

Guía de Configuración Completa de Servidor Proxy **Squid**

9/07/2025

Ángel Moreno García
CodeArts Solutions
Madrid

Introducción

Este documento detalla la instalación, configuración y puesta en marcha de un servidor proxy Squid en un entorno Linux, con configuración avanzada para control de acceso por grupos, autenticación básica y políticas de uso. También se incluyen pruebas de funcionamiento y resolución de problemas típicos.

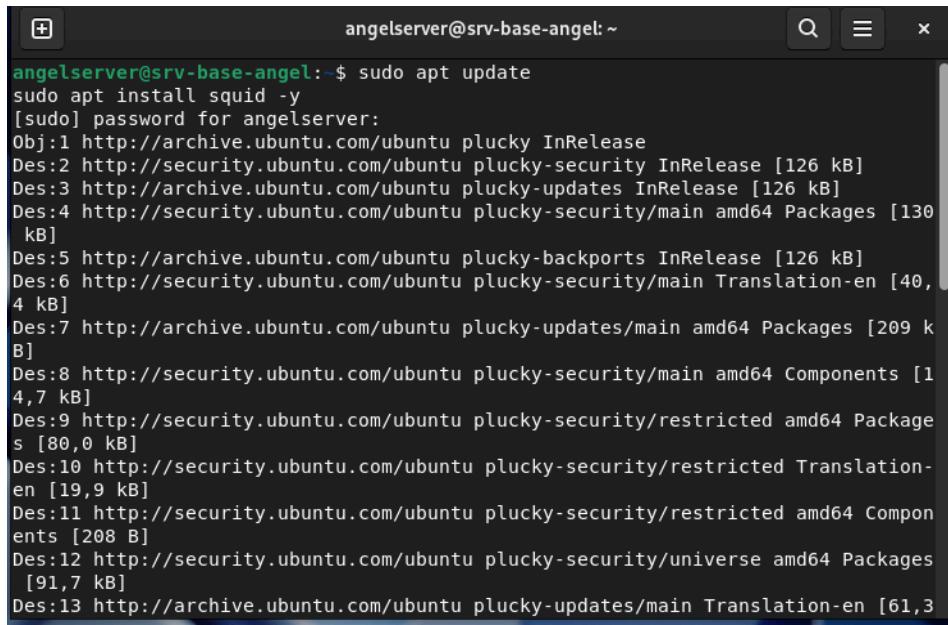
Instalación de Squid

Instalación del paquete

Empezaremos la instalación del paquete Squid introduciendo en la consola el comando:

sudo apt update

sudo apt install squid -y



```
angelserv...@...:~$ sudo apt update
[sudo] password for angelserv...
Obj:1 http://archive.ubuntu.com/ubuntu plucky InRelease
Des:2 http://security.ubuntu.com/ubuntu plucky-security InRelease [126 kB]
Des:3 http://archive.ubuntu.com/ubuntu plucky-updates InRelease [126 kB]
Des:4 http://security.ubuntu.com/ubuntu plucky-security/main amd64 Packages [130 kB]
Des:5 http://archive.ubuntu.com/ubuntu plucky-backports InRelease [126 kB]
Des:6 http://security.ubuntu.com/ubuntu plucky-security/main Translation-en [40,4 kB]
Des:7 http://archive.ubuntu.com/ubuntu plucky-updates/main amd64 Packages [209 kB]
Des:8 http://security.ubuntu.com/ubuntu plucky-security/main amd64 Components [14,7 kB]
Des:9 http://security.ubuntu.com/ubuntu plucky-security/restricted amd64 Packages [80,0 kB]
Des:10 http://security.ubuntu.com/ubuntu plucky-security/restricted Translation-en [19,9 kB]
Des:11 http://security.ubuntu.com/ubuntu plucky-security/restricted amd64 Components [208 B]
Des:12 http://security.ubuntu.com/ubuntu plucky-security/universe amd64 Packages [91,7 kB]
Des:13 http://archive.ubuntu.com/ubuntu plucky-updates/main Translation-en [61,3 kB]
```

Verificación del servicio

Realizaremos una prueba de funcionamiento del servicio con:

sudo systemctl status squid

Se debe mostrar el servicio activo y escuchando.

Configuración básica de Squid

Definición de ACLs para red interna y localhost

Agregar al archivo `/etc/squid/squid.conf`:

Escuchar en el puerto estándar

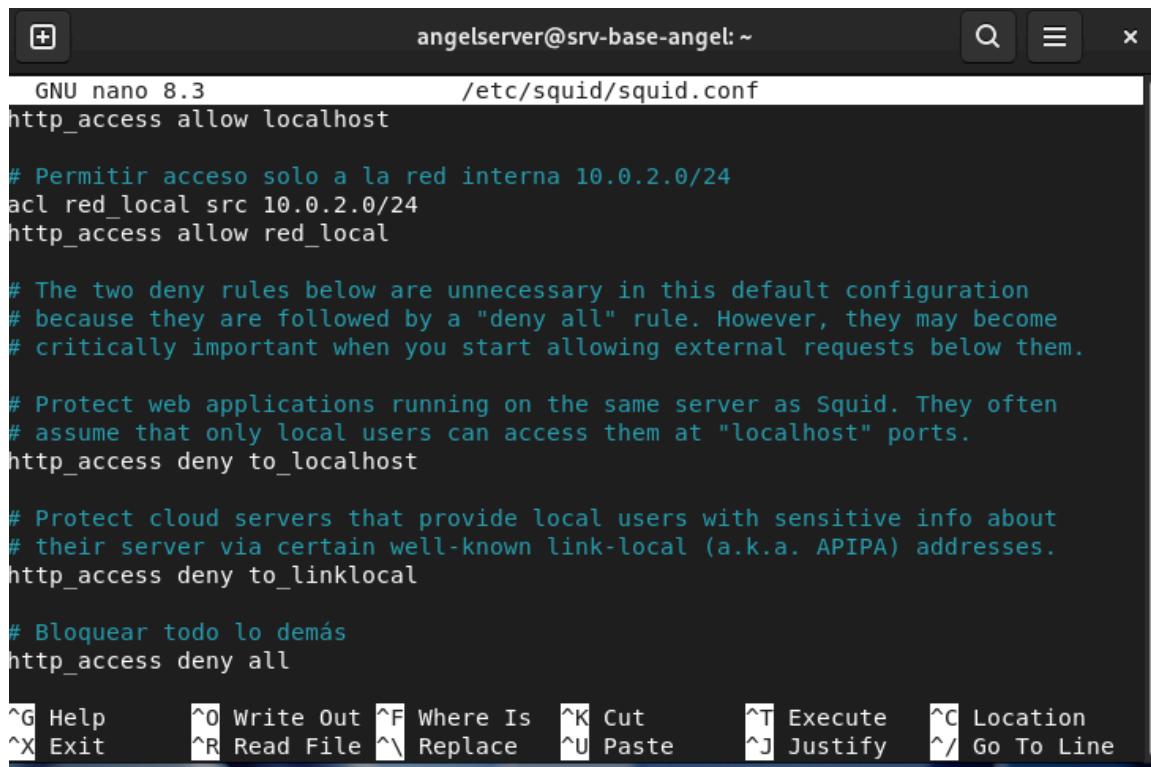
`http_port 3128`

Permitir acceso solo a la red interna 10.0.2.0/24

`acl red_local src 10.0.2.0/24`

`http_access allow red_local`

`http_access deny all`



The screenshot shows a terminal window with the title bar "angelserver@srv-base-angel: ~". The window contains the configuration file for Squid. The code is as follows:

```
GNU nano 8.3          /etc/squid/squid.conf
http_access allow localhost

# Permitir acceso solo a la red interna 10.0.2.0/24
acl red_local src 10.0.2.0/24
http_access allow red_local

# The two deny rules below are unnecessary in this default configuration
# because they are followed by a "deny all" rule. However, they may become
# critically important when you start allowing external requests below them.

# Protect web applications running on the same server as Squid. They often
# assume that only local users can access them at "localhost" ports.
http_access deny to_localhost

# Protect cloud servers that provide local users with sensitive info about
# their server via certain well-known link-local (a.k.a. APIPA) addresses.
http_access deny to_linklocal

# Bloquear todo lo demás
http_access deny all
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

- ^G Help
- ^O Write Out
- ^F Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste
- ^J Justify
- ^/ Go To Line

Configuración de grupos y políticas de acceso

Definición de grupos según IP

Asignaremos a cada grupo una IP:

```
acl desarrollo src 10.0.2.21
```

```
acl administracion src 10.0.2.22
```

```
acl marketing src 10.0.2.23
```

Estructura de directorios

Primero crea una carpeta para almacenar tus listas de URLs:

```
sudo mkdir -p /etc/squid/acl
```

Luego crearemos las listas de *ocio* y *administración*:

```
sudo nano /etc/squid/acl/sitios_ocio.txt
```

 con contenido:

facebook.com

instagram.com

tiktok.com

netflix.com

youtube.com



```
angelsrv@angelsrv-base-angel: ~
GNU nano 8.3          /etc/squid/acl/sitios_ocio.txt *
facebook.com
instagram.com
tiktok.com
netflix.com
youtube.com
```

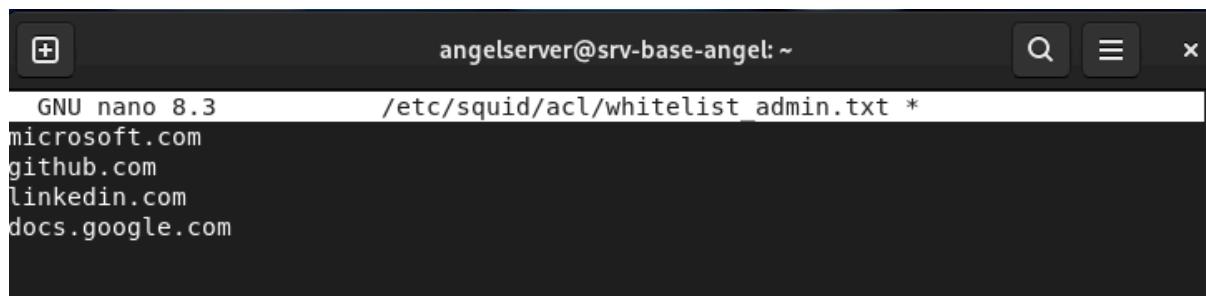
sudo nano /etc/squid/acl/whitelist_admin.txt con contenido:

microsoft.com

github.com

linkedin.com

docs.google.com



```
GNU nano 8.3          /etc/squid/acl/whitelist_admin.txt *
microsoft.com
github.com
linkedin.com
docs.google.com
```

Definir horarios(para marketing)

Edita el archivo de configuración de Squid y agregaremos:

ACL por IPs de cada grupo (puedes adaptarlas según tu red)

acl desarrollo src 10.0.2.21

acl administracion src 10.0.2.22

acl marketing src 10.0.2.23

ACLs de sitios

acl sitios_ocio dstdomain "/etc/squid/acl/sitios_ocio.txt"

acl whitelist_admin dstdomain "/etc/squid/acl/whitelist_admin.txt"

Horarios de descanso marketing (Lun-Dom 11:00-11:30 y 16:00-16:30)

acl horario_descanso time MTWHFA 11:00-11:30



```
acl horario_descanso_tarde time MTWHFA 16:00-16:30
```

```
# --- Reglas para cada grupo ---
```

```
# DESARROLLO: acceso total excepto sitios de ocio
```

```
http_access deny sitios_ocio desarrollo
```

```
http_access allow desarrollo
```

```
# ADMINISTRACIÓN: solo sitios aprobados
```

```
http_access allow administracion whitelist_admin
```

```
http_access deny administracion
```

```
# MARKETING: acceso solo en horario de descanso
```

```
http_access allow marketing horario_descanso
```

```
http_access allow marketing horario_descanso_tarde
```

```
http_access deny marketing
```



angelserver@srv-base-angel: ~

```
GNU nano 8.3          /etc/squid/squid.conf *

# Perfiles de grupo

# IPs de cada grupo
acl desarrollo src 10.0.0.2.21
acl administracion src 10.0.0.2.22
acl marketing src 10.0.0.2.23

# ACLs de sitios
acl sitios_ocio dstdomain "/etc/squid/acl/sitios_ocio.txt"
acl whitelist_admin dstdomain "/etc/squid/acl/whitelist_admin.txt"

# Horarios de descanso para marketing (lunes a viernes)
acl horario_descanso time MTWTF 11:00-11:30
acl horario_descanso_tarde time MTWTF 16:00-16:30

# --- Reglas por grupo ---

# Acceso total excepto sitios de ocio
http_access deny desarrollo sitios_ocio
http_access allow desarrollo

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

angelserver@srv-base-angel: ~

```
GNU nano 8.3          /etc/squid/squid.conf *

#DESARROLLO: acceso total excepto sitios de ocio
http_access deny desarrollo sitios_ocio
http_access allow desarrollo

# ADMINISTRACIÓN: solo sitios aprobados
http_access allow administracion whitelist_admin
http_access deny administracion

# MARKETING: solo en horarios de descanso
http_access allow marketing horario_descanso
http_access allow marketing horario_descanso_tarde
http_access deny marketing

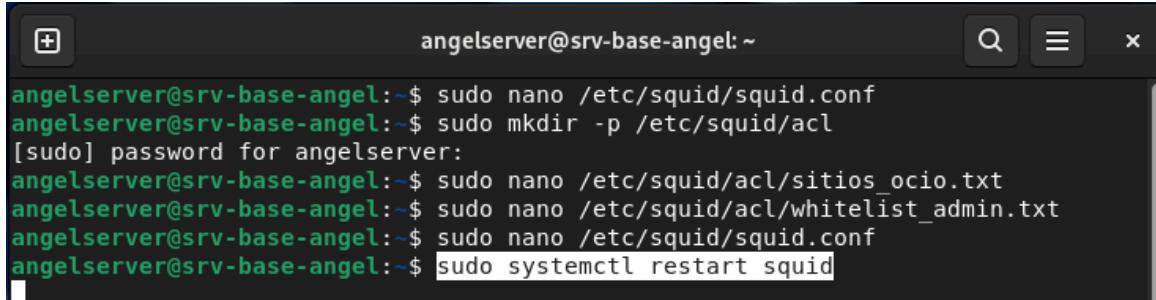
# REGLAS DE SEGURIDAD FINAL

# The two deny rules below are unnecessary in this default configuration
# because they are followed by a "deny all" rule. However, they may become
# critically important when you start allowing external requests below them.

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

Después reiniciamos con:

```
sudo systemctl restart squid
```



The screenshot shows a terminal window titled "angelserv@srv-base-angel: ~". The command history is displayed, starting with "sudo nano /etc/squid/squid.conf", followed by "sudo mkdir -p /etc/squid/acl", then a [sudo] password prompt for "angelserv@srv-base-angel". Subsequent commands include "sudo nano /etc/squid/acl/sitios_ocio.txt", "sudo nano /etc/squid/acl/whitelist_admin.txt", "sudo nano /etc/squid/squid.conf", and finally "sudo systemctl restart squid". The last command is highlighted with a white background.

```
angelserv@srv-base-angel:~$ sudo nano /etc/squid/squid.conf
angelserv@srv-base-angel:~$ sudo mkdir -p /etc/squid/acl
[sudo] password for angelserv@srv-base-angel:
angelserv@srv-base-angel:~$ sudo nano /etc/squid/acl/sitios_ocio.txt
angelserv@srv-base-angel:~$ sudo nano /etc/squid/acl/whitelist_admin.txt
angelserv@srv-base-angel:~$ sudo nano /etc/squid/squid.conf
angelserv@srv-base-angel:~$ sudo systemctl restart squid
```

Crear ACL para extensiones bloqueadas

Añade estas líneas en `squid.conf`:

```
# Bloquear descargas de archivos por extensión
acl bloqueados_archivos urlpath_regex -i \.exe$ \.mp4$ \.zip$
http_access deny bloqueados_archivos
```

Bloquear sitios multimedia para todos

Vamos a crear una ACL llamada `sitios_multimedia` que contenga los dominios a bloquear:

Crea un archivo `/etc/squid/acl/sitios_multimedia.txt` con estos contenidos:

`youtube.com`

`netflix.com`

`twitch.tv`

`vimeo.com`

Añade en `squid.conf`:

```
acl sitios_multimedia dstdomain "/etc/squid/acl/sitios_multimedia.txt"
http_access deny sitios_multimedia
```

Ajustar whitelist para administración

Ya tienes la whitelist para administración, ahora agregaremos:

```
acl whitelist_admin dstdomain "/etc/squid/acl/whitelist_admin.txt"
```

```
http_access allow administracion whitelist_admin
```

```
http_access deny administracion
```

Esta regla sólo permite a la administración navegar por los sitios de la whitelist, y bloquea cualquier otro.

Orden correcto de las reglas

Es muy importante que las reglas de denegación para extensiones y sitios multimedia estén **antes** de las reglas de acceso general para cada grupo.

```
# Bloqueo de extensiones
```

```
http_access deny bloqueados_archivos
```

```
# Bloqueo de sitios multimedia
```

```
http_access deny sitios_multimedia
```

```
# Reglas específicas de grupos (ya definidas en Fase 2)
```

```
http_access deny desarrollo sitios_ocio
```

```
http_access allow desarrollo
```

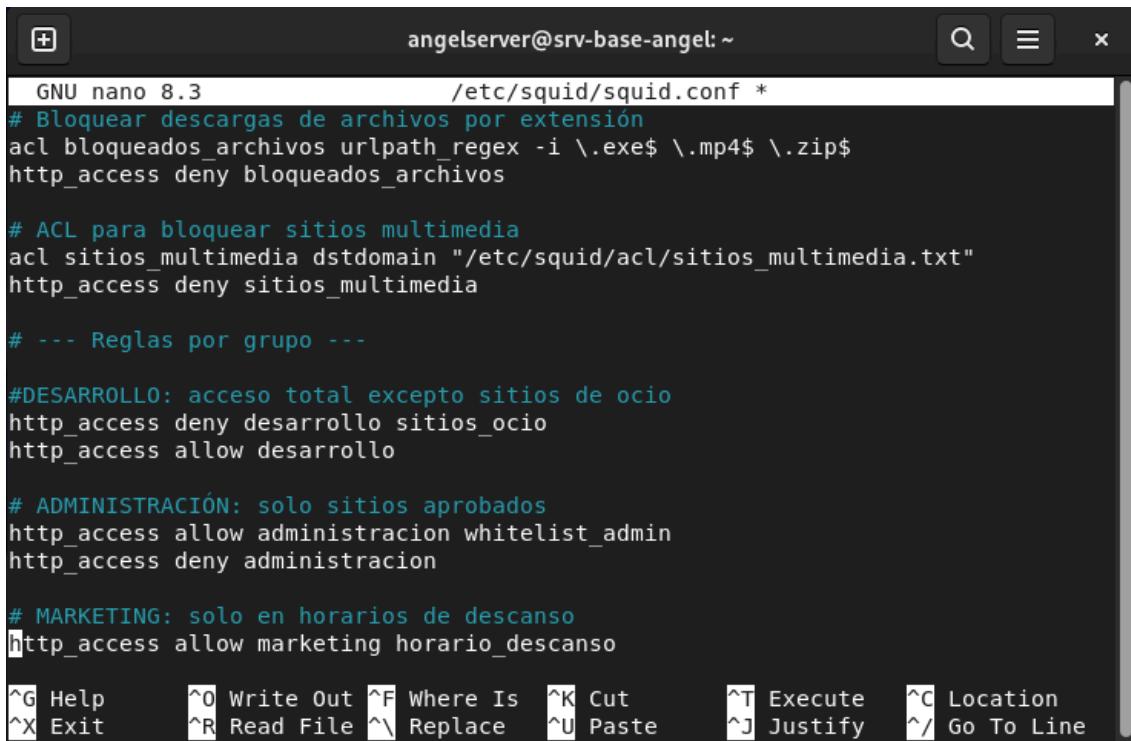
```
http_access allow administracion whitelist_admin
```

```
http_access deny administracion
```

```
http_access allow marketing horario_descanso
```

```
http_access allow marketing horario_descanso_tarde
```

```
http_access deny marketing
```



```

GNU nano 8.3          /etc/squid/squid.conf *
# Bloquear descargas de archivos por extensión
acl bloqueados_archivos urlpath_regex -i \.exe$ \.mp4$ \.zip$
http_access deny bloqueados_archivos

# ACL para bloquear sitios multimedia
acl sitios_multimedia dstdomain "/etc/squid/acl/sitios_multimedia.txt"
http_access deny sitios_multimedia

# --- Reglas por grupo ---

#DESARROLLO: acceso total excepto sitios de ocio
http_access deny desarrollo sitios_ocio
http_access allow desarrollo

# ADMINISTRACIÓN: solo sitios aprobados
http_access allow administracion whitelist_admin
http_access deny administracion

# MARKETING: solo en horarios de descanso
http_access allow marketing horario_descanso

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^L Replace   ^U Paste    ^J Justify  ^/ Go To Line

```

Reinicia Squid con:

sudo systemctl restart squid

Gestión por IP y usuarios con autenticación

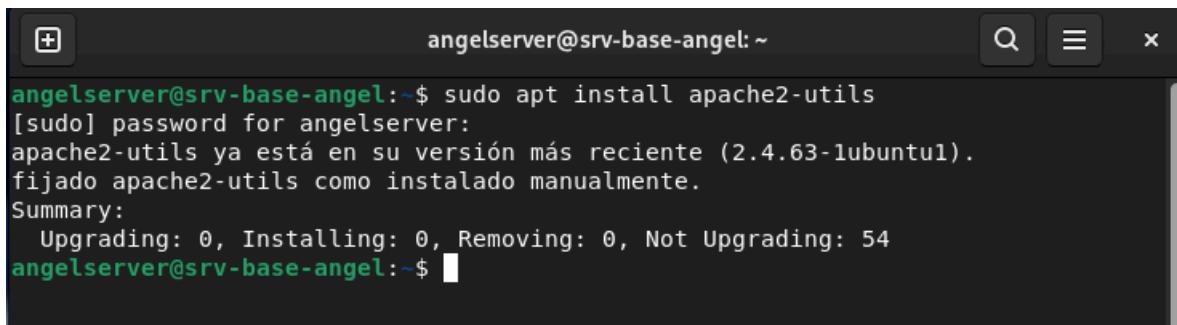
En esta fase, vamos a implementar autenticación de usuarios con Squid usando el método `basic_auth` con archivos de contraseñas gestionados por `htpasswd`. Esto permite que ciertos equipos de la red tengan que identificarse con usuario y contraseña antes de navegar.

Instalar las herramientas necesarias

Las instalaremos con:

sudo apt install apache2-utils

Esto instalará `htpasswd`, que se usará para crear el archivo de usuarios.



```
angelserv@srv-base-angel:~$ sudo apt install apache2-utils
[sudo] password for angelserv:
apache2-utils ya está en su versión más reciente (2.4.63-1ubuntu1).
fijado apache2-utils como instalado manualmente.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 54
angelserv@srv-base-angel:~$
```

Crear archivo de usuarios

Creamos el archivo donde se guardarán los usuarios y contraseñas:

```
sudo htpasswd -c /etc/squid/usuarios_autenticados invitado1
```

Te pedirá una contraseña. (*al ser de prueba usaremos: 1234.*)

Para añadir más usuarios:

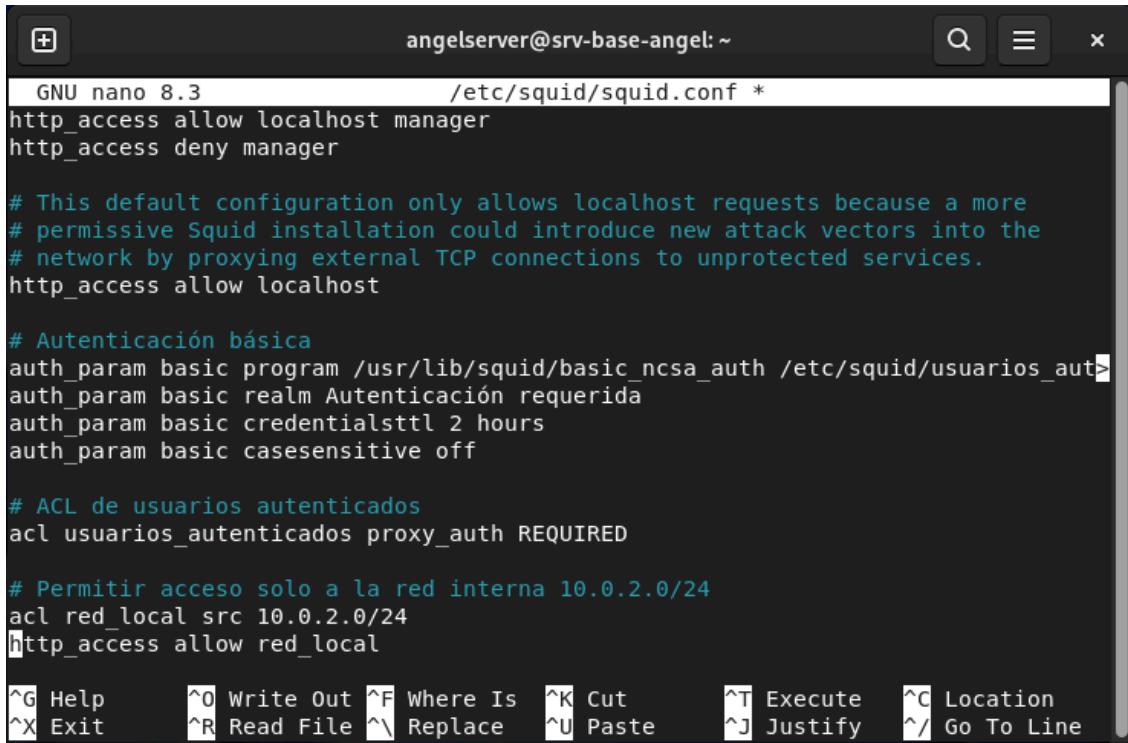
```
sudo htpasswd /etc/squid/usuarios_autenticados invitado2
```

Configurar autenticación en Squid

Abre tu archivo de configuración principal (por ejemplo `/etc/squid/squid.conf`) y añade estas líneas:

```
# Autenticación básica
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/usuarios_autenticados
auth_param basic realm Autenticación requerida
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
```

```
# ACL de usuarios autenticados
acl usuarios_autenticados proxy_auth REQUIRED
```



```

GNU nano 8.3          /etc/squid/squid.conf *
http_access allow localhost manager
http_access deny manager

# This default configuration only allows localhost requests because a more
# permissive Squid installation could introduce new attack vectors into the
# network by proxying external TCP connections to unprotected services.
http_access allow localhost

# Autenticación básica
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/usuarios_autenticados
auth_param basic realm Autenticación requerida
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off

# ACL de usuarios autenticados
acl usuarios_autenticados proxy_auth REQUIRED

# Permitir acceso solo a la red interna 10.0.2.0/24
acl red_local src 10.0.2.0/24
http_access allow red_local

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^V Replace   ^U Paste    ^J Justify  ^/ Go To Line

```

ACL para IPs que requieren autenticación

Los invitados tienen IP fija 10.0.2.21, 10.0.22. y 10.0.2.23 por lo que usaremos:

acl invitados src 10.0.2.21

acl invitados src 10.0.2.22

acl invitados src 10.0.2.23

Y permitimos acceso solo si están autenticados:

http_access allow invitados usuarios_autenticados

http_access deny invitados

ACL para IPs que requieren autenticación

Después de hacer cambios en el archivo de configuración:

sudo systemctl reload squid

Optimización de la configuración Squid

Por último, cambiaremos en el `.conf`, parámetros de caché y log para optimizar el programa, para ello añadiremos:

Carpeta de caché en disco

```
cache_dir ufs /var/spool/squid 100 16 256
```

Tamaño máximo de objetos en caché

```
maximum_object_size 50 MB
```

```
minimum_object_size 1 KB
```

Política de reemplazo de objetos (LRU: Least Recently Used)

```
cache_replacement_policy lru
```

Tiempo de vida para objetos inactivos (TTL)

```
refresh_pattern ^ftp: 1440 20% 10080
```

```
refresh_pattern ^gopher: 1440 0% 1440
```

```
refresh_pattern -i \.jpg$ 10080 90% 43200
```

```
refresh_pattern -i \.png$ 10080 90% 43200
```

```
refresh_pattern . 1440 20% 10080
```

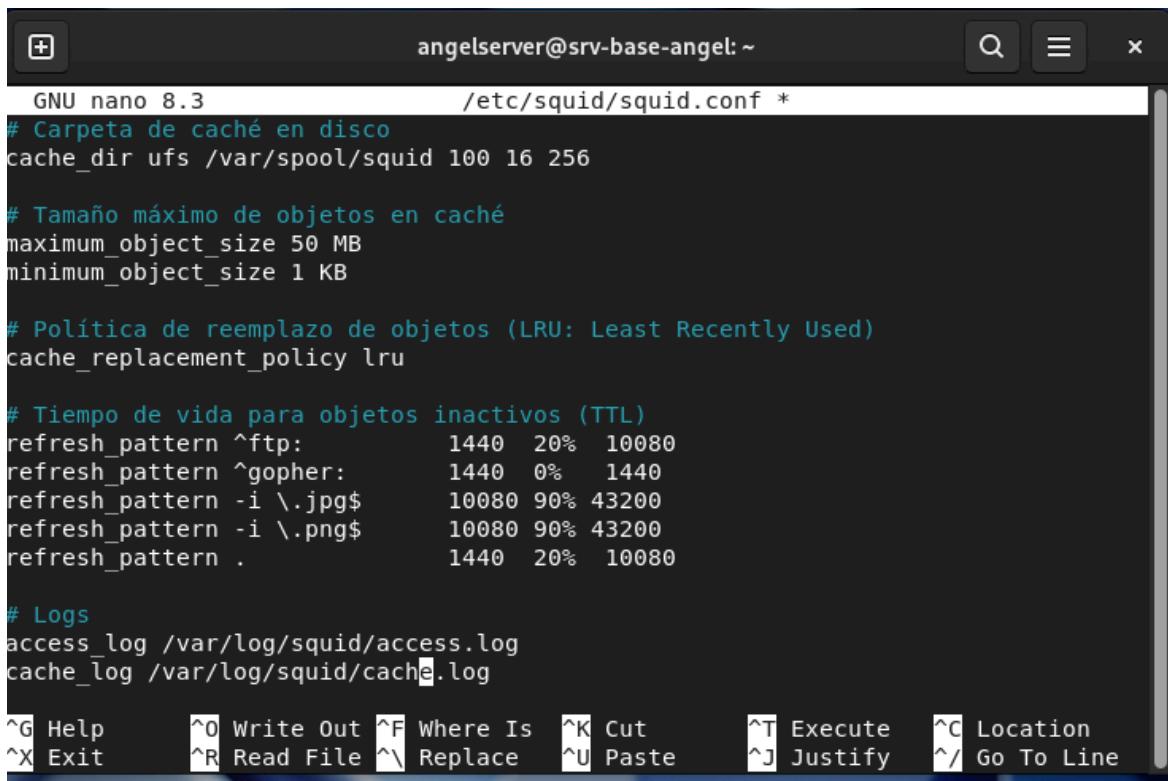
Logs

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

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

```
cache_store_log /var/log/squid/store.log
```

```
logfile_rotate 10
```



```
GNU nano 8.3          /etc/squid/squid.conf *

# Carpeta de caché en disco
cache_dir ufs /var/spool/squid 100 16 256

# Tamaño máximo de objetos en caché
maximum_object_size 50 MB
minimum_object_size 1 KB

# Política de reemplazo de objetos (LRU: Least Recently Used)
cache_replacement_policy lru

# Tiempo de vida para objetos inactivos (TTL)
refresh_pattern ^ftp:          1440 20% 10080
refresh_pattern ^gopher:        1440 0% 1440
refresh_pattern -i \.jpg$       10080 90% 43200
refresh_pattern -i \.png$       10080 90% 43200
refresh_pattern .               1440 20% 10080

# Logs
access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log

^G Help      ^O Write Out ^F Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Conclusión

La configuración detallada permite controlar el acceso a Internet según perfiles de usuario y horarios, con autenticación segura y restricciones específicas. El proxy Squid se integra correctamente en la red interna y facilita la gestión centralizada del tráfico web.