# 

PROYECTO DE ADMINISTRACIÓN DE SERVICIOS DE RED BAJO LINUX

**Profesora:** Msig. Adriana Collaguazo Jaramillo

**Grupo # 2**

**Elaborado por:**

* Emilio Ponce Alvarado
* Jonathan Yagual Peredo
* Angelo Yagual Cobos

SERVIDOR DE GOOGLE CACHÉ PARA RÁPIDA NAGEVACIÓN

# RESUMEN EJECUTIVO

Un cliente para realizar una petición debe hacerlo mediante un servidor Proxy, el cual para llegar a la nube de internet debe pasar por un router que se encarga de las peticiones al servidor de Internet.

Se debe configurar un cortafuego evitando que el cliente no evite al servidor proxy (no vaya directamente al router que da acceso a internet).

El cortafuego, cliente y servidor Proxy pertenecen a una red Interna. La red LAN incluye todos dispositivos mencionados anteriormente además un ordenador adicional que configure correctamente el cortafuego.

# DESCRIPCIÓN DEL PROBLEMA

Los usuarios pueden experimentar problemas al intentar cargar y navegar sobre un sitio web cuando la red de internet se encuentra congestionada, ya sea por un creciente número de solicitudes hacia un servidor o un plan de internet limitado en cuanto a velocidad por parte de un ISP.

En este proyecto se busca desarrollar un servidor de proxy-caché para crear copias instantáneas de páginas web en un servidor basado en Linux para agilizar la velocidad de carga de un sitio web en los clientes pertenecientes a una red local simulada.

# OBJETIVOS ESPECÍFICOS

Crear un servidor de proxy-caché utilizando Ubuntu Server para la obtención de una respuesta más rápida de navegación.

Montar un cortafuego mediante un router que impida que las peticiones del cliente siempre pase el por el servidor proxy.

Guardar los resultados de navegación en una memoria cache o disco duro para tener un registro de las páginas visitadas.

# FUNCIONAMIENTO DEL PROTOTIPO

El funcionamiento del proyecto consiste en que el cliente (VM UbuntuMATE) realizará una petición hacia el servidor caché (UBUNTU LTS), el cual una vez que le llegue la petición, realizará la petición a los servidores de internet a través del router, además para que el cliente no se salte el servidor caché, se configura el router para que únicamente permita los accesos desde la IP desde Ubuntu server, de esta forma se optimiza el acceso al internet porque cargará más rápido a las páginas que ya se habían cargado con anterioridad.

No obstante, todo este proceso es transparente para el cliente, lo único que notará es el menor tiempo que le toma para cargar cualquier página que ya haya frecuentado.

# QUE SE VA A CONSTRUIR PARA LA RESOLVER EL PROBLEMA

El proyecto consta de un servidor caché, un cliente que realiza peticiones mediante el navegador web, un router que nos permitirá acceder a internet, un cortafuego para que el cliente no evite al servidor caché. Además constara de otro ordenador para configurar el cortafuego.

Se va a virtualizar el router, el servidor con Ubuntu server y el cliente con Ubuntu Mate.

# HARDWARE / SOFTWARE

Los componentes de Hardware a utilizar son:

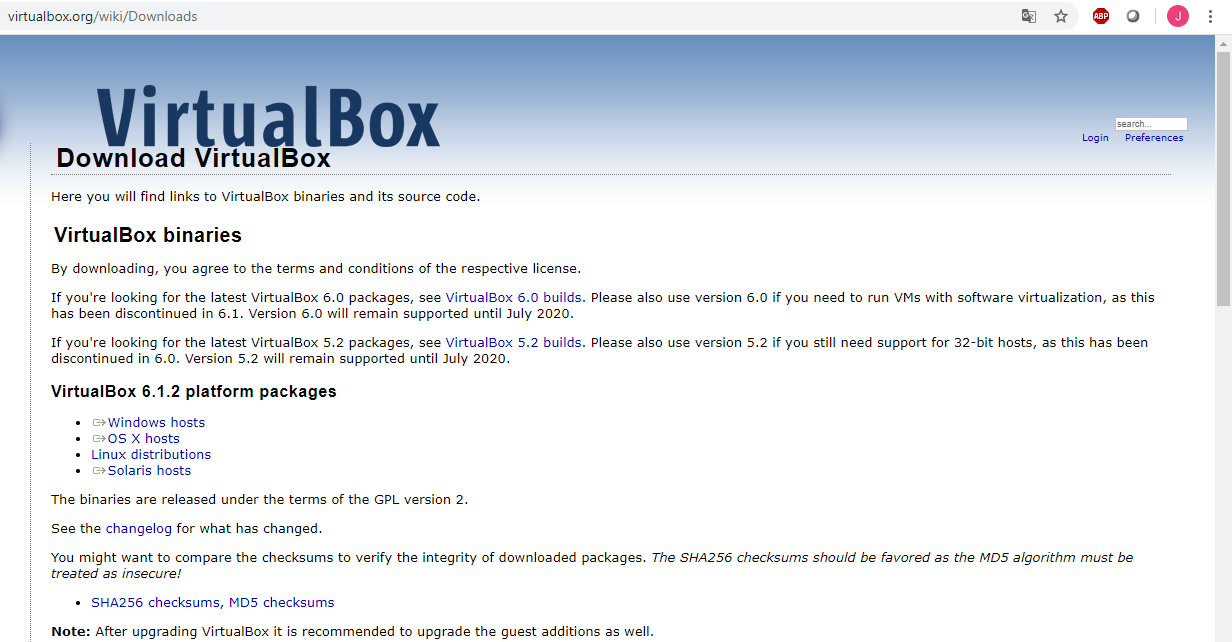
* 1 laptop – W10

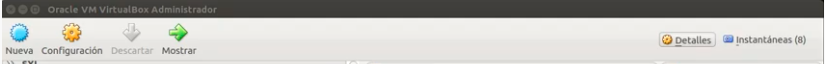
Los componentes de Software a utilizar son:

* Oracle VM Virtual Box
* Ubuntu LTS – servidor
* Squid3
* Ubuntu Mate – cliente
* RouterOS

# EXPLICACION PASO A PASO DE LA IMPLEMENTACION

1. Instalar un Administrador de Máquinas virtuales, en este caso Virtual Box.

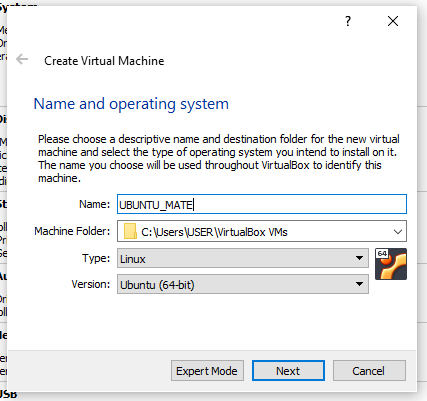




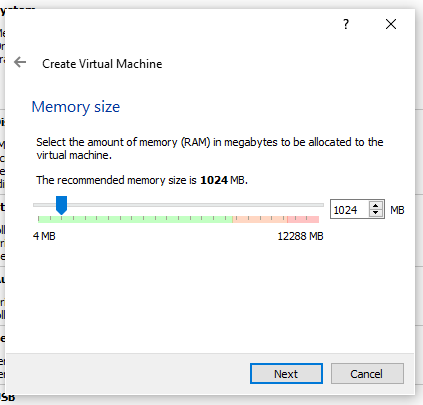
1. Crear 3 máquinas virtuales con memoria RAM de acuerdo con las capacidades de su computador anfitrión.

*Se recomiendo un mínimo de 1 GB para cada distribución de Linux y 256MB para RouterOS.*

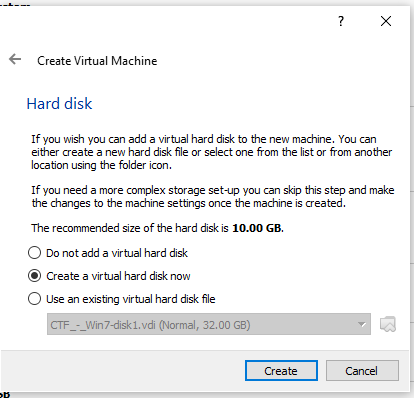
1. Abra VirtualBox y seleccione NEW.
2. Escriba el nombre de su VM y haga click en NEXT.



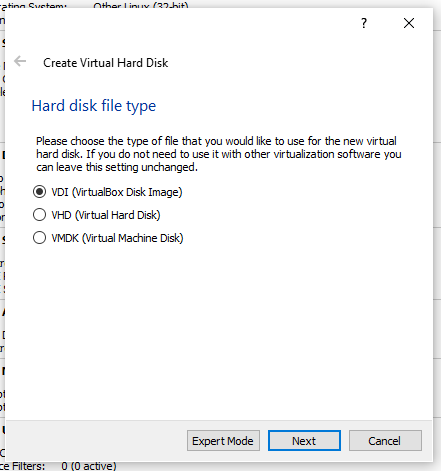
1. Selecciona la cantidad de RAM deseada y haga click en NEXT.



1. Click en CREATE.



1. Seleccione VDI y haga click en NEXT.

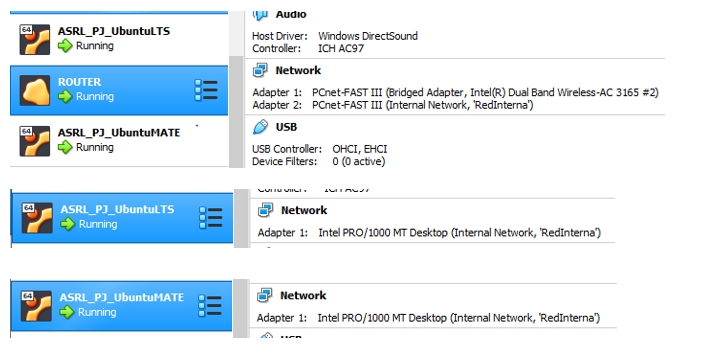


1. En las siguientes ventanas haga click en NEXT y CREATE.
2. Descargue las Imágenes de Ubuntu y ROUTEROS

* UBUNTU MATE: <https://ubuntu-mate.org/>
* UBUNTU LTS: <https://ubuntu.com/download/desktop>
* ROUTEROS: <https://mikrotik.com/download>

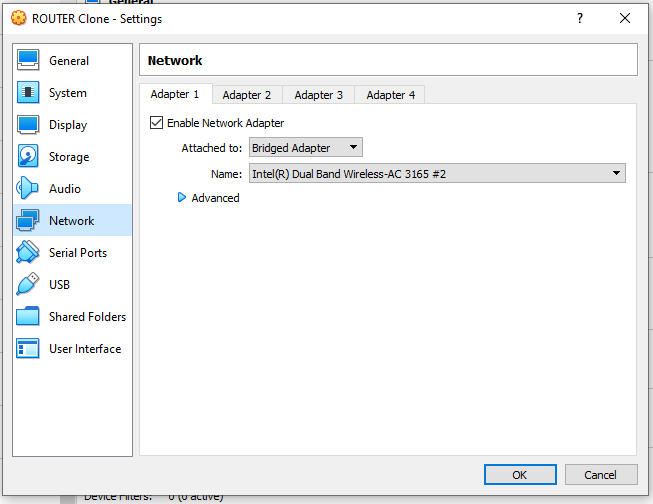
En la página de Mikrotik, seleccione la siguiente opción: **CD IMAGE**



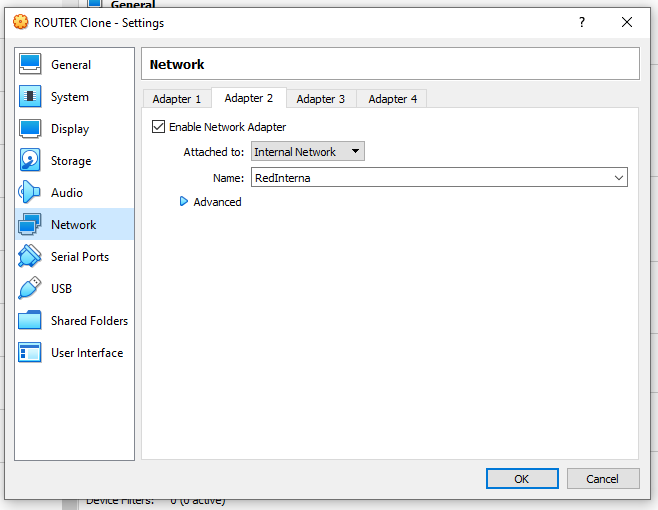
1. Instalar Ubuntu Server, Ubuntu Mate, y RouterOS.
2. Configurar las interfaces de red de las máquinas virtuales como se detalla a continuación:

PARA ROUTEROS:

Habilite el adaptador 1 y configúrelo como **Bridged Adapter**

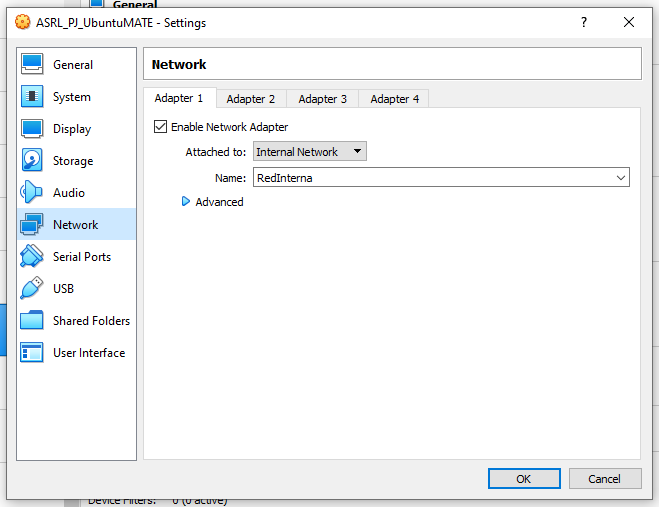


Habilite el adaptador 2 y configúrelo como **Internal Network** y proporcione un nombre de su preferencia para dicha red.



PARA LAS DISTRIBUCIONES DE LINUX:

Habilite el adaptador 1, configúrelo como **Internal Network**, elija la red creada en el Router.



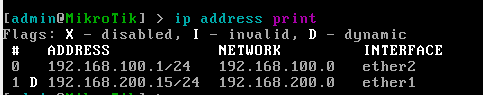
1. Una vez instalado RouterOS:

Las credenciales para iniciar sesión son:

* Username: admin
* Password:

Ejecutar el siguiente comando, Para observa la dirección IP adquirida por la interfaz eth1, que equivale a la conexión al router que nos proporciona internet:

**ip address print**



*Nota: En la captura se muestra la dirección de red de la interfaz eth2, que se configurará en el siguiente paso.*

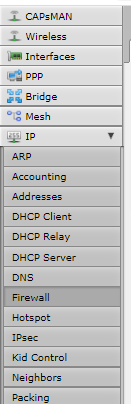
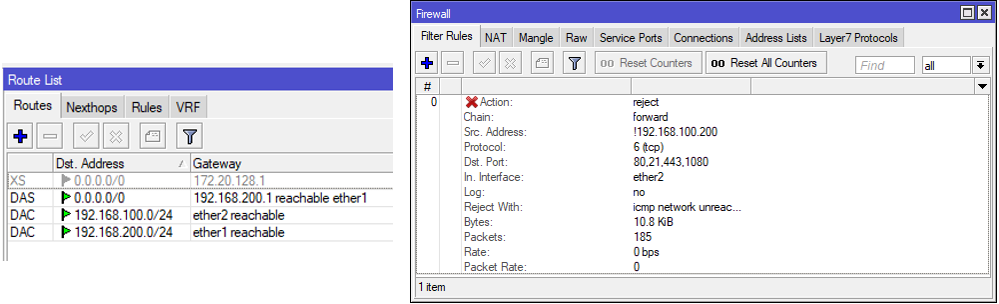
Con la dirección 192.168.200.15 ingresar en un navegador para configurar la IP de la red virtual, en QUICKSET:

En **Local Network** se configurará la dirección de la red virtual, 192.168.100.0/24, siendo la interfaz eth2 la dirección 192.168.100.1/24.

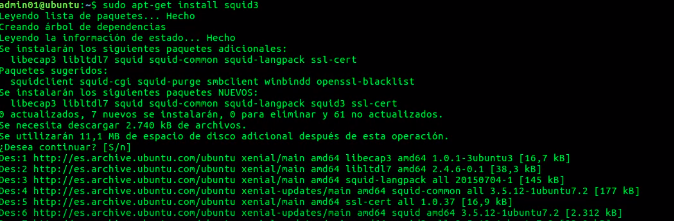
Habilitar DHCP server y NAT



En WebFig, dirígase a IP – FIREWALL:

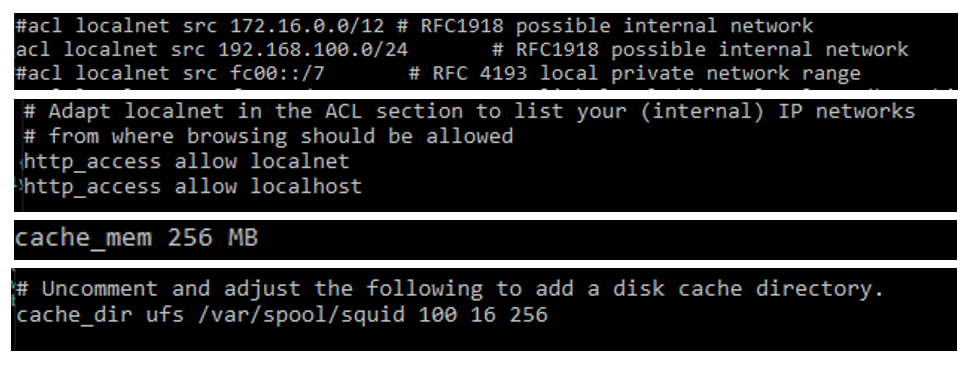
 

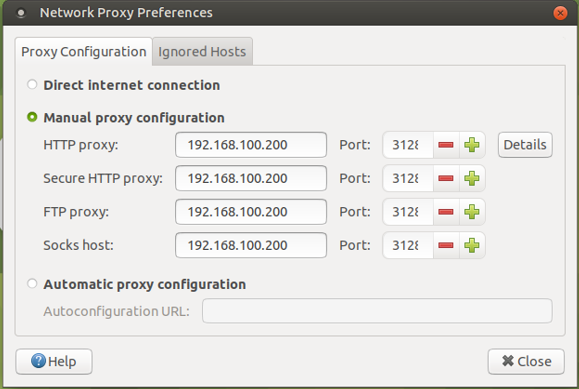
1. Instalar Squid en el servidor Ubuntu.



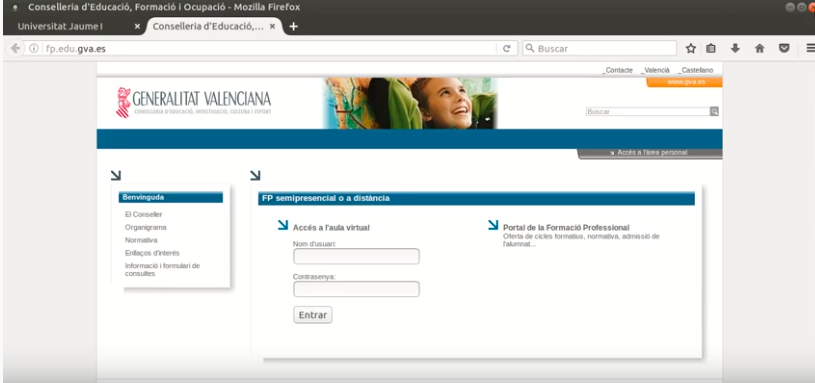
1. Configurar Squid en el archivo squid.conf.

**sudo nano /etc/squid/squid.conf**

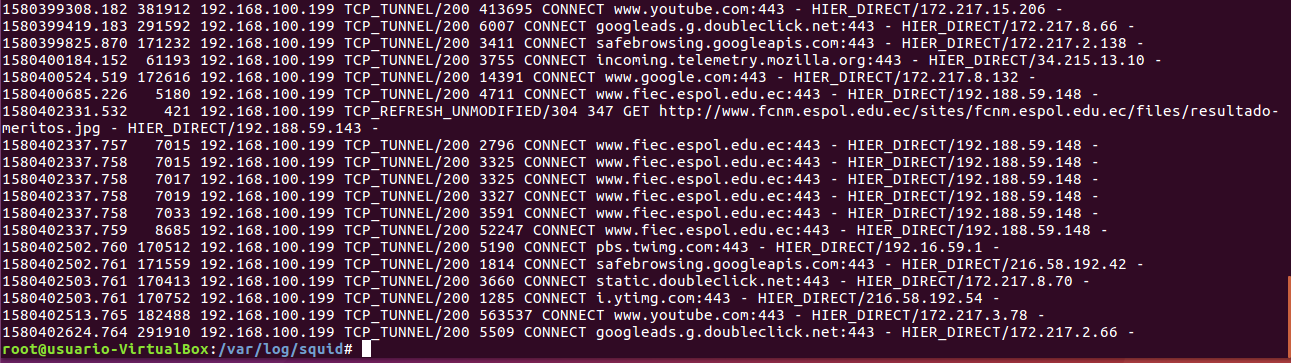
1. Aquí definimos un localnet, para que permita acceder a localnet, el puerto por donde vamos a escuchar 3128, la cantidad de memoria caché a utilizar y por ultimo poner que se utilice el disco duro como caché. 
2. Crear una estructura en /var/spool/squid para usar disco duro (esto se hace de forma automática).
3. Configurar el cliente (Ubuntu Mate) en su navegador el proxy en forma manual.



1. Finalmente entrar en el navegador a alguna página.



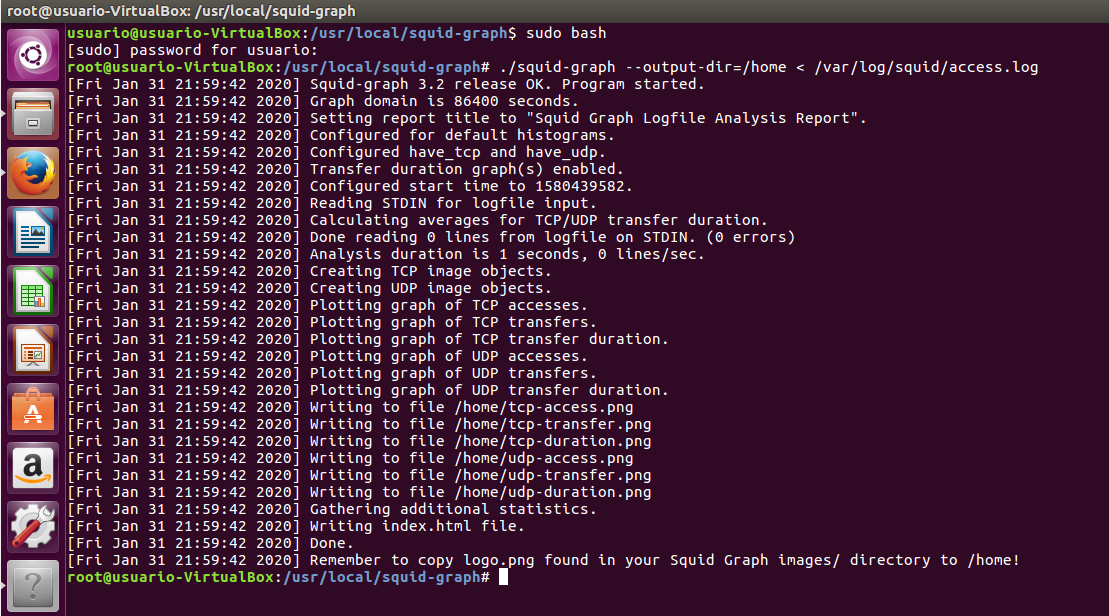
1. Para comprobar podemos revisar los archivos cache.log o Access.log donde estará guardada nuestras conexiones que hemos realizado por el cliente.

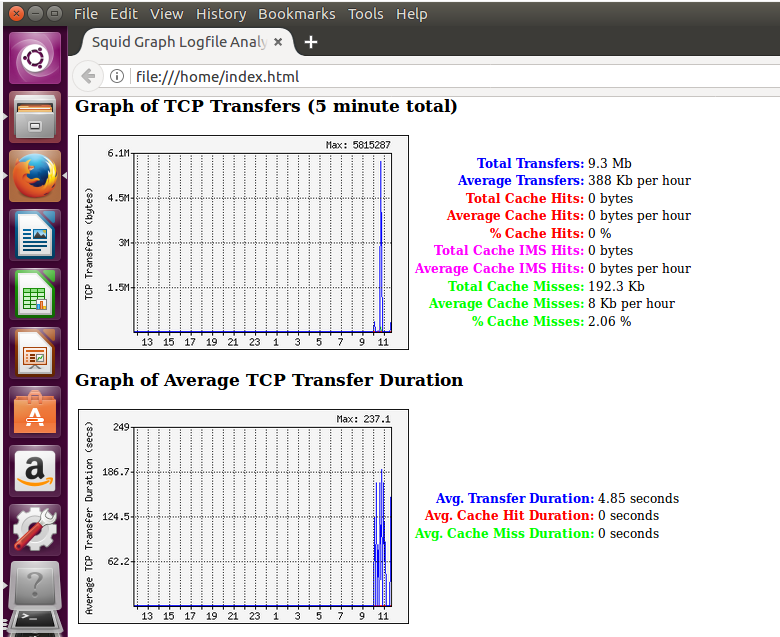


14 Para monitorear el sistema se uso una librería de SQUID – GRAPH:

Ejecute el script **squid-graph.pl** del repositorio de GITHUB:

**sudo ./squid-graph --output-dir=/home < /var/log/squid/access.log**



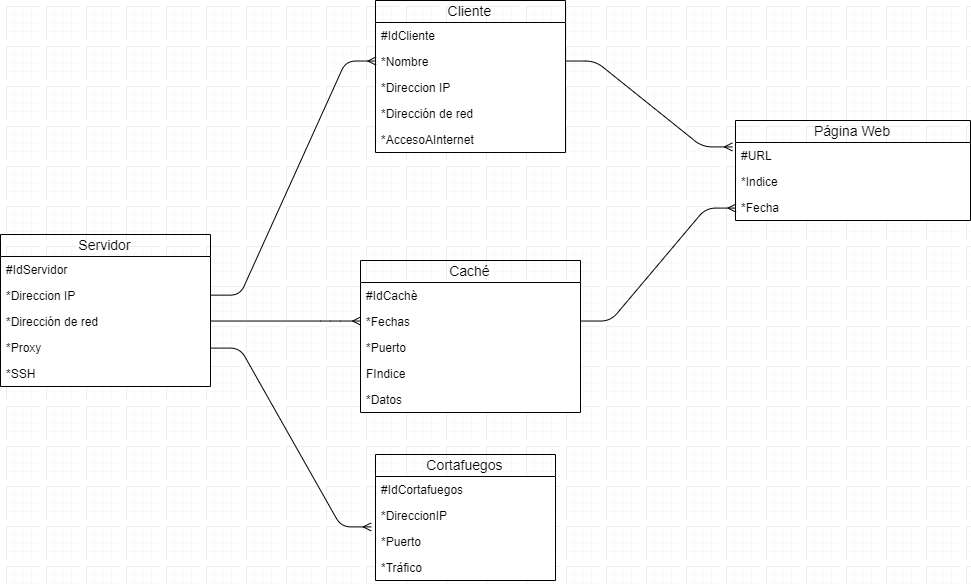


# DIAGRAMA DEL DISEÑO DEL PROYECTO

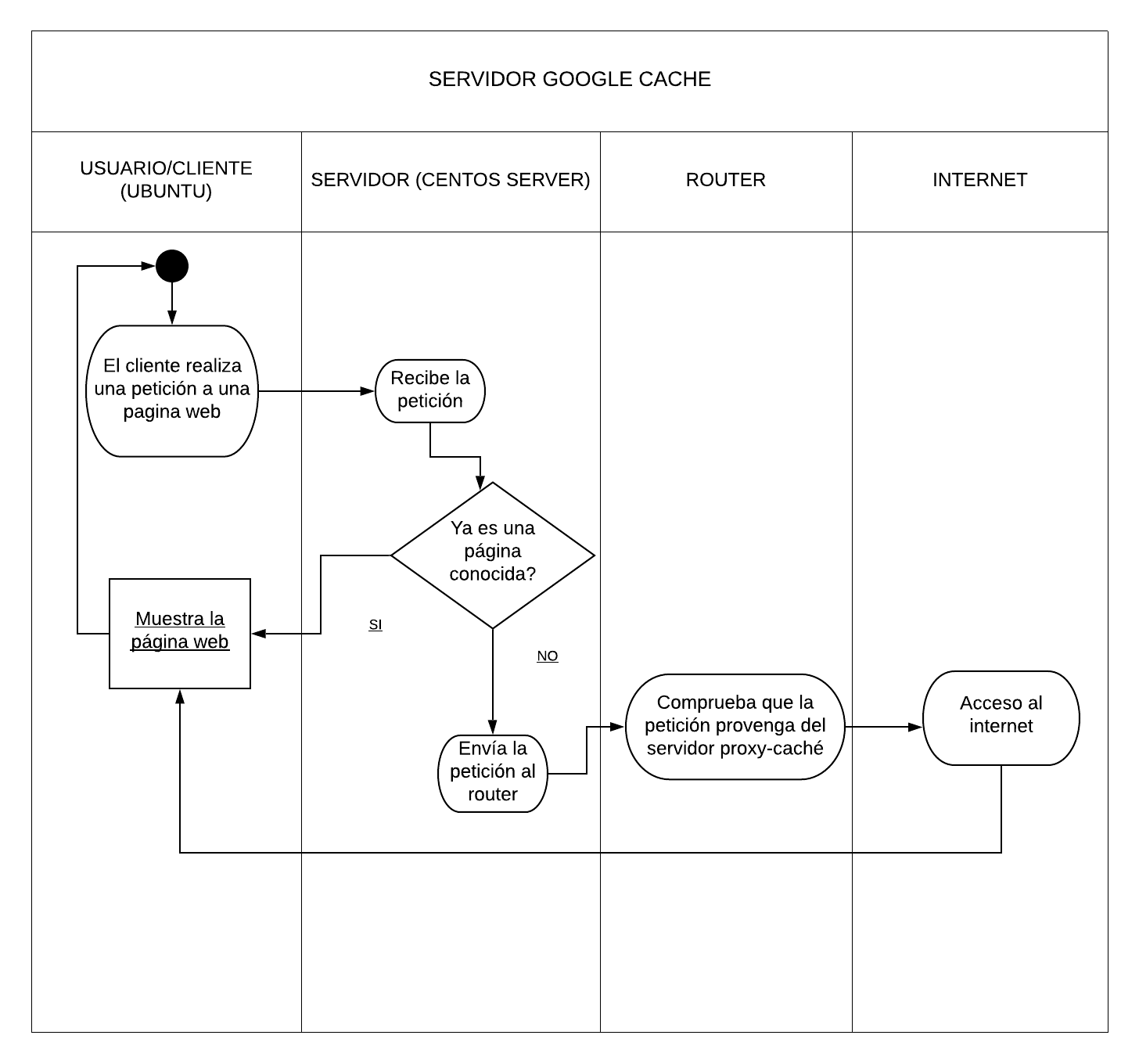
Imagen que contiene mapa

Descripción generada automáticamente

# MODELO ENTIDAD-RELACION



# DIAGRAMA DEL FUNCIONAMIENTO DEL PROYECTO



UBUNTU

DIAGRAMA DE CASOS

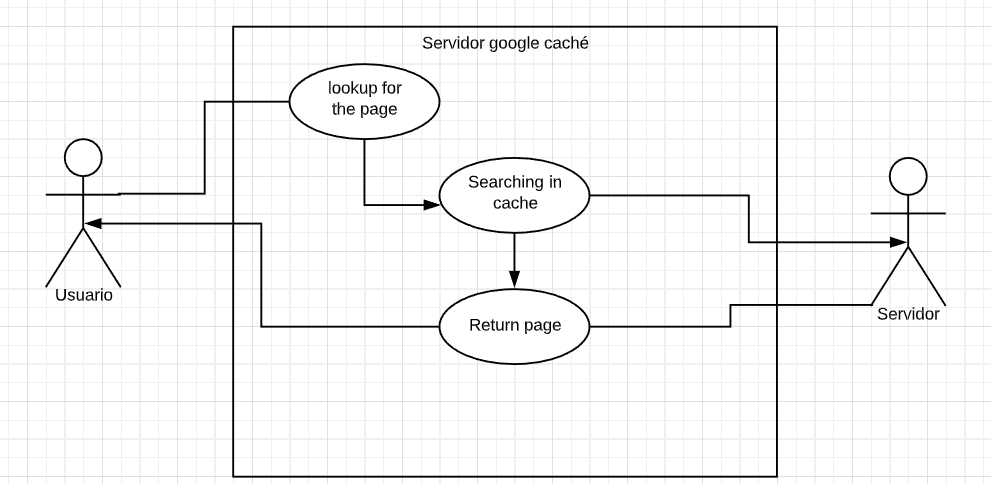
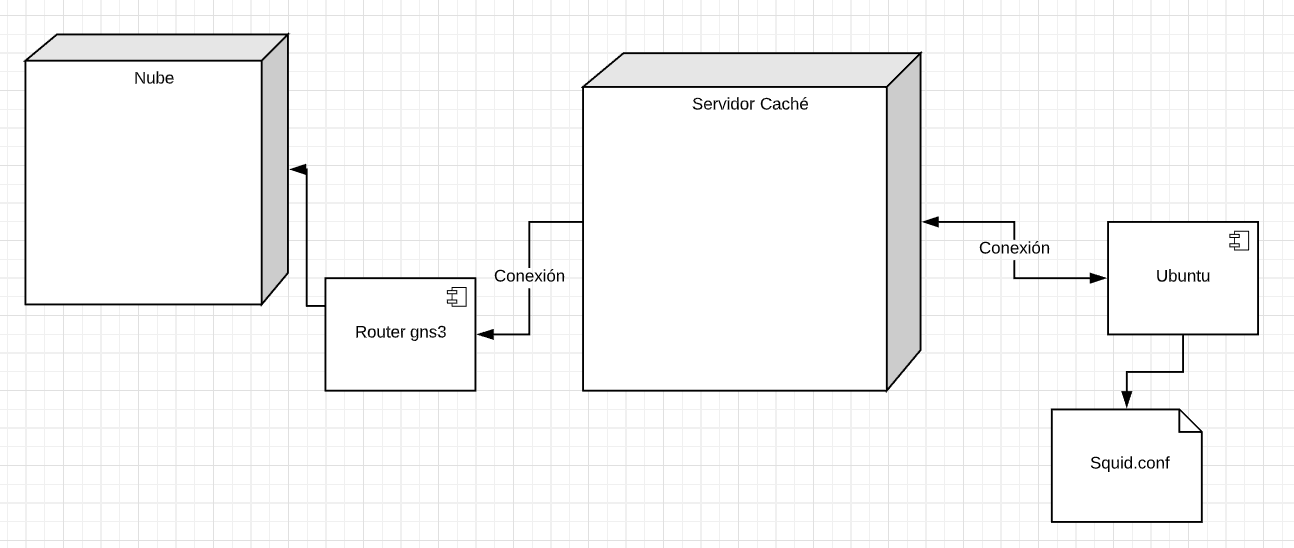
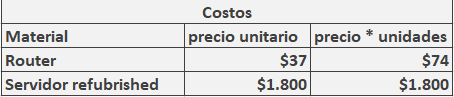


DIAGRAMA DE DESPLIEGUE



EXPLICACIÓN DEL CODIGO CON COMENTARIOS.

ANALISIS PRESUPUESTO



CONCLUSIONES

* Gracias a este modelo, pudimos mejorar el tiempo de búsqueda debido a que las páginas que ya habían sido visitadas, especialmente con más frecuencia, ya estaban almacenadas en el servidor caché.
* En vista de que sirve en nuestras máquinas virtuales con gns3, se puede replicar el mismo proyecto con equipos reales como routers, servidores, etc.
* Se demuestra que este proyecto presenta escalabilidad puesto que si se requiere de una mayor cantidad de accesos, en el caso de no tener el hardware preparado, se puede apilar de tal forma que pueda satisfacer la gran cantidad de sitios web solicitados.

Bibliografía:

* Luz S. (2011) ***Caché web (servidor proxy) ¿Qué es y cómo funciona?*** Obtenido en: <https://www.redeszone.net/2011/01/16/cache-web-servidor-proxy-que-es-y-como-funciona/>
* Fireware help ***Usar un servidor proxy de Caché. Obtenido en:*** <https://www.watchguard.com/help/docs/help-center/es-419/Content/es-419/Fireware/proxies/http/http_caching_proxy_server_c.html?TocPath=Controlar%20el%20Tr%C3%A1fico%20de%20Red%7CServidores%20Proxy%7CAcerca%20del%20Proxy%20HTTP%7C_____20>