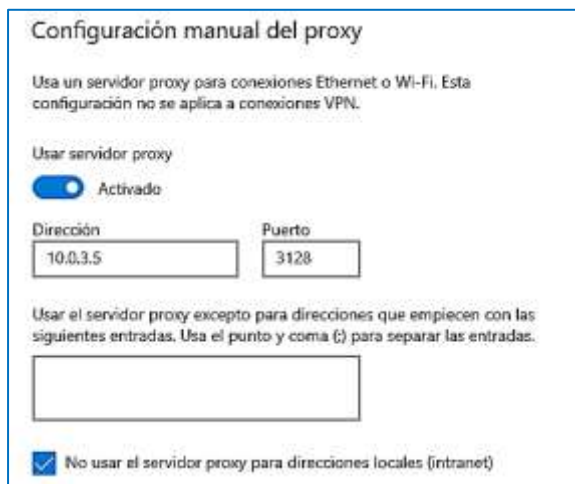
Puerto

3128

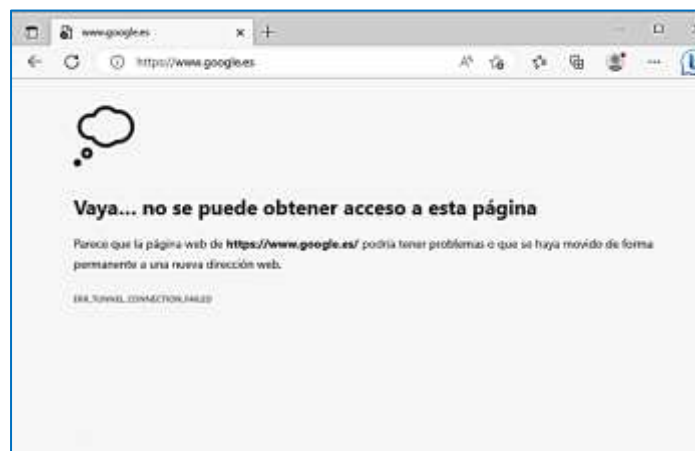
Usar el servidor proxy excepto para direcciones que empiecen con las siguientes entradas. Usa el punto y coma (;) para separar las entradas.

☒ No usar el servidor proxy para direcciones locales (intranet)

Lo haremos de forma manual, indicando la dirección IP del servidor proxy **Squid** y el puerto (por defecto es 3128):



Intentamos acceder a una URL externa y vemos que no hay conexión:



Para que el cliente pueda conectarse externamente, hay que configurar el servidor proxy **Squid**

## Configuración de Squid Proxy en Linux

Veremos cómo puede configurar y comenzar con el proxy **Squid** en su sistema Linux.

### 1. Comprobación del estado de Squid

Una vez finalizada la instalación del proxy **Squid**, debe verificar el estado para saber si está funcionando en su sistema o no. A veces, la falta de coincidencia con los archivos de configuración podría ser la causa de que el servidor proxy no funcione. Le recomendaría que copie la configuración predeterminada en un bloc de notas para corregirla si algo sale mal.

Sin embargo, ejecute el siguiente comando de control del sistema en el shell de su terminal para verificar el estado del proxy **Squid** en su sistema Linux. Vería el PID, las tareas, el uso de la memoria y otras piezas de información en el shell.

```
sudo systemctl status squid
```

```
pru@pru-VirtualBox: $ sudo systemctl status squid
● squid.service - Squid Web Proxy Server
   Loaded: loaded (/lib/systemd/system/squid.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-31 04:00:59 WEST; 7min ago
     Docs: man:squid(8)
  Process: 5123 ExecStartPre=/usr/sbin/squid --foreground -z (code=exited, status=0/SUCCESS)
    Main PID: 5126 (squid)
      Tasks: 4 (limit: 4614)
     Memory: 16.2M
        CPU: 178ms
    CGroup: /system.slice/squid.service
            └─5126 /usr/sbin/squid --foreground -sYC
              └─5128 "(squid-1)" --kid squid-1 --foreground -sYC
                └─5129 "(logfile-daemon)" /var/log/squid/access.log
                  └─5130 "(pinger)"

may 31 04:00:59 pru-VirtualBox squid[5128]: Using Least Load store dir selection
may 31 04:00:59 pru-VirtualBox squid[5128]: Set Current Directory to /var/spool/squid
may 31 04:00:59 pru-VirtualBox squid[5128]: Finished loading MIME types and icons.
may 31 04:00:59 pru-VirtualBox squid[5128]: HTCP Disabled.
may 31 04:00:59 pru-VirtualBox squid[5128]: Pinger socket opened on FD 14
may 31 04:00:59 pru-VirtualBox systemd[1]: Started Squid Web Proxy Server.
may 31 04:00:59 pru-VirtualBox squid[5128]: Squid plugin modules loaded: 0
may 31 04:00:59 pru-VirtualBox squid[5128]: Adaptation support is off.
may 31 04:00:59 pru-VirtualBox squid[5128]: Accepting HTTP Socket connections at conn3 local=[
may 31 04:01:00 pru-VirtualBox squid[5128]: storeLateRelease: released 0 objects
```



## 2. Configurar la red en Squid Proxy

La configuración de la red del proxy **Squid** le permitirá conectar su servidor proxy a otras máquinas. Los archivos de configuración se almacenan dentro del directorio `etc` y el `var` de un sistema de archivos Linux. Aquí, le voy a dar una nota donde puede encontrar el archivo de configuración en su sistema.

Encuentre todos los ajustes de configuración dentro del directorio `etc`.

```
/etc/squid/squid.conf
```

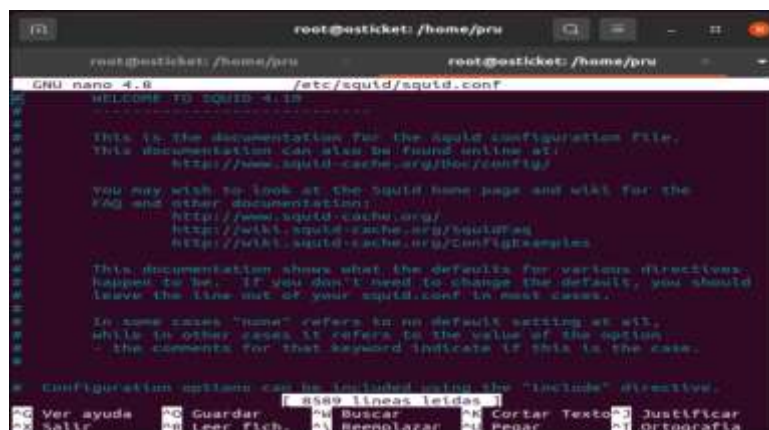
Encuentre la información de acceso y registro en el directorio `var`.

```
/var/log/squid/access.log
```

```
/var/log/squid/cache.log
```

El fichero de configuración de **Squid**, que se encuentra en `/etc/squid/`:

```
sudo nano /etc/squid/squid.conf
```



```
root@osticket: /home/pru
root@osticket: /home/pru
GNU nano 4.0 /etc/squid/squid.conf
#
# WELCOME TO SQUID 4.10
#
# This is the documentation for the Squid configuration file.
# This documentation can also be found online at:
#   http://www.squid-cache.org/Doc/config/
#
# You may wish to look at the Squid home page and wiki for the
# FAQ and other documentation:
#   http://www.squid-cache.org/
#   http://wiki.squid-cache.org/SquidWiki
#   http://wiki.squid-cache.org/ConfigExamples
#
# This documentation shows what the defaults for various directives
# happen to be. If you don't need to change the default, you should
# leave the line out of your squid.conf in most cases.
#
# In some cases "none" refers to no default setting at all,
# while in other cases it refers to the value of the option
# - the comments for that keyword indicate if this is the case.
#
# Configuration options can be included using the "include" directive.
#
# 8589 líneas leídas
# Ver ayuda  Guardar  Buscar  Cortar Texto  Justificar
# Salir  Leer fich.  Reemplazar  Pegar  Ortografía
```



El fichero está lleno de comentarios, ya que aparece todo el manual comentado. Podemos eliminar las líneas comentadas. Para ello entramos en Vim y realizamos lo siguiente:

```
:g/^\s*#/d
:g/^\$/d
:wq
```

Y el fichero queda:

```
acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
acl localnet src 10.0.0.0/8 # RFC 1918 local private network (LAN)
acl localnet src 100.64.0.0/10 # RFC 6598 shared address space (CGN)
acl localnet src 169.254.0.0/16 # RFC 3927 link-local (directly plugged) machines
acl localnet src 172.16.0.0/12 # RFC 1918 local private network (LAN)
acl localnet src 192.168.0.0/16 # RFC 1918 local private network (LAN)
acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
include /etc/squid/conf.d/*.conf
http_access allow localhost
http_access deny all
http_port 3128
coredump_dir /var/spool/squid
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern \/(Packages|Sources)(|\.bz2|\.gz|\.xz)$ 0 0% 0 refresh-ims
refresh_pattern \/(Release(|\.gpg)$ 0 0% 0 refresh-ims
refresh_pattern \/(InRelease$ 0 0% 0 refresh-ims
refresh_pattern \/(Translation-.*)(|\.bz2|\.gz|\.xz)$ 0 0% 0 refresh-ims
refresh_pattern . 0 20% 4320
```

Destacamos que el fichero de configuración está compuesto básicamente de 2 elementos:

- Elementos ACL. Son definiciones
- Listas de acceso. Indica acciones a realizar

Vamos a crear un fichero de reglas, en el que vamos a incluir las reglas que vamos a definir. Actualmente, el proxy está configurado para no aceptar ninguna conexión. El fichero de configuración **squid.conf** indica que va a incluir las reglas que se encuentren en el directorio /etc/squid/conf.d. Por tanto, Vamos a crear un fichero de reglas:

```
sudo nano /etc/squid/conf.d/myrules.conf
```

En primer lugar, añadimos una regla que permita la navegación de los equipos locales:

```
#Crear ACL para nuestra red local  
acl miredlocal src 10.0.3.0/24  
#Permitir navegación web para ACL miredlocal  
http_access allow miredlocal
```

```
#Crear ACL para nuestra red local  
acl miredlocal src 10.0.3.0/24  
  
#Permitir navegación web para ACL miredlocal  
http_access allow miredlocal
```

Para que funcione, reiniciamos squid:

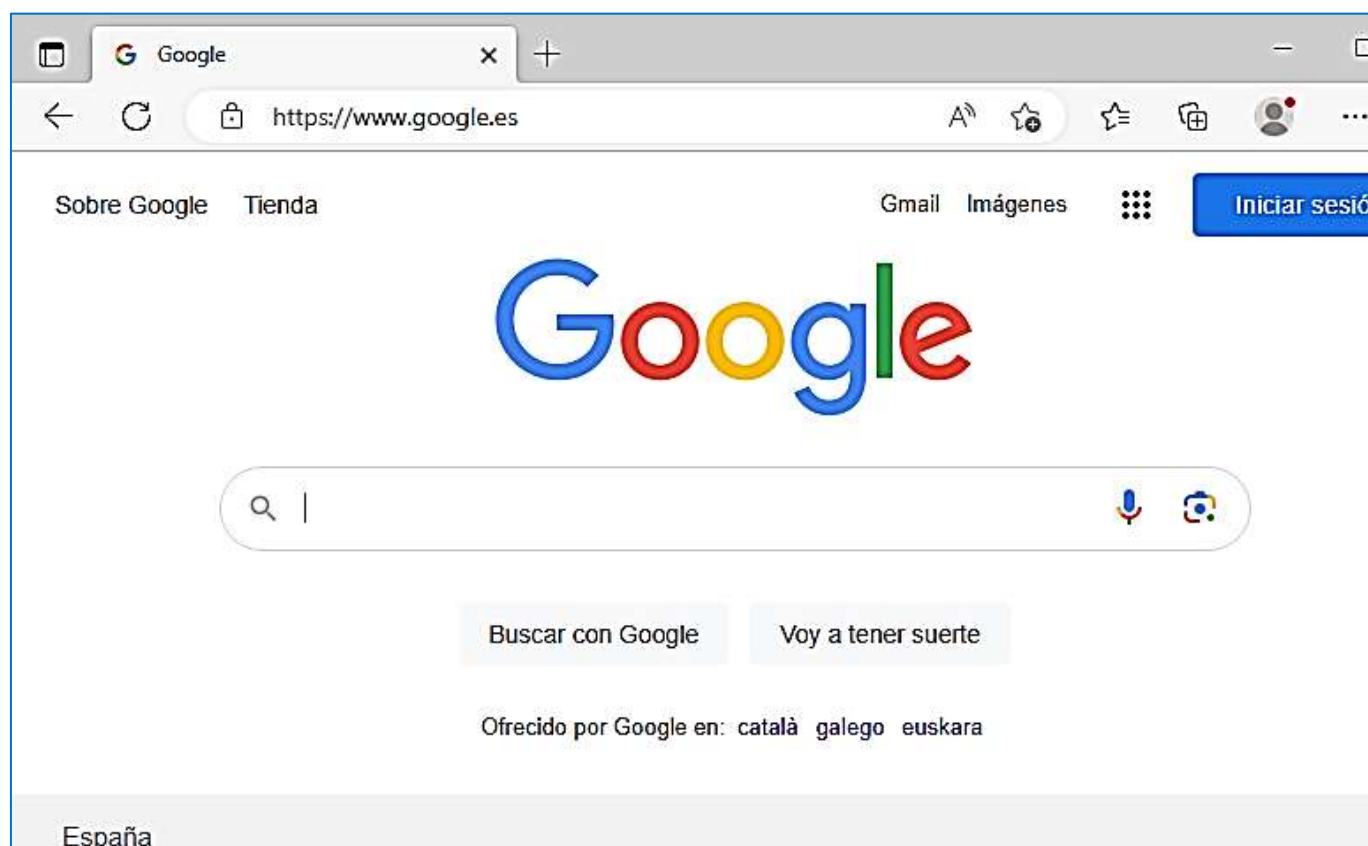
```
sudo service squid restart
```

**Nota.** podemos utilizar también:

```
sudo squid -k reconfigure
```

En caso de error, nos indica en qué lugar se encuentra.

Ahora, comprobamos en el equipo Windows que hay conexión a Internet:





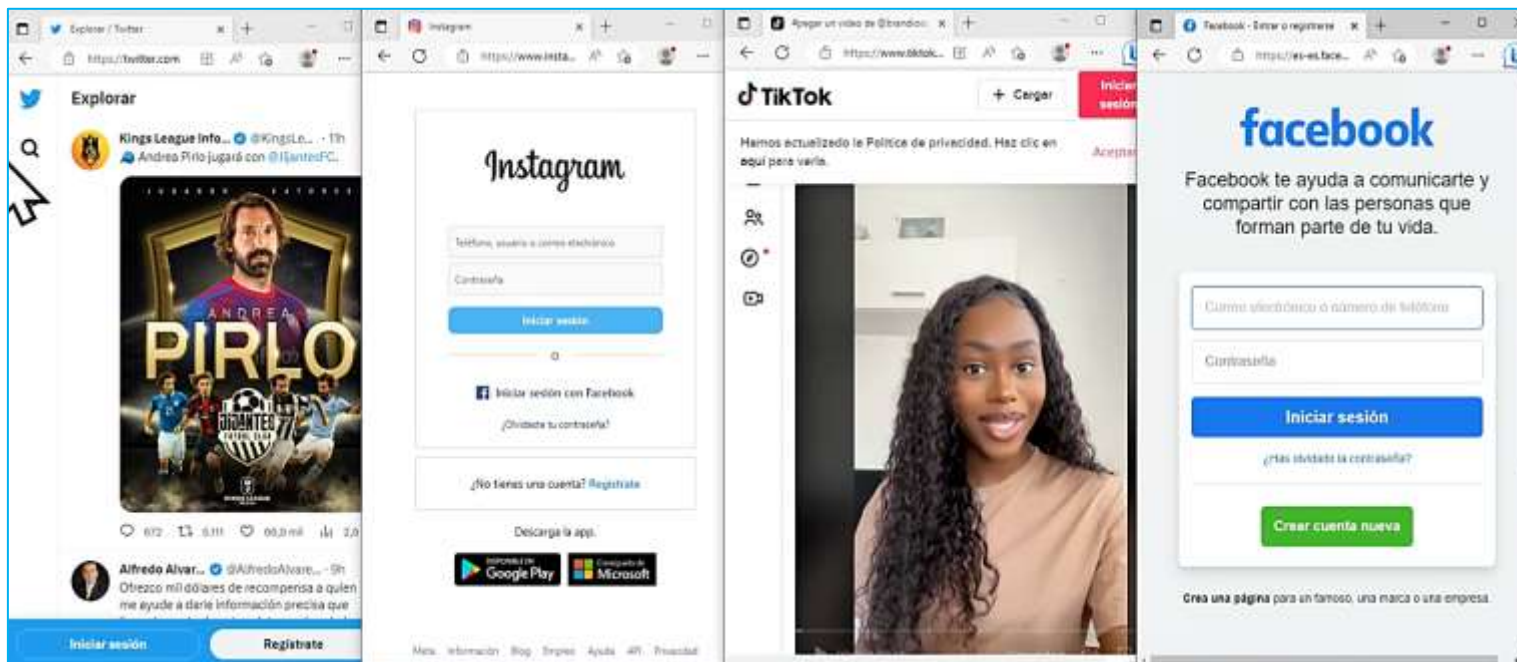
Podemos ver el fichero log:

```
tail -f /var/log/squid/access.log
```

```
pru@pru-VirtualBox:~$ sudo tail -f /var/log/squid/access.log
1685505755.527 122 10.0.3.4 TCP_MISS/304 433 GET http://ctldl.windowsupdate.
com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab? - HIER_DIRECT/93.184.221.240 -
1685505755.579 37 10.0.3.4 TCP_MISS/304 434 GET http://ctldl.windowsupdate.
com/msdownload/update/v3/static/trustedr/en/authrootstl.cab? - HIER_DIRECT/93.184.221.240 -
1685505755.627 37 10.0.3.4 TCP_MISS/304 434 GET http://ctldl.windowsupdate.
com/msdownload/update/v3/static/trustedr/en/pinrulesstl.cab? - HIER_DIRECT/93.184.221.240 -
1685505763.153 125092 10.0.3.4 TCP_TUNNEL/200 7486 CONNECT www.bing.com:443 - HIER_DIRECT/204.79.197.200 -
1685505763.294 125482 10.0.3.4 TCP_TUNNEL/200 8823 CONNECT www.bing.com:443 - HIER_DIRECT/204.79.197.200 -
1685505763.856 124634 10.0.3.4 TCP_TUNNEL/200 578 CONNECT edge.microsoft.com:443 - HIER_DIRECT/13.107.21.239 -
1685505879.409 240225 10.0.3.4 TCP_TUNNEL/200 70751 CONNECT www.gstatic.com:443 - HIER_DIRECT/142.250.185.3 -
1685505879.813 241156 10.0.3.4 TCP_TUNNEL/200 539915 CONNECT www.google.es:443 - HIER_DIRECT/142.250.184.3 -
1685505879.982 240234 10.0.3.4 TCP_TUNNEL/200 1950 CONNECT adservice.google.es:443 - HIER_DIRECT/216.58.215.130 -
1685505883.892 245185 10.0.3.4 TCP_TUNNEL/200 5617 CONNECT fonts.gstatic.com:443 - HIER_DIRECT/142.250.200.131 -
```

### 3. Bloquear sitios web a través de Squid

Veamos, en primer lugar, que si tenemos acceso a diversas redes sociales:



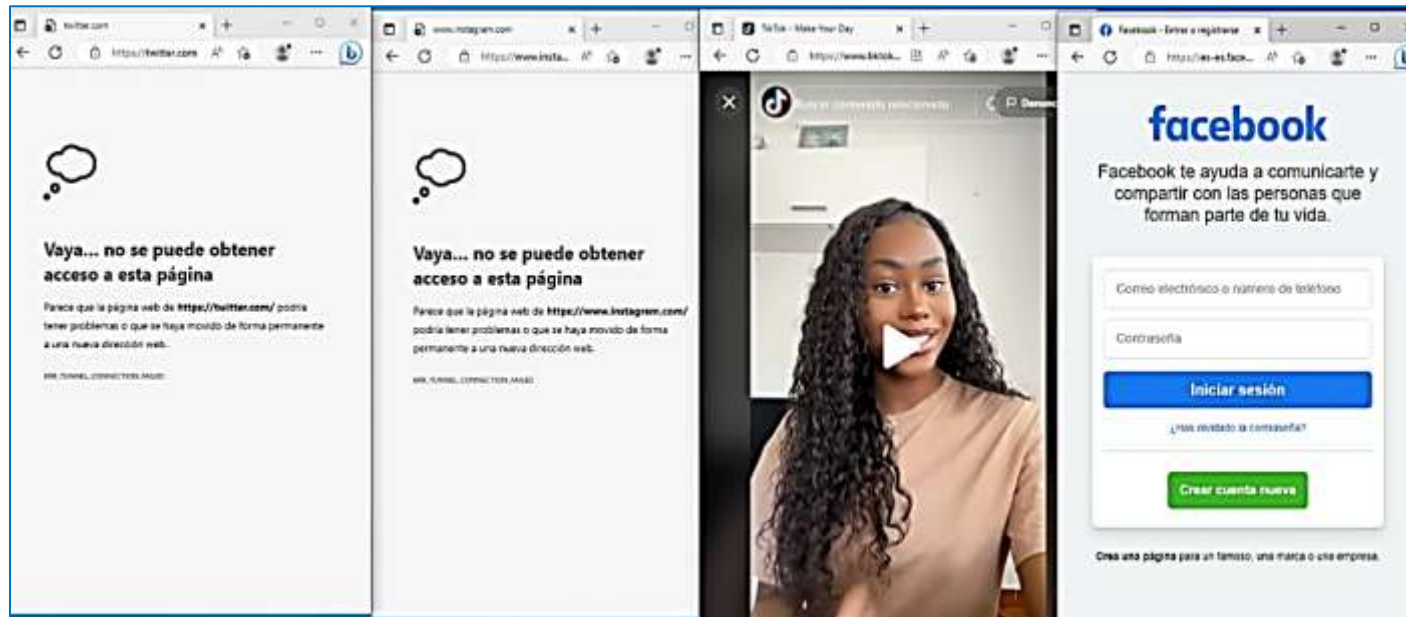
Si desea bloquear algunos sitios web en su servidor proxy, puede hacerlo agregando el script dentro del fichero de reglas que hemos creado (myrules.conf). Vamos a añadir un ACL indicando los dominios que vamos a bloquear y luego una regla denegándolo:

```
#Crear ACL indicando dominios (redes sociales)
acl filtro_rrss dstdomain .twitter.com .x.com .instagram.com

#Denegar el acceso a las Redes sociales indicadas
http_access deny filtro_rrss
```

```
#Crear ACL para nuestra red local
acl miredlocal src 10.0.2.0/24
#Crear ACL indicando dominios (Redes sociales)
acl filtro_rrss dstdomain .twitter.com .x.com .instagram.com
#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss
#Permitir navegación web para ACL miredlocal
http_access allow miredlocal
```

Comprobamos que no tenemos acceso a las redes sociales indicadas, pero si a otras:



Observemos que el orden de las reglas es muy importante. Hemos puesto las reglas de bloqueo de acceso antes de la regla que permite el acceso a todo.

Otra forma de indicar los sitios bloqueados es creando un fichero en el cual se incluyan los dominios bloqueados. Para ello, en el fichero de reglas, le indicamos que los dominios bloqueados se encuentran en un fichero (dominios-denegados):

```
#Crear ACL indicando dominios en un fichero
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"
```

```
#Crear ACL para nuestra red local
acl miredlocal src 10.0.3.0/24

#Crear ACL indicando dominios en un fichero
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"

#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss

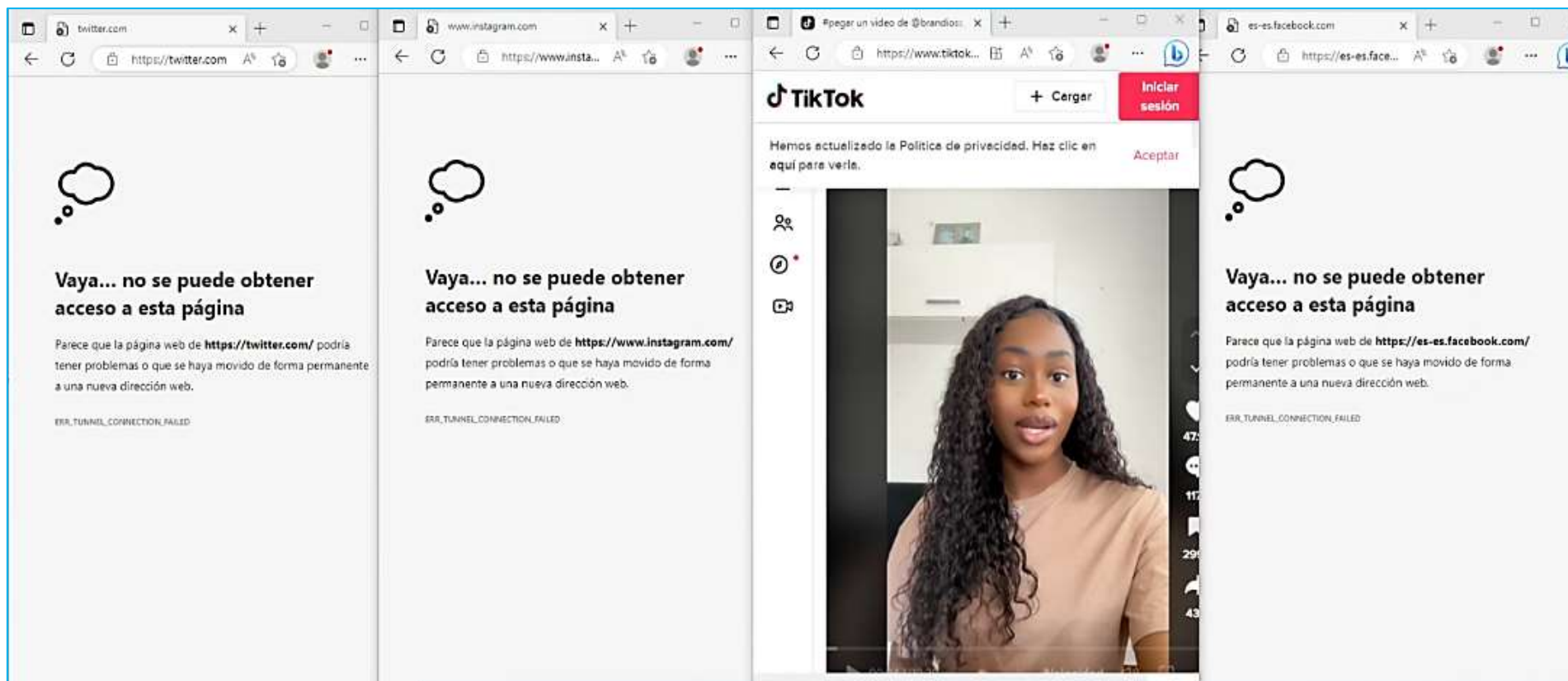
#Permitir navegación web para ACL miredlocal
http_access allow miredlocal
```

```
sudo nano /etc/squid/dominios-denegados
```

```
.facebook.com
.twitter.com
.x.com
.instagram.com|
```



Reiniciamos **Squid** y podemos ver que ahora está bloqueado el acceso a las redes sociales indicadas:



## 4. Bloquear con expresiones regulares

Por ejemplo, vamos a indicarle que bloquee el acceso a cualquier página que contenga dentro de su URL una expresión determinada. Para ello, creamos en el fichero myrules.conf una nueva lista, dirigida al archivo /etc/squid/bloqu-exp que contendrá la expresión regular:

```
#Crear ACL con expresiones regulares a prohibir
acl block-exp url_regex "/etc/squid/block-exp"
#Denegar navegación al elemento block-exp
http_access deny block-exp
```

```
#Crear ACL para nuestra red local
acl miredlocal src 10.0.3.0/24

#Crear ACL indicando dominios en un fichero
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"

#Crear ACL con expresiones regulares a prohibir
acl block-exp url_regex "/etc/squid/block-exp"

#Denegar navegación al elemento block-exp
http_access deny block-exp

#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss

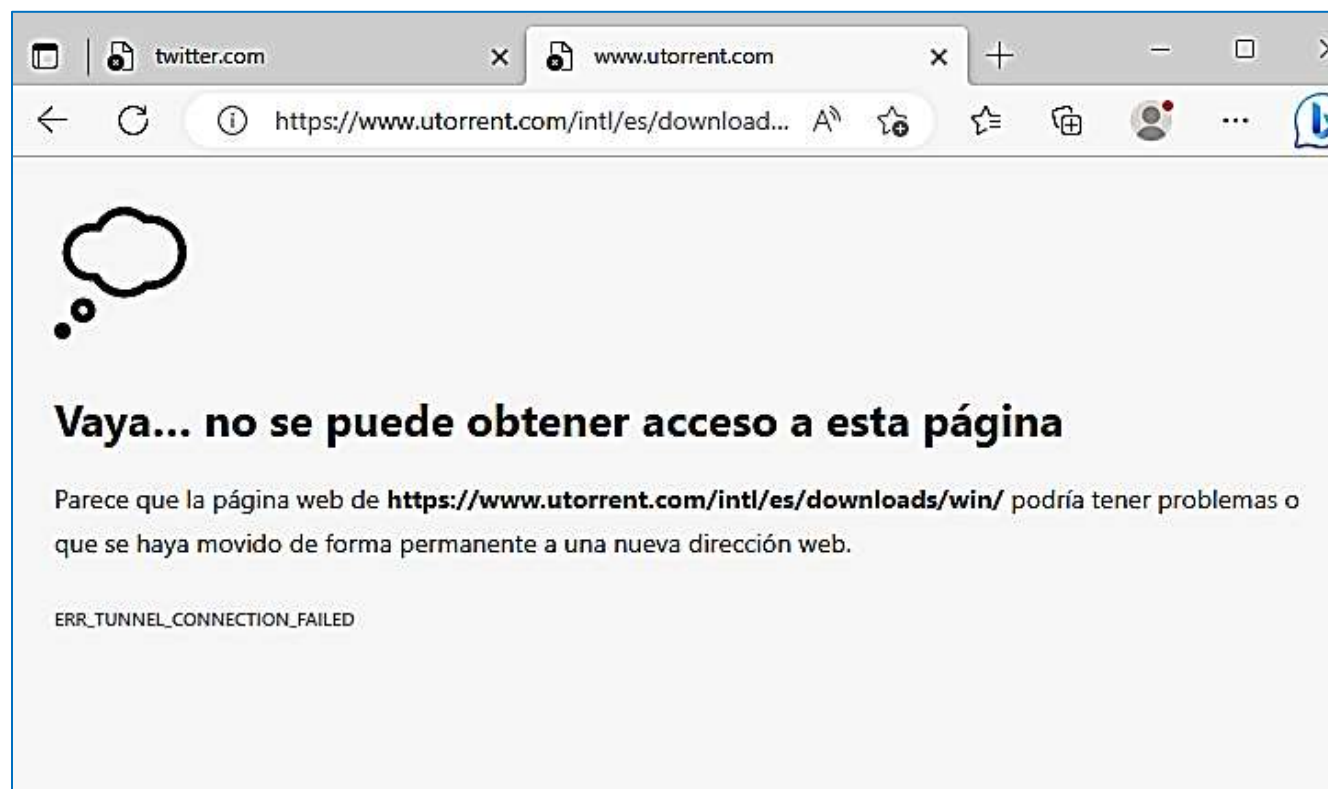
#Permitir navegación web para ACL miredlocal
http_access allow miredlocal
```

Creamos el fichero /etc/squid/bloq-exp:

```
pru@pru-VirtualBox:~$ sudo nano /etc/squid/block-exp
```

```
GNU nano 6.2
torrent
```

Con lo que le indicamos que excluya los sitios que contengan en su URL la palabra torrent. Reiniciamos el servidor y comprobamos:



## 5. Bloquear en fechas y horas determinadas

Veamos, ahora como **bloquear en fechas y horas determinadas**. Por ejemplo, para indicarle que pueda acceder en un horario determinado (de lunes a viernes de 08:00 a 17:30 h). Para ello, creamos en el fichero `myrules.conf` una nueva acl, que indicará el horario en el que está permitido el acceso:

```
#Crear ACL indicando días y horario de trabajo
acl HORARIO time MTWHF 08:00-17:30
```

```
#Denegar navegación fuera del horario
http_access deny ;HORARIO
```

```
#Crear ACL para nuestra red local
acl miredlocal src 10.0.3.0/24

#Crear ACL indicando dominios en un fichero
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"

#Crear ACL con expresiones regulares a prohibir
acl block-exp url_regex "/etc/squid/block-exp"

#Crear ACL indicando días y horario de trabajo
acl HORARIO time MTWHF 08:00-17:30

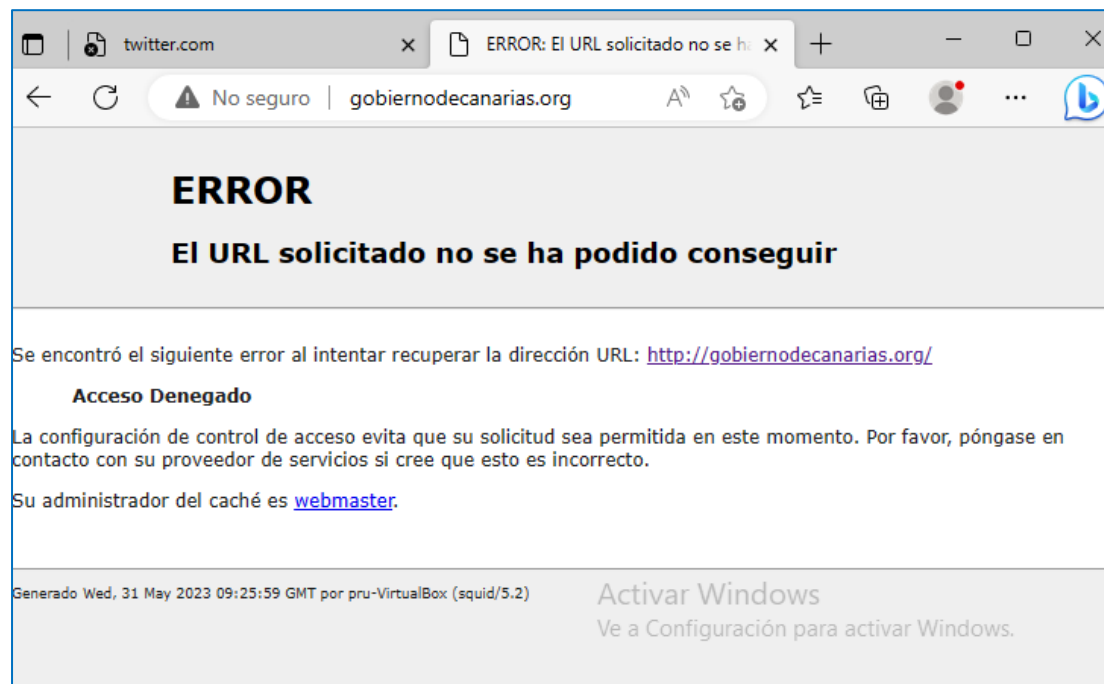
#Denegar navegación fuera del horario
http_access deny HORARIO

#Denegar navegación al elemento block-exp
http_access deny block-exp

#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss

#Permitir navegación web para ACL miredlocal
http_access allow miredlocal
```

Reiniciamos el servidor y comprobamos:





## 6. Configurar el proxy para identificarse con usuario y contraseña

Para permitir a un conjunto de usuarios el acceso, hay varias formas de hacerlo. Vamos a hacerlo usando las herramientas **Apache Utils**:

```
sudo apt-get install apache2-utils
```

```
pru@pru-VirtualBox:~$ sudo apt-get install apache2-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios
  libflashrom1 libftdi1-2 libllvm13
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libapr1 libaprutil1
Se instalarán los siguientes paquetes NUEVOS:
  apache2-utils libapr1 libaprutil1
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 10 no actualizados.
Se necesita descargar 290 kB de archivos.
Se utilizarán 992 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

A continuación, vamos a crear el fichero que va a contener los usuarios y sus claves (hash):

```
sudo htpasswd -c /etc/squid/squid_password pru
```

```
pru@pru-VirtualBox:~$ sudo htpasswd -c /etc/squid/squid_password pru
New password:
Re-type new password:
Adding password for user pru
```

Añadimos varios usuarios:

```
sudo htpasswd /etc/squid/squid_password usuario1
sudo htpasswd /etc/squid/squid_password usuario2
sudo htpasswd /etc/squid/squid_password usuario3
```

```
pru@pru-VirtualBox:~$ sudo htpasswd /etc/squid/squid_password usuario1
New password:
Re-type new password:
Adding password for user usuario1
pru@pru-VirtualBox:~$ sudo htpasswd /etc/squid/squid_password usuario2
New password:
Re-type new password:
Adding password for user usuario2
pru@pru-VirtualBox:~$ sudo htpasswd /etc/squid/squid_password usuario3
New password:
Re-type new password:
Adding password for user usuario3
```

Podemos observar que se han guardado los usuarios con el hash de las contraseñas:

```
pru@pru-VirtualBox:~$ cat /etc/squid/squid_password
pru:$apr1$JcTBjej8$Xd1VeFkSwcsDw7hY8ri920
usuario1:$apr1$aPPsN2K3$ri2Ru0WN2rLokoEi2WmWj.
usuario2:$apr1$q.hVU5zm$B4C4V/P7PTJVmdnHQRNum1
usuario3:$apr1$pPFy3R6I$9CBjS1T6MBFv5ZJWUJ0la/
```



Para asegurarnos de su funcionamiento, vamos a comprobar donde está el fichero `basic_ncsa_auth`:

```
sudo find / basic_ncsa_auth | grep basic_ncsa_auth
```

```
pru@pru-VirtualBox:~$ sudo find / basi_ncsa_auth |grep basic_ncsa_auth
find: '/run/user/1000/doc': Permiso denegado
find: '/run/user/1000/gvfs': Permiso denegado
/usr/lib/squid/basic_ncsa_auth
/usr/share/man/man8/basic_ncsa_auth.8.gz
find: 'basi_ncsa_auth': No existe el archivo o el directorio
```

Comprobamos que el fichero `basic_ncsa_auth` se encuentra en `/usr/lib/squid/`.

Podemos comprobar las contraseñas ejecutando `basic_ncsa_auth` de la siguiente forma:

```
/usr/lib/squid/basic_ncsa_auth /etc/squid/squid_password
```

Escribimos un usuario y su contraseña separada por espacio en blanco:

```
pru@pru-VirtualBox:~$ /usr/lib/squid/basic_ncsa_auth /etc/squid/squid_password
pru pru
OK
usuario1 usuario1
OK
```

Ahora, hay que configurar **Squid** para que requiera usuario y contraseña:

```
sudo nano /etc/squid/squid.conf
```

y añadimos, al principio del fichero, los siguientes parámetros:

```
#Indicar la ubicación del archivo basic_ncsa_auth y del fichero con los usuarios
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/squid_password

#Indicamos el mensaje para el cliente
auth_param basic realm Autenticación de usuarios de Squid

#Nº máximo de login
auth_param basic children 5

#Duración máxima del login
auth_param basic credentialsttl 2 hours
```

```
#Indicar la ubicación del archivo basic_ncsa_auth y del fichero con los usuarios
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/squid_password
#Indicamos el mensaje para el cliente
auth_param basic realm Autenticación de usuarios de Squid
#Nº máximo de login
auth_param basic children 5
#Duración máxima del login
auth_param basic credentialsttl 2 hours

acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
```

Ahora, tenemos que añadir en el fichero de reglas **myrules.conf** la indicación de que se requiere autenticación de usuarios:

```
sudo nano /etc/squid/conf.d/myrules.conf
```

y añadimos, al principio, los siguientes parámetros:

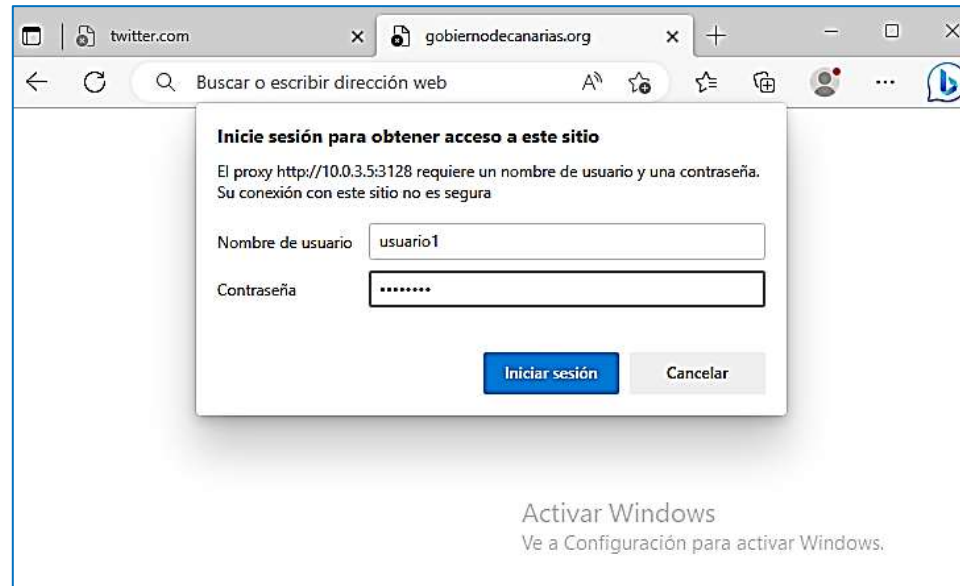
```
#Creación de ACL para requerir autenticación de usuario
acl usuarios proxy_auth REQUIRED

#Permitir solo el acceso http para usuarios autenticados
http_access allow usuarios
```

```
#Creación de ACL para requerir autenticación de usuario
acl usuarios proxy_auth REQUIRED
#Crear ACL para nuestra red local
acl miredlocal src 10.0.2.0/24
#Crear ACL indicando dominios (Redes sociales)
#acl filtro_rrss dstdomain .twitter.com .x.com .instagram.com
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"
#Crear ACL con expresiones regulares a prohibir
acl block-exp url_regex "/etc/squid/block-exp"
#Crear ACL indicando días y horario de trabajo
acl HORARIO time MTWHF 05:00-17:30

#Permitir solo el acceso http para usuarios autenticados
http_access allow usuarios
#Denegar navegación fuera del horario
http_access deny !HORARIO
#Denegar navegación al elemento block-exp
http_access deny block-exp
#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss
```

Reiniciamos el servidor y comprobamos:



## 7. Listas públicas de bloqueo de dominios

Existen en Internet listas de acceso para bloquear dominios no seguros. Veamos una de estas.

### Blackweb:

**Blackweb** es un proyecto que recopila y unifica listas públicas de bloqueo de dominios (porno, descargas, drogas, malware, spyware, trackers, bots, redes sociales, warez, venta de armas, etc) para hacerlas compatibles con **Squid**.

Para ello, vamos a descargarlo de **GitHub**:

```
cd /etc/squid/  
sudo git clone https://github.com/maravento/blackweb.git  
cd blackweb  
ls  
sudo tar -xzf blackweb.tar.gz  
ls
```

Ahora, añadimos la lista al fichero **myrules.conf**

```
#Creación de ACL indicando ubicación de fichero blackweb  
acl blackweb dstdomain "/etc/squid/blackweb/blackweb.txt"  
  
#Denegar el acceso a los dominios blackweb  
http_access deny blackweb
```

```
#Creación de ACL para requerir autenticación de usuario
acl usuarios proxy_auth REQUIRED
#Crear ACL para nuestra red local
acl miredlocal src 10.0.2.0/24
#Creación de ACL indicando ubicación de fichero blackweb
acl blackweb dstdomain "/etc/squid/blackweb/blackweb.txt"
#Crear ACL indicando dominios (Redes sociales)
#acl filtro_rrss dstdomain .twitter.com .x.com .instagram.com
acl filtro_rrss dstdomain "/etc/squid/dominios-denegados"
#Crear ACL con expresiones regulares a prohibir
acl block-exp url_regex "/etc/squid/block-exp"
#Crear ACL indicando días y horario de trabajo
acl HORARIO time MTWHF 05:00-17:30

#Permitir solo el acceso http para usuarios autenticados
http_access allow usuarios
#Denegar el acceso a los dominios blackweb
http_access deny blackweb
#Denegar navegación fuera del horario
http_access deny !HORARIO
#Denegar navegación al elemento block-exp
http_access deny block-exp
#Denegar el acceso a las redes sociales indicadas
http_access deny filtro_rrss
#Permitir navegación web para ACL miredlocal
http_access allow miredlocal
```



Ahora comprobamos que no podemos acceder a sitios prohibidos y acceder a sitios permitidos:

