



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD CIENCIAS DE LA INGENIERÍA
ESCUELA DE INFORMÁTICA

TESIS DE GRADO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS

TEMA

IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHE BAJO LA PLATAFORMA
GNU/LINUX, AUTOMATIZANDO SCRIPTS PARA ÓPTIMIZAR EL
MANTENIMIENTO DEL SERVIDOR.

AUTORA

STEPHANIE ELIZABETH MORA ANDRADE

DIRECTOR

ING. CARLOS CAIZAGUANO

QUEVEDO – LOS RIOS- ECUADOR

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **STEPHANIE ELIZABETH MORA ANDRADE**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento **“IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHE BAJO LA PLATAFORMA GNU/LINUX, AUTOMATIZANDO SCRIPPS PARA ÓPTMIMIZAR EL MANTENIMIENTO DEL SERVIDOR”**, es de mi exclusiva responsabilidad.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

STEPHANIE ELIZABETH MORA ANDRADE
C.I: 1205994161

CERTIFICACIÓN DEL DIRECTOR DE TESIS

El suscrito, **Ing. Carlos Oswaldo Caizaguano Chimbo**, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el Egresado **Stephanie Elizabeth Mora Andrade**, realizó la tesis de grado previo a la obtención del título de **Ingeniería en Sistemas** de grado titulada **“IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHE BAJO LA PLATAFORMA GNU/LINUX, AUTOMATIZANDO SCRITPS PARA ÓPTMIMIZAR EL MANTENIMIENTO DEL SERVIDOR”**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Carlos Oswaldo Caizaguano Chimbo.

CERTIFICACIÓN DE REDACCIÓN

Yo, **Lcda. Marjorie Torres Bolaños** con CC N°. **0701827560**, Docente de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo, certifico que he revisado la Tesis de grado del egresado **STEPHANIE ELIZABETH MORA ANDRADE**, No. Cédula **1205994161** previo a la obtención del título de Ingeniero en Sistemas, titulada **“IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHE BAJO LA PLATAFORMA GNU/LINUX, AUTOMATIZANDO SCRIPPS PARA ÓPTMIMIZAR EL MANTENIMIENTO DEL SERVIDOR”**, habiendo cumplido con la redacción y corrección ortográfica que se ha indicado.

LCDA. MARJORIE TORRES BOLAÑOS



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
TRIBUNAL DE TESIS

Presentado al Consejo Directivo como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado:

PHD. AMILKAR PURIS

ING. ANDREA ZUÑIGA

ING. ARIOSTO VICUÑA

QUEVEDO – LOS RÍOS – ECUADOR

2014

DEDICATORIA

Agradecer hoy y siempre a mi familia por sus esfuerzos realizados. El apoyo y respaldo en mis estudios. A mis padres y demás familiares ya que me brindan el apoyo, la alegría y me dan la fortaleza necesaria para seguir adelante.

A mi pequeña hija que esta siempre a mi lado Esto es para ti, Emily, pues tú eres sin duda, la parte más importante de mi vida.

Stephanie.

AGRADECIMIENTO

A Dios por sobre todas las cosas, a mi Familia por esa grata ayuda y apoyo incondicional, me permiten en este ciclo de vida seguir adelante hacia un buen futuro.

A mi tutor, Ing. Carlos Caizaguano por haberme dedicado parte de su tiempo y toda la experiencia que posee, en el transcurso de la elaboración de esta tesis.

Stephanie.

INDICE GENERAL

CAPITULO I	1
MARCO CONTEXTUAL DE LA INVESTIGACIÓN	1
1.1 Introducción	2
1.1.1. Antecedentes	2
1.2 Análisis del problema.....	3
1.2.1 Situación actual del problema	3
1.2.2 Formulación del problema.	4
1.2.2.1 Sistematización	4
1.3 Justificación	4
1.4 Objetivos.....	6
1.4.1 Generales.....	6
1.4.2 Específicos	6
1.5.1 Planteamiento	7
1.5.2 Matriz de Operacionalización	7
CAPITULO II	8
MARCO TEÓRICO DE LA INVESTIGACIÓN	8
2.1 Fundamentación Conceptual.....	9
2.1.1 Servidor	9
2.1.2 Servidor web cache	10
2.1.3 Scripts	12
2.1.4 Liberar memoria	14
2.2 Fundamentación Teórica	14
2.2.1 Sistema operativo GNU/LINUX	15
2.2.1.1 Diseño y filosofía de GNU/LINUX.....	16
2.2.1.2 Distribuciones GNU/Linux	17
2.2.2 Herramientas para implementar el servidor.....	18
2.2.2.1 Debian	18

2.2.2.2 Thundercache	19
2.2.2.3 Putty	19
2.2.2.4 Winbox	20
2.2.2.5 Squid y thundercache.....	20
2.2.2.6 Cacti en thundercache.....	21
2.2.3 Herramientas para automatizar scripts.....	21
2.2.3.1 ¿Qué es cron?.....	21
2.2.3.2 ¿Qué es crontab?.....	21
2.3 Marco Legal.....	23
2.3.1 Decreto 1014 software libre ecuador.....	23
2.4 Fundamentación Referencial.....	24
CAPITULO III.....	26
METODOLOGIA DE LA INVESTIGACIÓN.....	26
3.1 Materiales y métodos.....	27
3.1.1 Métodos y técnicas.....	27
3.1.1.1. Método Inductivo.....	27
3.1.1.2. Método Deductivo	27
3.1.2 Materiales y presupuesto	27
3.1.3 Factibilidad	30
3.1.3.1 Recursos y Financiamiento	30
3.1.4 Cronograma	30
3.2 Tipos de Investigación	32
3.2 .1 Investigación cuasi-experimental	32
3.2.2 Investigación de campo.....	32
3.3 Diseño de la Investigación.....	33
3.3.1 Técnicas de Investigación	33
3.3.2 Encuesta	33
3.4 Población Y Muestra	33
3.4.1 Población.....	33
3.4.2 Muestra	34

3.4.3 Aplicación de la muestra	34
3.4.4 Cuestionario para encuesta dirigida a los estudiantes de los laboratorios de TICS	35
3.4.5 Análisis e interpretación de resultados	36
CAPITULO IV	45
RESULTADOS Y DISCUSIÓN	45
4.1 Desarrollo del proyecto	46
4.1.1 Instalación y configuración del sistema operativo Debian	46
4.1.2 Instalación del sistema de catching Thundercache	46
4.1.3 Instalación de winbox	55
4.1.4 Configuración del Routerboard 750 Gl	55
4.1.5 Diseño y desarrollo de scripts	69
4.1.6 Automatización script	72
4.1.7 Funcionamiento de las tareas crontab	74
4.2 Comprobación De Hipótesis	74
4.2.1 Variables Independientes	74
4.2.2 Variables Dependientes	75
4.2.3 Mediciones realizadas en el edificio de tics	75
4.2.4 Análisis de resultados	81
4.2.5 Gráficas de rendimiento tomadas en laboratorio de tics	83
CAPITULO V	88
CONCLUSIONES Y RECOMENDACIONES	88
5. 1 Conclusiones	89
5.2 Recomendaciones	90
CAPITULO VI	91
LITERATURA CITADA	91
6.1 Bibliografía	92
6.2 Enlaces web	93
CAPITULO VII	95
ANEXOS	95

Anexo 1: Instalación Sistema Operativo Debian.....	97
Anexo 2: Diseño de una red con thundercache	101
Anexo 3: Funcionamiento de la red cuando no hay cache.	102
Anexo 4: Funcionamiento de la red cuando hay cache.	103
Anexo 5: Árbol de problemas	104
Anexo 6: Datos Generales.....	105

INDICE DE TABLAS

Tabla 1: Matriz de Operacionalización -----	7
Tabla 2: Comandos Básicos Debian-----	13
Tabla 3: Formato Crontab -----	22
Tabla 4: Hardware requerido -----	28
Tabla 5: Software requerido.-----	28
Tabla 6: Suministros requeridos. -----	29
Tabla 7: Presupuesto del proyecto. -----	29
Tabla 8: Pregunta 1 -----	36
Tabla 9: Pregunta 2 -----	37
Tabla 10: Pregunta 3-----	38
Tabla 11: Pregunta 4-----	39
Tabla 12: Pregunta 5-----	40
Tabla 13: Pregunta 6-----	41
Tabla 14: Pregunta 7-----	42
Tabla 15: Pregunta 8-----	43
Tabla 16: Pregunta 9-----	44

INDICE DE FIGURAS

FIGURA 1: Memoria ram	14
FIGURA 2: Comandos instalación thundercache	47
FIGURA 3: Ingreso de comandos thundercache.	47
FIGURA 4: Permisos de lectura.	48
FIGURA 5: Instalación squid.	49
FIGURA 6: Usuario de mysql.	49
FIGURA 7: Contraseña mysql.	50
FIGURA 8: Contraseña cacti.	50
FIGURA 9: Instalación thundercache finalizada.	50
FIGURA 10: Formateo disco.	51
FIGURA 11: Disco agregado thundercache listo.	52
FIGURA 12: Interfaz gráfica thundercache	53
FIGURA 13: Reporte del cache y sus estadísticas	54
FIGURA 14: Conexión rb y winbox	55
FIGURA 15: Winbox detecta al rb mediante mac.	56
FIGURA 16: Conectarse con winbox.	56
FIGURA 17: Menú winbox.	57
FIGURA 18: Lista de interfaces winbox.	59
FIGURA 19: Nombrando las interfaces.	60
FIGURA 20: Asignando ip a interfaces.	61
FIGURA 21: Configurando nat.	62
FIGURA 22: Configurando el mascarado.	62
FIGURA 23: Ruteando la puerta de enlace.	63
FIGURA 24: Agregando clientes y asignando ancho de banda.	64
FIGURA 25: Amarre de las interfaces con la mac.	65
FIGURA 26: Configuración dns.	66
FIGURA 27: Redireccionando a los usuarios.	67
FIGURA 28: Asignando mascarado.	67

FIGURA 29: Colocando ip servidor para redirecciones.-----	68
FIGURA 30: Re direccionando a los usuarios.-----	68
FIGURA 31: Vista firewall.-----	69
FIGURA 32: Funcionamiento script.-----	70
FIGURA 33: Estado de cron.-----	73
FIGURA 34: Editando fichero crontab.-----	73
FIGURA 35: Fichero crontab.-----	73
FIGURA 36: Proceso ejecutado por el demonio.-----	74
FIGURA 37: Cronograma.-----	31
FIGURA 39: Monitoreo descarga de video -----	83
FIGURA 40: Monitoreo tráfico cache.-----	84
FIGURA 41: Monitoreo descarga instalador-----	85
FIGURA 42: Monitoreo descarga de instalador.-----	86
FIGURA 43: Monitoreo tráfico cache.-----	87
FIGURA 44: Inicio debian-----	97
FIGURA 45: Selección del país debian -----	98
FIGURA 46: Selección idioma del teclado debian -----	98
FIGURA 47: Configuración de red debian -----	99
FIGURA 48: Configuración usuarios y contraseña debian -----	99
FIGURA 49: Asignación de usuario a debian -----	99
FIGURA 50: Configurando la red debian -----	100
FIGURA 51: Asignando ip al equipo debian -----	100
FIGURA 52: Servicio grub debian -----	100
FIGURA 53: Instalación finalizada debian -----	100
FIGURA 54: Diseño de la red.-----	101
FIGURA 55: Objeto web no encontrado en cache.-----	102
FIGURA 56: Objeto web encontrado en cache.-----	103

RESUMEN EJECUTIVO

En la presente investigación se ha implementado un servidor web cache, automatizando sus procesos de mantenimiento mediante la utilización de scripts.

El servidor trabaja mediante un sistema operativo de plataforma libre el cual es Debian 3.1.2, thundercache que será el encargado de cachear los objetos web, para responder a nuevas peticiones de usuarios.

Se utiliza un routerboard 750GL que fue configurado mediante el winbox para su uso en la red, entre las configuraciones que se realizan: el re direccionamiento de las peticiones hacia el servidor y no directamente hacia la WAN.

Entre las herramientas que se usaron tenemos el Putty que es la terminal que usamos para manipular comandos Debian desde otro equipo hacia el servidor, El winscp utilizado cuando se desea acceder de forma gráfica al contenido del servidor.

Después de implementado el servidor, el funcionamiento de la red será eficiente, los usuarios de la red experimentarán una mejora en el rendimiento de la misma entre las cuales citamos:

Reduce los tiempos de respuesta.

Si la página web que se solicita está en la caché del servidor, ésta se sirve sin necesidad de acceder de nuevo al servidor original, con lo cual se ahorra tiempo.

Disminuye el tráfico en la red y el consumo de ancho de banda.

Si la página web está almacenada en la caché del servidor, la petición no sale de la red local y no será necesario hacer uso de la línea exterior consiguiendo así un ahorro en la utilización del ancho de banda.

La creación y automatización de los scripts nos permitirá brindarle al servidor un mantenimiento idóneo ya que este proceso es de importancia para el buen funcionamiento del equipo.

Otro factor importante de esta investigación es la contribución al aprendizaje de nuevas tecnologías basadas en plataformas libres incidiendo en economizar en gastos de software.

ABSTRAC

In the present investigation we have implemented a web cache server maintenance by automating processes using scripts.

The server works through a free operating system platform which is Debian 3.1.2 , which will be responsible Thundercache to cache web objects , to respond to new user requests.

The re routing of requests to the server and not directly to the wan -a 750GL routerboard that was configured using the winbox for use in the network between the configurations we used .

Among the tools we have used Putty which is the terminal commands used to manipulate Debian from another computer to the server , the Winscp used when access graphically server content .

After the server implemented on the network operation will be efficient , the network users experience an improvement in performance of the same among which are:

Reduce response times.

If the web page you requested is in the cache server , it is served without logging back to the original server , thus saving time.

Reduces network traffic and bandwidth consumption.

If the web page is stored in the cache server , the request does not leave the local network and will not need to use the outside line and achieving savings in bandwidth utilization .

The creation and automation scripts to allow us to provide a suitable server maintenance as this process is important for the proper functioning of the team.

Another important factor of this research is to contribute to the learning of new technologies based on free platforms stressing save on software costs .

CAPÍTULO I

MARCO CONTEXTUAL DE LA INVESTIGACIÓN

1.1 Introducción

1.1.1. Antecedentes

Se llama caché web a la caché que almacena documentos web (es decir, páginas, imágenes) para reducir el ancho de banda consumido, la carga de los servidores y el retardo en la descarga. Un caché web almacena copias de los documentos que pasan por él, de forma que subsiguientes peticiones pueden ser respondidas por el propio caché,

ThunderCache es un potente software de lo que se conoce como *Web Cache*, una especie de “disco duro” que poseen los grandes servidores de Internet en el que se almacenan datos, te permite gestionar el contenido y así cuando uno de los usuarios acceda de nuevo a ese contenido no tendrá que reiniciar la descarga desde el principio, lo que se traduce en una mayor velocidad para él usuario y en un menor consumo de ancho de banda para una empresa.

ThunderCache surgió para dar solución al problema de solicitar cada vez más ancho de banda, con una inversión mucho menor y más eficiente. Sin importar el tamaño de tu red, sean 10, 100 o 1000 usuarios, los beneficios de utilizar esta tecnología son asombrosos y muy convenientes. Esta poderosa aplicación sigue un paradigma completamente distinto al existente y es capaz de mejorar el servicio y la experiencia del cliente es un sistema tremendamente estable a pesar de lo *pesado* y complejo de la tarea que realizan.

En la UTEQ se cuenta con laboratorios que son utilizados por un gran número de estudiantes con motivos académicos, realizan una variedad de actividades como descargas entre otros, debido a esto la tecnología web cache podría optimizar el rendimiento de la red en la UTEQ.

1.2 Análisis del problema

1.2.1 Situación actual del problema

En la UTEQ son muchos los estudiantes que utilizan los laboratorios de la institución para realizar servicios on-line entre los que podemos citar (documentos, trámites, consultas, email, información, descargas), que están vinculados con su formación profesional.

Dichos estudiantes para cumplir con sus labores académicas tienen necesidad de realizar descargas de diferentes sitios en la web, suele ocurrir que alumnos de un curso deben realizar la misma descarga significando que se va a consumir el ancho de banda de forma innecesaria para descargar el archivo que fue descargado con anterioridad.

Implicando que en cierto momento habrá una congestión dentro de la red sin considerar el tamaño del archivo que se ha descargado que suelen ser isos, de gran tamaño, igualmente ocurre con videos, ver anexo 5.

La UTEQ no cuenta con un sistema de catching, siendo la conexión directamente de firewall, incitando un consumo innecesario de ancho de banda y demasiadas solicitudes al enlace WAN.

Una solución a estos inconvenientes es implementar un sistema web cache que comparta todos los recursos web almacenados, de manera que estos estén disponibles cuando los usuarios los requieran.

1.2.2 Formulación del problema.

¿Cómo mejorar el acceso a internet en los laboratorios de la unidad de tics en la UTEQ?

1.2.2.1 Sistematización

¿Al implementar un servidor web cache podremos reducir el consumo de ancho de banda?

¿Al implementar un servidor web cache podremos disminuir el tráfico y congestión en el enlace WAN?

¿Al implementar el servidor web cache se disminuirá el tiempo de respuesta a las peticiones de los usuarios?

1.3 Justificación

SEGÚN DECRETO EJECUTIVO 1014 EXPEDIDO POR LA PRESIDENCIA DE LA REPÚBLICA DEL ECUADOR, establece como política pública para las entidades de la administración pública central la utilización de software libre en sus sistemas y equipamientos informáticos.

Adicionalmente LA SUBSECRETARÍA DE TECNOLOGÍAS DE LA INFORMACIÓN DEL ECUADOR, nos dice que el uso de Software Libre asegura la soberanía tecnológica, impulsa la innovación nacional, optimiza el gasto estatal fortaleciendo el desarrollo local y facilita la inclusión digital.

De acuerdo a lo que establece la subsecretaría de tecnologías del Ecuador:

Adoptando Software Libre y con las posibilidades que éste ofrece de acceder al código fuente, muchos usuarios pasarán de ser consumidores a ser desarrolladores de software.

Utilizando Software Libre se puede saber qué está haciendo realmente un programa, qué tipo de información maneja y cómo lo hace. Una buena seguridad debe basarse en la transparencia.

El Software Libre es producido utilizando especificaciones y estándares tecnológicos libres y públicos.

El Software Libre favorece la democratización de la información permitiendo la utilización de protocolos, formatos y lenguajes.

Las redes informáticas son un sistema de comunicación que conecta ordenadores y otros equipos informáticos entre sí, con la finalidad de compartir recursos entre los usuarios.

Pero este mecanismo suele ser afectado por muchas razones que provocan inestabilidad de la red. Las empresas tienen que combatir cualquier tipo de problema que ataque el rendimiento global de la organización.

Thundercache es un potente software conocido como web cache que cumple con las disposiciones gubernamentales realizadas en el país, este sistema permitirá dar un mejor servicio a los usuarios de los laboratorios en la UTEQ, siendo implementado de forma correcta en las horas pico donde el porcentaje de accesos es alto por parte de los usuarios, se reducirá las cantidades de solicitudes del navegador que tienen que viajar y regresar a través de internet, hacia y desde los servidores donde se encuentra el contenido original.

El caching eliminará instantáneamente muchos problemas relacionados con el congestionamiento y tráfico de internet, cuando se acceda a la información de la cache se verá una reducción en el tiempo de acceso a la página, se disfrutará de descargas más rápidas.

Esto significará que entre más solicitudes del navegador puedan ser ejecutadas por el cache menos será el ancho de banda requerido por el enlace WAN.

1.4 Objetivos

1.4.1 Generales

Implementar un servidor web cache para mejorar el servicio de internet a los usuarios de los laboratorios de la unidad de Tics en la UTEQ.

1.4.2 Específicos

- Reducir el consumo de ancho de banda de los laboratorios de la unidad de tics en la UTEQ.
- Reducir la carga de accesos y así evitar congestión en el enlace WAN de los laboratorios de la unidad de tics en la UTEQ.
- Reducir el tiempo y Aumentar la velocidad en las descargas.
- Implementar y automatizar un script para realizar el mantenimiento del servidor.

1.5 Hipótesis

1.5.1 Planteamiento

La implementación del servidor web cache en los laboratorios de tics en la UTEQ nos permitirá mejorar el servicio de internet a los usuarios de dicha institución.

1.5.2 Matriz de Operacionalización

Tabla 1: Matriz de Operacionalización

VARIABLE	DEFINICIÓN	DIMENSIÓN	INDICADORES
INDEPENDIENTE Servidor web cache	Servidor que almacena objetos web, el cual responderá a las peticiones de los usuarios.	EFICIENCIA	-Tiempo de descarga. -Tiempo de respuesta a peticiones.
		FUNCIONALIDAD	-Menor tráfico y congestión para internet.
DEPENDIENTE Rendimiento de la red.	La forma con la que interactúa la red con los usuarios.	DOWNLOAD	-Velocidad de descarga.

CAPÍTULO II

MARCO TEÓRICO DE LA INVESTIGACIÓN

2.1 Fundamentación Conceptual

2.1.1 Servidor¹

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

Servidor de archivos: es aquel que almacena y sirve ficheros a equipos de una red.

Servidor de Directorio Activo/Dominio: es el que mantiene la información sobre los usuarios, equipos y grupos de una red.

Servidor de Impresión: se encarga de servir impresoras a los equipos cliente y poner en la cola los trabajos de impresión que estos generan.

Servidor de Correo: se encarga de gestionar el flujo de correo electrónico de los usuarios, envía, recibe y almacena los correos de una organización.

Servidor de Fax: gestiona el envío, recepción y almacenamiento de los faxes.

¹ <http://www.masadelante.com/faqs/servidor>

Servidor Proxy: su principal función es guardar en memoria caché las páginas web a las que acceden los usuarios de la red durante un cierto tiempo, de esta forma las siguientes veces que estos acceden al mismo contenido, la respuesta es más rápida.

Servidor Web: Almacena contenido web y lo pone al servicio de aquellos usuarios que lo solicitan.

Servidor de Base de Datos: es aquel que provee servicios de base de datos a otros programas o equipos cliente.

Servidor DNS: permite establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.

Servidor DHCP: este dispone de un rango de direcciones con el cual, asigna automáticamente los parámetros de configuración de red IP a las máquinas cliente cuando estas realizan una solicitud.

Servidor FTP: su función es permitir el intercambio de ficheros entre equipos, normalmente su aplicación va muy ligada a los servidores Web.

2.1.2 Servidor web cache²

Es un equipo dentro de la red que recibe solicitudes HTTP en nombre de un servidor web de origen. La caché consiste en almacenar los objetos que han sido solicitados por el cliente recientemente. Ver anexo 2.

² <http://www.redeszone.net/2011/01/16/cache-web-servidor-proxy-que-es-y-como-funciona/>

Funcionamiento

Imaginemos que abrimos una página web, lo primero que hace el navegador es establecer una conexión TCP con el servidor proxy y envía una solicitud HTTP para el objeto almacenado en dicho servidor proxy.

A continuación, el servidor proxy (o caché web) comprueba si tiene una copia del objeto o archivo que queremos recibir, si la tiene, la caché web devuelve el objeto pedido rápidamente. Ver anexo 4.

Si no tiene dicho objeto, la caché web abre una conexión TCP con el servidor de origen para solicitar el objeto, después de recibirlo lo almacena y envía una copia al cliente primero que lo ha solicitado. Ver anexo 3.

Ventajas

- Podemos reducir el tiempo de respuesta a la solicitud de un cliente de forma muy acentuada, sobre todo si el ancho de banda entre el cliente y el servidor de origen es mucho menor que entre el cliente y la caché.
- La caché, al estar instalada de forma local, existe una gran velocidad entre el cliente y la caché web y podrá suministrarle el objeto solicitado rápidamente.
- Reducen el tráfico de salida a internet, así las empresas o instituciones no tienen que mejorar las conexiones de internet y por tanto, los costes son mucho menores que la instalación de dicha caché web. También son útiles para no colapsar tanto las redes de internet y que las aplicaciones que verdaderamente necesiten salir a internet, mejoren sus prestaciones.

2.1.3 Scripts

Es un fichero de texto con una serie de comandos metidos dentro y al que le damos permiso para ejecutarse. De esta manera al ejecutar lo se irán ejecutando los comandos que contiene el fichero.

Un ejemplo sencillo. Imaginemos un script que no es más que el siguiente texto y que tiene permisos de ejecución.

```
#!/bin/bash  
#Un script de ejemplo  
echo "Hola mundo"
```

Si ejecutamos este script, lo único que hará es mostrar en la pantalla el mensaje "Hola mundo".

La primera línea le indica al sistema que es un script y que está pensado para bash. Nosotros lo vamos a poner siempre así, porque siempre vamos a crear scripts para bash.

La segunda línea es un comentario, para que si más tarde editas el script, tengas orientación de que hiciste y como. Es muy recomendable siempre poner comentarios para que cuando editemos los scripts o los pasemos a otros, entiendan que se hizo. Los comentarios son lo que hay en una línea después del #.

Tabla 2: Comandos Básicos Debian

Comando	Descripción
chmod	Da permisos a un archivo
mkdir archivos	Crea un directorio llamado archivos
Init 0	Apaga el equipo
reboot	Reinicia el equipo
echo	Muestra un mensaje
date	Muestra hora y fecha
cd/home	Cambiar al directorio home
cd..	Regresar un nivel
cd../..	Regresar dos niveles
cd	Cambiar al directorio cd
ls	Ver archivos del directorio
rm -f file1	Borra archivo con el nombre file1
rmdir dir1	Borra directorio con el nombre dri1
kill -9 ID_proceso	Forzar el cierre y terminación de un proceso
crontab -e	Edita crontab
crontab -l	Muestra lista de crontabs creados por el usuario
crontab -r	Elimina los crontabs del usuario
vi	Editar un archivo
top	Muestra la lista de procesos activos

2.1.4 Liberar memoria

La memoria RAM es uno de los componentes fundamentales de nuestra PC, y es la que determinará si obtendremos una alta o baja performance en el rendimiento cotidiano de nuestro equipamiento informático.

Figura 1: Memoria RAM



Para mejorar el funcionamiento del equipo, más allá de disponer de memorias RAM de gran capacidad y velocidad de transferencia, también es necesario realizar periódicamente un mantenimiento relacionado a este componente, con el fin de lograr que la memoria trabaje siempre de manera precisa y utilizando todos los recursos de los que dispone.

2.2 Fundamentación Teórica

La implementación del servidor web cache en los laboratorios de la UTEQ, se fundamenta en la siguiente base teórica:

2.2.1 Sistema operativo GNU/LINUX³

GNU/Linux es un sistema operativo libre desarrollado por voluntarios de todo el mundo.

Sus principales características son:

- multitarea: se pueden realizar varias actividades a la vez (navegar por Internet, editar un documento, compilar un programa,...)
- *multiusuario*: varios usuarios pueden trabajar concurrentemente en un único ordenador con varios terminales (teclado y monitor) de forma que tengan la sensación de que es el único que está trabajando en el sistema. Cada usuario almacena sus datos (programas, documentos de texto, imágenes,...) en una cuenta privada o “home”. Notar que para que sea multiusuario es imprescindible que sea multitarea.
- *conectividad*: permite las comunicaciones en red y el acceso a recursos remotamente. Por ejemplo, podemos acceder a nuestros datos situados en una máquina a través de otro equipo, conectados ambos a Internet.
- *multiplataforma*: se puede instalar en multitud de dispositivos, desde todo tipo de ordenadores de sobremesa y portátiles y servidores hasta videoconsolas o incluso teléfonos móviles.
- libre: su código fuente está disponible. Cualquiera puede usarlo, modificarlo y distribuir.

³ <http://www.gnu.org/gnu/gnu-history.html>

GNU/Linux ofrece todo lo necesario para trabajar en red con TCP/IP (el protocolo de Internet). Desde manejadores para las tarjetas de red más populares, PPP (que permite acceder a una red TCP/IP utilizando un módem y la línea telefónica), PPPoE (acceso TCP/IP mediante ADSL), etc.

Y también existen gran cantidad de aplicaciones relacionadas con Internet, como navegadores, clientes de correo, clientes de mensajería instantánea, etc (GNU, 2012).

2.2.1.1 Diseño y filosofía de GNU/LINUX⁴

En un sistema operativo comercial, como puede ser Windows, el software es desarrollado siguiendo una rigurosa política de mantenimiento de la calidad, con métodos de revisiones para las fuentes y documentación.

Los desarrolladores no pueden añadir cosas nuevas por su cuenta: cualquier cambio será en respuesta a un informe de fallo detectado y será cuidadosamente registrado en el sistema de control de versiones, de manera que podrá volverse atrás sin problemas. Cada desarrollador tiene asignada una o más partes de código y sólo ese desarrollador puede alterar esas secciones.

Por el contrario, GNU/Linux ha sido desarrollado principalmente por un grupo de programadores de todo el mundo unidos por Internet. A través de Internet, cualquiera tiene la oportunidad de unirse al grupo y ayudar al desarrollo y depuración de GNU/Linux, portar nuevo software, escribir documentación o ayudar a los nuevos usuarios. Hoy en día, grandes compañías como IBM, SUN, HP, Novell y RedHat, entre otras muchas, aportan a GNU/Linux grandes ayudas tanto económicas como del trabajo de sus propios empleados. El sistema se ha diseñado siguiendo una filosofía abierta y de crecimiento.

⁴ 00.introduccion_a_linux.pdf

Normalmente, GNU/Linux se comparte como un kernel junto con otros programas en lo que se conoce como *distribución*. Existen numerosas distribuciones GNU/Linux. Se diferencian principalmente entre ellas por los programas que incorporan (por ejemplo, existen distribuciones temáticas: de educación, científicas,...) (GNU, 2012).

2.2.1.2 Distribuciones GNU/Linux

Una distribución GNU/Linux es un sistema operativo que reúne todo el software necesario para poner un ordenador a punto de uso.

Está formado por un núcleo del sistema, que son programas que controlan cada uno de los dispositivos/ componentes del ordenador, y un conjunto de aplicaciones diseñadas para los usuarios, como por ejemplo Netscape, Corel Draw o StarOffice.

En la actualidad existe una gran cantidad de distribuciones, en su mayoría creadas para satisfacer las necesidades concretas de colectivos determinados.

Las distintas distribuciones GNU/Linux existentes hacen hincapié en temas tales como la facilidad de uso, la seguridad, las artes gráficas, etc.

Ventajas

- económicas: es gratuito
- privacidad: los programas de GNU/Linux no recopilan información sobre nuestros gustos, historial de navegación, para enviársela a terceras empresas

- seguridad: no se ve afectado por los virus
- robustez: mucho más estable que los sistemas Windows.
- funcionalidad: con las actuales versiones de GNU/Linux, tenemos los programas equivalentes a cualquier distribución comercial. Además, existen aplicaciones científicas disponibles sólo para GNU/Linux.
- desarrollo: GNU/Linux cuenta con numerosas herramientas para el desarrollo de programas (GNU, 2012).

2.2.2 Herramientas para implementar el servidor

2.2.2.1 Debian

Debian GNU/Linux es un sistema operativo libre, desarrollado por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet.

La dedicación de Debian al software libre, su base de voluntarios, su naturaleza no comercial y su modelo de desarrollo abierto la distingue de otras distribuciones del sistema operativo GNU. Todos estos aspectos y más se recogen en el llamado Contrato Social de Debian.

Nació en el año 1993, de la mano del proyecto Debian, con la idea de crear un sistema GNU usando Linux como núcleo ya que el proyecto Debian, organización responsable de su mantenimiento en la actualidad, también desarrolla sistemas GNU basados en otros núcleos (Debian GNU/Hurd, Debian GNU/NetBSD y Debian GNU/kFreeBSD).

Uno de sus principales objetivos es separar en sus versiones el software libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Deben no vender directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respeta su licencia.

2.2.2.2 Thundercache⁵

ThunderCache es un poderoso sistema de Web Caché que tiene como principal característica hacer caching de contenidos de datos estáticos y dinámicos en la web, incluyendo videos on-line y actualizaciones de windows y antivirus, prestaciones que no brindan los proxys regulares.

La gente utiliza cada día más servicios on-line y la gran mayoría de ellos a través de páginas web (documentos, trámites, negocios, consultas, email, información, etc.).

Con ThunderCache instalado en la red, se acelera la navegación de los usuarios, se optimiza el uso de la conexión a internet, se reducen los costos operativos y se transforma la experiencia de uso de internet en todo su dinamismo.

2.2.2.3 Putty

Putty, es un cliente de acceso remoto a máquinas informáticas de cualquier tipo mediante SSH, Telnet o RLogin, para plataformas Windows 32bits y UNIX. Esta

⁵ <http://thundercache.sislandsoft.com/acerca-de-thundercache/-que-es-thundercache>

herramienta, será útil en el momento que deseemos acceder a otros sistemas que sean o no compatibles con el formato que estamos ejecutando.

Por ejemplo, si estamos en una máquina con un sistema Windows y deseamos acceder al prompt de otra máquina con sistema UNIX de nuestra red local, Putty nos lo permitirá.

2.2.2.4 Winbox

Winbox es una pequeña utilidad que permite la administración de Mikrotik RouterOS usando una interfaz gráfica de usuario fácil y simple. Es un binario Win32 nativo, pero se puede ejecutar en Linux y Mac OSX usando Wine.

Todas las funciones de la interfaz winbox están tan cerca como sea posible a la consola funciona, es por eso que no hay secciones winbox en el manual.

Algunos de avanzada y configuraciones importantes del sistema que no son posibles de winbox, como cambio de dirección MAC de una interfaz.

2.2.2.5 Squid y thundercache⁶

THUNDER CACHE es un sistema que trabaja en conjunto con Squid, reescribiendo las direcciones URL, haciendo cache de los archivos de video como de youtube, y actualizaciones de antivirus y windows update, esto, para mejorar el rendimiento de un sistema de red local.

Es un módulo ideal para los proveedores de servicios de Internet, ya que además de dar un rendimiento excepcional, proporciona un notable ahorro de ancho de banda (PERUFW, 2012).

⁶ <http://perufw.wordpress.com/2010/04/22/instalando-thunder-cache-3-en-ubuntu/>

2.2.2.6 Cacti en thundercache

Es una solución para generar gráficos en red. Esta herramienta, desarrollada en PHP, provee un pooler ágil, plantillas de gráficos avanzadas, múltiples métodos para la recopilación de datos, y manejo de usuarios. Tiene una interfaz de usuario fácil de usar, que resulta conveniente para instalaciones del tamaño de una LAN, así como también para redes complejas con cientos de dispositivos (PERUFW, 2012).

2.2.3 Herramientas para automatizar scripts

2.2.3.1 ¿Qué es cron?

Cron es el planificador de procesos incluido en prácticamente todos los Unix. Es un pequeño *demonio* que se encarga de ejecutar los procesos descritos en el fichero crontab cuando corresponda.

Por defecto, *cron* se ejecuta en segundo plano una vez cada minuto y revisa la tabla de tareas que debe de ejecutar (si las hay).

Cron permite ejecutar programas como "*daemons*" sin serlo realmente, ya que es él el que se encarga de realizar la ejecución periódica (GNU, 2012).

2.2.3.2 ¿Qué es crontab?

Crontab es un pequeño fichero de texto, situado en */etc/crontab* y que únicamente contiene una pequeña configuración para cada uno de los procesos que se deben de ejecutar.

Dicho fichero está distribuido por líneas y cada línea tiene la fecha en la que se debe de repetir (frecuencia de ejecución), el usuario con el que hacerlo (permisos) y el comando a ejecutar.

En el fichero *crontab* se puede especificar los comandos que desea y la periodicidad de los mismos. La única diferencia es que no hay campo usuario, ya se asume que es el propio usuario dueño de dicho fichero. Cualquier usuario puede generar y manejar su propio fichero con el comando *crontab* (GNU, 2012).

Tareas en crontab

Una de las partes más esenciales de *cron*, como ya dijimos es *crontab* y dentro de éste fichero, lo más lógico es saber cómo crear una nueva línea. Recordemos que cada línea es una tarea o trabajo, así que una nueva línea es un nuevo trabajo que se ejecutará.

Tabla 3: Formato Crontab

Actualizar sistema a las 3:15 todos los días	1 5 3 * * * root apt-get update
Actualizar sistema todos los domingos a las 10:00	00 10 * * 0 root apt-get -y update
Actualizar sistema el día 15 de cada mes a las 15:15	15 15 15 * * root apt-get update
Actualizar sistema todos los días de la semana a las 15:15	15 15 * * 1,2,3,4,5 root apt-get update
Actualizar sistema el día 1, 15 y 30 de cada mes	00 00 1,15,30 * * root apt-get update

--	--

2.3 Marco Legal

2.3.1 Decreto 1014 software libre ecuador

El día jueves 10 de Abril del 2008 se emitió el decreto 1014 por parte de la presidencia del Ec. Rafael Correa Delgado que promueve el uso de software libre en las instituciones públicas del Ecuador.

Art. 1: Establecer como política pública para las entidades de administración Pública central la utilización del Software Libre en sus sistemas y equipamientos informáticos.

Art. 2: Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas. Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna.
- Estudio y modificación de programa (Requisito: código fuente disponible).
- Publicación del programa mejorado (Requisito: código fuente disponible).

Art. 3: Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para este tipo de software.

Art. 4: Se faculta la utilización de software propietario(no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando esté en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno (Decreto 1014, 2008).

2.3.2 Ley Orgánica De Educación Superior

Art. 32.- Programas informáticos.- Las empresas que distribuyan programas informáticos tienen la obligación de conceder tarifas preferenciales para el uso de las licencias obligatorias de los respectivos programas, a favor de las instituciones de educación superior, para fines académicos.

Las instituciones de educación superior obligatoriamente incorporarán el uso de programas informáticos con software libre (Ley Orgánica de Educación Superior, 2010).

2.4 Fundamentación Referencial

Joomla es un Sistema de gestión de contenidos (en inglés Content Management System, o CMS) que permite desarrollar sitios web dinámicos e interactivos. Permite crear, modificar o eliminar contenido de un sitio web de manera sencilla a través de un *Panel de Administración*.

Es un software de código abierto, desarrollado en PHP y liberado bajo licencia GPL. Este administrador de contenidos puede utilizarse en una PC local (en Localhost), en una Intranet o a través de Internet y requiere para su

funcionamiento una base de datos creada con un gestor MySQL, así como de un servidor HTTP Apache.

La caché de página es aquella en la que Joomla hace una copia de una página completa cuando ésta se muestra por primera vez.

Si esta página es visitada de nuevo, simplemente se toma esta copia que se guardó y se vuelve a mostrar, evitando la mayoría del código y muchas de las consultas a la base de datos necesarias para construirla desde cero de nuevo.

La implementación del servidor web cache también tiene como objetivo hacer copias de los objetos web para responder a las peticiones de los usuarios parecido a la Cache de página de Joomla. Adicionando a esta investigación los scripts para la instalación y eliminación del tráfico zombie del servidor.

CAPÍTULO III

METODOLOGIA DE LA INVESTIGACION

3.1 Materiales y métodos

3.1.1 Métodos y técnicas

Para el desarrollo de la investigación, por su carácter tecnológico, científico, se utilizaron los siguientes métodos de investigación.

3.1.1.1. Método Inductivo

El método inductivo permitió obtener conclusiones generales a partir de casos, hechos o fenómenos particulares, para identificar las herramientas y recursos necesarios para la elaboración del proyecto, desde la comprobación de hipótesis y la exposición de la tecnología a implantarse.

3.1.1.2. Método Deductivo

Este método permite representar conceptos o principios generales como la automatización del proceso de mantenimiento del servidor, para extraer las conclusiones o consecuencias que se derivan del mismo.

3.1.2 Materiales y presupuesto

A continuación detallaremos los requerimientos para la implementación de este proyecto en los siguientes ámbitos, hardware, software, suministro, presupuesto.

Tabla 4: Hardware requerido Autor: Mora Andrade Stephanie.			
Detalle	Descripción	Costo unitario	Total
Computador	Computador de escritorio del sistema (Core dos duo, 3 GB RAM, 2 Disco Duro de 250Gb, DVD RE-Writer, pantalla 16.1").	\$ 600.00	\$ 600.00
Impresora hp laserjet p1505n	Toner	\$ 200.00	\$ 200.00
Routerboard	Mikrotik 150 Gl	\$ 90.00	\$ 90.00
Switch	QPcom 8 ports	\$ 25.00	\$ 25.00
			\$915.00

Tabla 5: Software requerido. Autor: Mora Andrade Stephanie.		
Detalle	Costo unitario	Total
Linux/ Devitan	\$ 0.00	\$ 0.00
Putty	\$ 0.00	\$ 0.00
WinSock	\$ 0.00	\$ 0.00

Thundercache 3.1.2	\$ 0.00	\$ 0.00
		\$0.00

Tabla 6: Suministros requeridos. Autor: Mora Andrade Stephanie.			
Ítem	Detalle	Costo unitario	Total
4	Resmas de hojas A4	\$ 3.50	\$ 14.00
4	Lapiceros	\$ 0.35	\$ 1.40
1	Disco externo 1T	\$ 120.00	\$120.00
	Gastos varios	\$ 50.00	\$ 50.00
			\$ 185.40

Tabla 7: Presupuesto del proyecto. Autor: Mora Andrade Stephanie.	
Tipo de Recurso	Costo Total
Hardware	\$915.00

Software	\$ 0.00
Suministros	\$ 185.40
Total	\$ 1100.40

3.1.3 Factibilidad

3.1.3.1 Recursos y Financiamiento

La tecnología web cache se está utilizando para mejorar el rendimiento de las redes informáticas, pero los costos de implementación resultan ser algo excesivos, y no todas las personas o empresas están dispuestas a utilizar esta tecnología aun sabiendo las grandes ventajas que brinda.

Esta investigación tiene un buen grado de factibilidad fomentando la utilización de servidores web cache, al orientar en el proceso de creación y mantenimiento de dicho servidor, aportando a la disminución de costos para de esta forma hacerla más accesible ante la sociedad interesada.

Entre otras ventajas de la investigación se encuentran que el investigador del proyecto no tendrá que realizar gastos de contratación de personal, se cuenta con el hardware y software requerido para implementarlo.

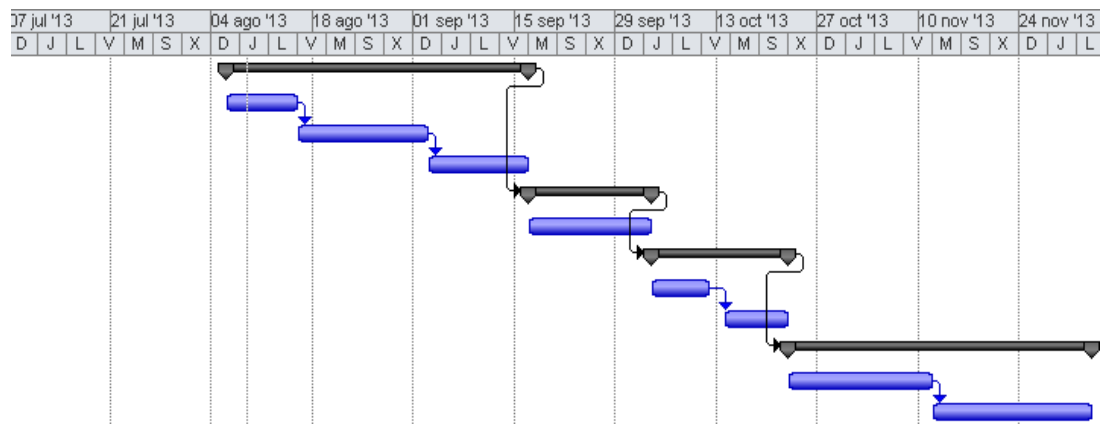
Se posee la impresora para elaborar la documentación, y el software a utilizar es libre y de fácil acceso prescindiendo de la compra de licencias que podrían resultar muy costosas.

3.1.4 Cronograma

Figura 2: Cronograma de actividades.

Autor: Mora Andrade Stephanie.

	Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos
	Análisis de requisitos	30 días	mar 06/08/13	lun 16/09/13	
	Analisis de hardware	8 días	mar 06/08/13	jue 15/08/13	
	Analisis scripts	12 días	vie 16/08/13	lun 02/09/13	
	Analisis de software	10 días	mar 03/09/13	lun 16/09/13	
	Diseño	13 días	mar 17/09/13	jue 03/10/13	
	Diseño scripts	13 días	mar 17/09/13	jue 03/10/13	
	Construcción	13 días	vie 04/10/13	mar 22/10/13	
	construccion scripts	6 días	vie 04/10/13	vie 11/10/13	
	construccion servidor	7 días	lun 14/10/13	mar 22/10/13	
	Implementación	30 días	mié 23/10/13	mar 03/12/13	
	Implementacion servidor	14 días	mié 23/10/13	lun 11/11/13	
	Implementación y automatización de scripts	16 días	mar 12/11/13	mar 03/12/13	



3.2 Tipos de Investigación

3.2 .1 Investigación cuasi-experimental

Por el tipo de investigación que se realizará, se considera apropiado el método cuasi-experimental, de un solo grupo con medidas pretexto y postes, es decir, con una medición antes y después de implementar la aplicación.

En la investigación cuasi-experimental se identifica la variable independiente mediante la letra X. La variable dependiente es designada con la letra O.

Si existe una medida antes del tratamiento la denominaremos pretexto, cuando la medida se realiza tras el tratamiento la denominamos postes.

Su notación es: **O1 X O2**

Dónde:

X = Servidor web cache.

O1 = Medición antes del experimento (Laboratorios con la tecnología actual).

O2 = Medición después del experimento (Laboratorios con tecnología web cache).

3.2.2 Investigación de campo

La investigación se la hará mediante técnicas de campo con uso de herramientas como la entrevista, observación directa; etc.

3.3 Diseño de la Investigación

3.3.1 Técnicas de Investigación

Las técnicas de investigación que se utilizarán en este proyecto de tesis son entrevistas y observar directamente a los involucrados, siendo esta una forma óptima para poder llegar a la exactitud del problema, y poder mejorarlo con la implementación del servidor web cache.

3.3.2 Encuesta

Esta técnica sirvió para explorar toda la información que se pretende investigar; se aplicaron encuestas a los estudiantes, sobre la realidad de la problemática. Se basará, exclusivamente en un cuestionario que tiene plena relación con el planteamiento de la hipótesis.

3.4 Población Y Muestra

3.4.1 Población

La población tomaremos para esta investigación es los estudiantes que utilizan los laboratorios de la unidad de tics, que finalmente serán los beneficiarios de la implementación de este proyecto.

Estos laboratorios de la UTEQ son visitados por muchos estudiantes diariamente tomaremos como población a 100 personas para seguir con nuestra investigación.

3.4.2 Muestra

Variables:

N = Población

n = Tamaño de la muestra

E = Error tolerable de muestreo

K = Coeficiente de corrección del error

V = Constante de varianza

Ecuación.

$$n = \frac{V \cdot N}{\frac{E^2 (N-1)}{K^2} + V}$$

3.4.3 Aplicación de la muestra

$N = 100$

$E = 10\% = 0.1$

$K = 2$

$V = 0.25$

$$n = \frac{0.25 * 100}{\frac{0.1^2 (100 - 1)}{2^2} + 0.25}$$

$$n = \frac{25}{\frac{0,99}{4} + 0.25}$$

$$n = \frac{25}{0,4975}$$

$$n = 50,25$$

$$n = 50$$

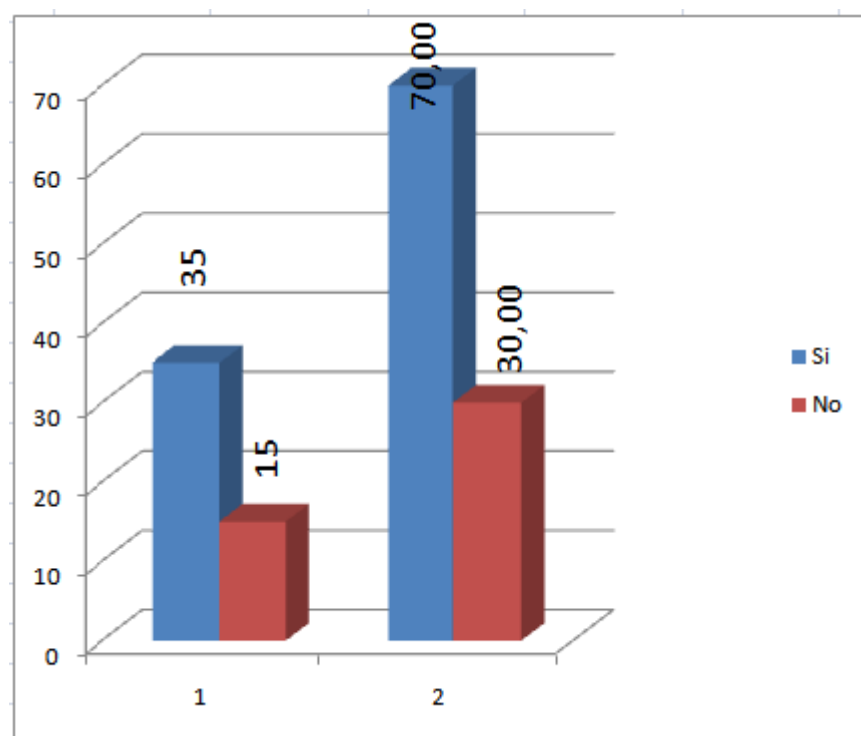
3.4.4 Cuestionario para encuesta dirigida a los estudiantes de los laboratorios de TICS.

1. ¿Con que frecuencia Ud. usa los laboratorios de tics?
2. ¿Ud. usa los laboratorios de tics por motivos académicos?
3. ¿Ha tenido Ud. inconvenientes en algún laboratorio de la unidad de tics?
4. ¿Cree usted que un sistema web cache mejorará el acceso a internet en los laboratorios de tics?
5. Cree usted que un sistema web cache reducirá el consumo de ancho de banda en los laboratorios de la unidad de tics?
6. Cree usted que un sistema web cache reducirá los accesos al enlace WAN en los laboratorios de la unidad de tics.
7. ¿Cree usted que un sistema web cache reducirá el tiempo que los estudiantes esperan en que se realicen sus descargas?
8. ¿Cree usted que un sistema web cache le brindará a los usuarios una mayor velocidad de descarga?
9. ¿Cree Ud. que el sistema web cache es una herramienta eficiente que se debe implementar en los laboratorios de la unidad de tics en la UTEQ??

3.4.5 Análisis e interpretación de resultados

Tabla 8: Pregunta 1

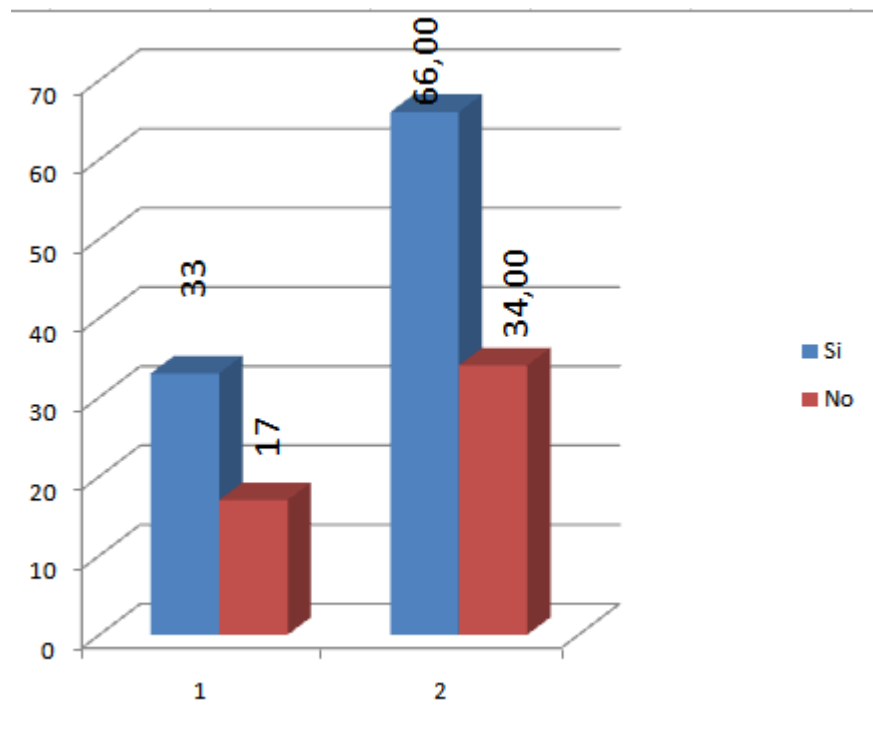
1. ¿Con que frecuencia Ud. usa los laboratorios de tics?		
Respuesta	Cantidad	Porcentaje
Muy frecuente	35	70%
Poco frecuente	15	30%
Total	50	100%



Análisis e interpretación: El 70% de los estudiantes que fueron tomados como muestra han dicho que usan de forma muy frecuente los laboratorios de la institución. Siendo este un dato de mucha relevancia que indica que hay un alto índice de usuarios que se beneficiarán del proyecto que se desea implementar.

Tabla 9: Pregunta 2

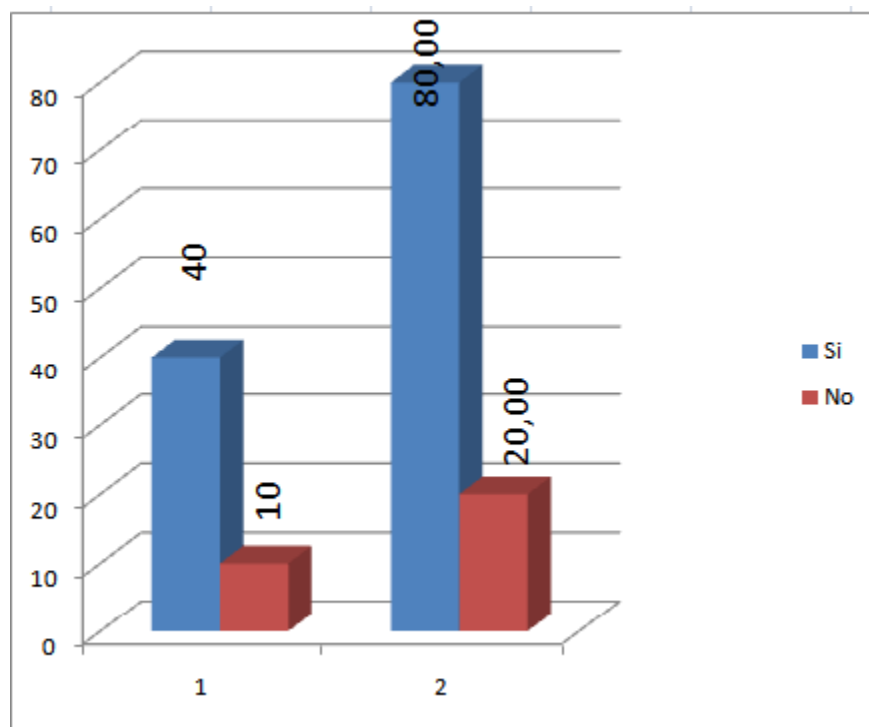
2. ¿Ud. usa los laboratorios de tics por motivos académicos?		
Respuesta	Cantidad	Porcentaje
Si	33	66%
No	17	34%
Total	50	100%



Análisis e interpretación: De la muestra tomada notamos que el 66% de los estudiantes usan los laboratorios institucionales para cumplir sus labores académicas. Esta referencia nos revela que la implementación del servidor web cache incide a mejorar el desarrollo de las tareas académicas que realizan en dichos laboratorios.

Tabla 10: Pregunta 3

3. ¿Ha tenido Ud. inconvenientes en algún laboratorio de la unidad de tics?		
Respuesta	Cantidad	Porcentaje
Si	40	80%
No	10	20%
Total	50	100%

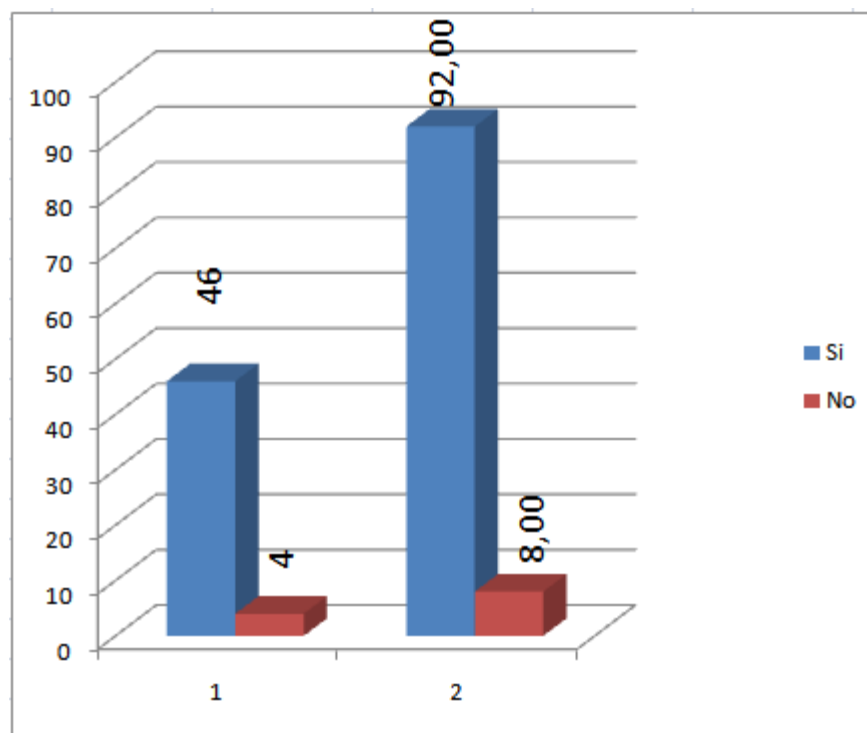


Análisis e interpretación: De la muestra se ha reflejado que el 80% de los encuestados han sufrido percances al utilizar los laboratorios de la unidad de tics.

Estos datos nos indican que el desarrollo del proyecto disminuirá los inconvenientes que experimentan los estudiantes que cumplen con las tareas académicas de la institución.

Tabla 11: Pregunta 4

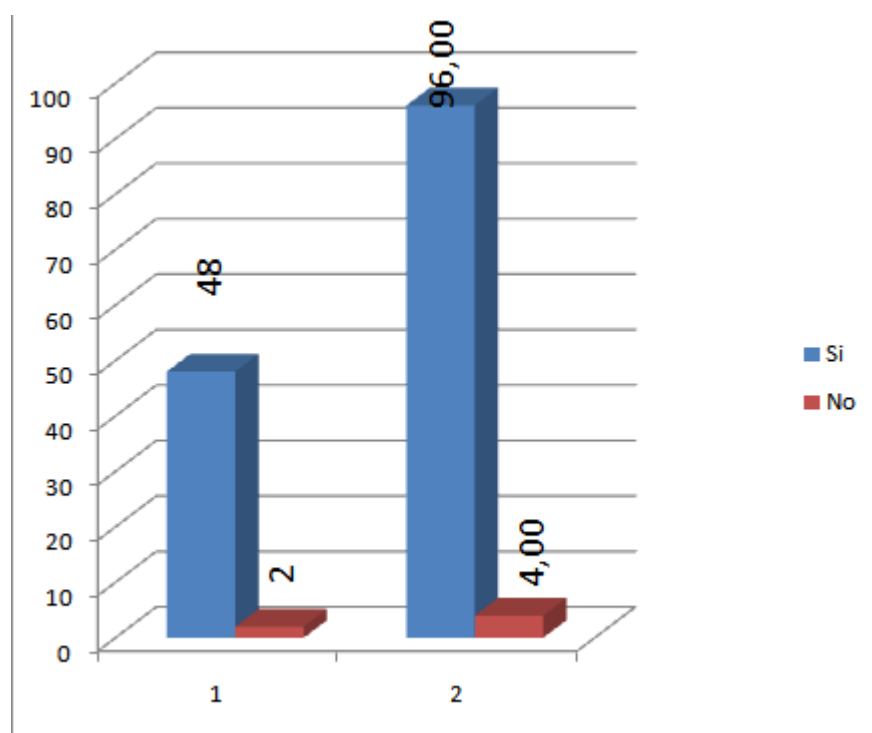
4. ¿Cree usted que un sistema web cache mejorará el acceso a internet en los laboratorios de tics?		
Respuesta	Cantidad	Porcentaje
Si	46	92%
No	4	8%
Total	50	100%



Análisis e interpretación: El 92% de la población tomada como muestra dice que un servidor web cache en de la unidad de tics incurre en mejorar el rendimiento de la red de laboratorios. Mejorando la interacción entre los usuarios y los recursos disponibles en la web.

Tabla 12: Pregunta 5

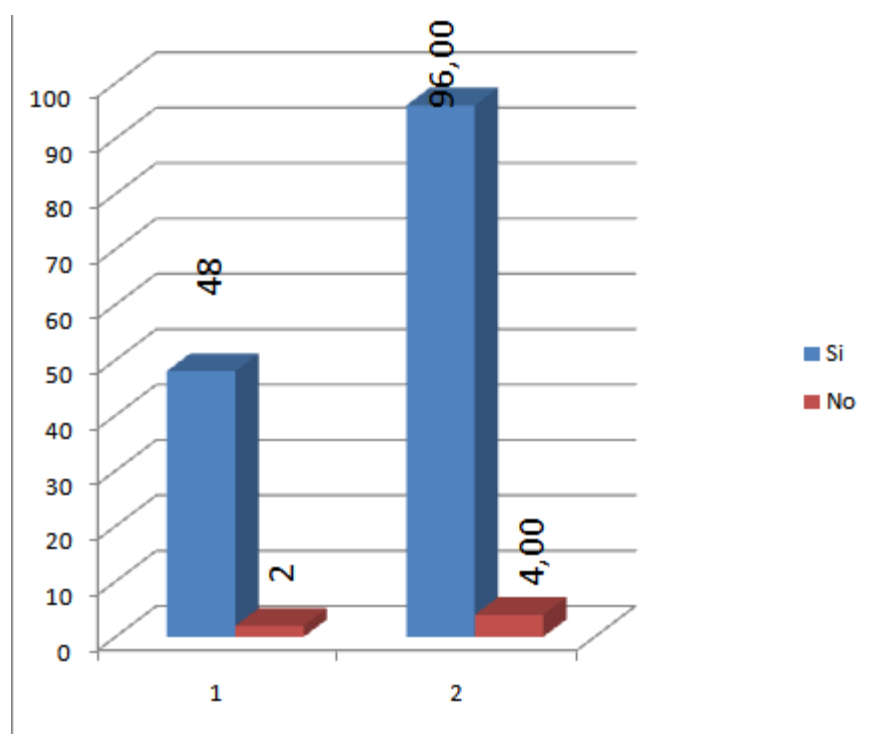
5. Cree usted que un sistema web cache reducirá el consumo de ancho de banda en los laboratorios de la unidad de tics?		
Respuesta	Cantidad	Porcentaje
Si	48	96%
No	2	4%
Total	50	100%



Análisis e interpretación: De los encuestados obtuvimos que el 96% de ellos creen que la implementación de un servidor en los laboratorios favorecería a la reducción del consumo de ancho de banda beneficiando a los usuarios y a la institución.

Tabla 13: Pregunta 6

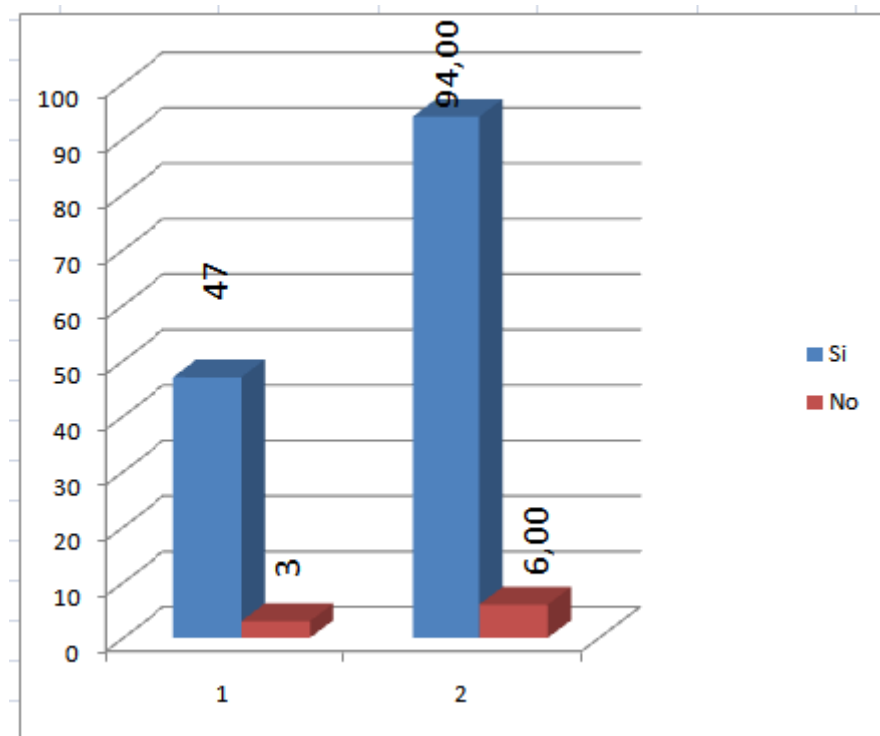
6. Cree usted que un sistema web cache reducirá los accesos al enlace WAN en los laboratorios de la unidad de tics?		
Respuesta	Cantidad	Porcentaje
Si	48	96%
No	2	4%
Total	50	100%



Análisis e interpretación: El 98% de la muestra tomada mencionó que un servidor cache implementado en las redes de los laboratorios de la institución, incide en reducir el número de peticiones que salen a la web, al ser el servidor quien tienen la prioridad de responder a objetos web solicitadas.

Tabla 14: Pregunta 7

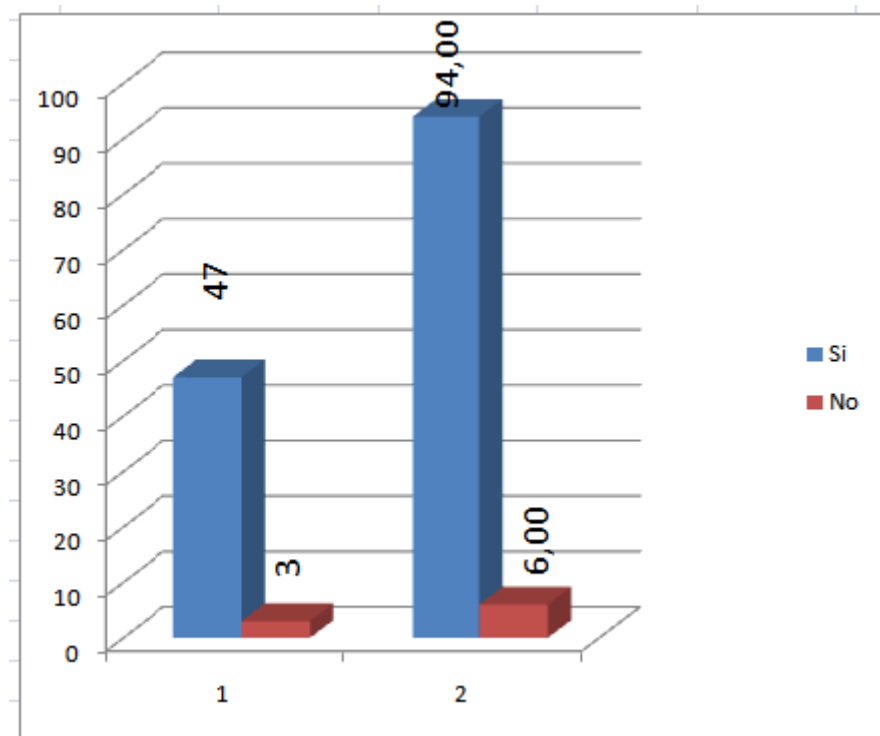
7. ¿Cree usted que un sistema web cache reducirá el tiempo que los estudiantes esperan en que se realicen sus descargas?		
Respuesta	Cantidad	Porcentaje
Si	47	94%
No	3	6%
Total	50	100%



Análisis e interpretación: El 94% de los encuestados nos mencionan que la tecnología web cache tiene un alto grado de incidencia en la reducción del tiempo de espera por parte de los usuarios en cada descarga realizada, debido a ser el servidor el que responde con objetos web guardados en su cache.

Tabla 15: Pregunta 8

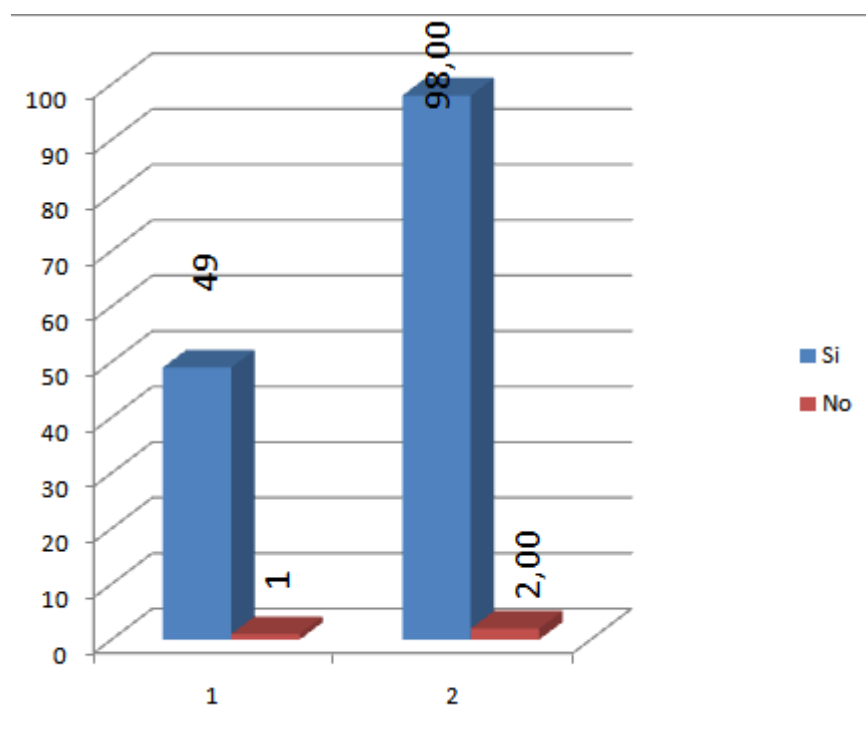
8. ¿Cree usted que un sistema web cache le brindará a los usuarios una mayor velocidad de descarga?		
Respuesta	Cantidad	Porcentaje
Si	47	94%
No	3	6%
Total	50	100%



Análisis e interpretación: El 94% de la muestra correspondiente cree que sí el servidor web cache favorece a los usuarios para poder realizar descargas más rápidas debido a la gran velocidad de transferencia que existe entre la cache y el usuario.

Tabla 16: Pregunta 9

9. ¿Cree Ud. que el sistema web cache es una herramienta eficiente que se debe implementar en los laboratorios de la unidad de tics en la UTEQ??		
Respuesta	Cantidad	Porcentaje
Si	49	98%
No	1	2%
Total	50	100%



Análisis e interpretación: El 98% de nuestros encuestados respondió que la tecnología web cache es una herramienta eficiente que ayuda a optimizar el rendimiento de las redes en los laboratorios, siendo un instrumento que debe ser acogido e implementado para mejorar el servicio que brinda la institución a sus alumnos.

CAPITULO IV

RESULTADOS Y DISCUSION

4.1 Desarrollo del proyecto

El desarrollo a la propuesta de implementar un servidor web cache en los laboratorios de tics en la UTEQ será ejecutada de acuerdo a los siguientes procesos:

4.1.1 Instalación y configuración del sistema operativo Debian

Se requiere un computador corre 2 dúo, e ge de RAM, 2 discos duros unos para el sistema operativo y otra para guardar la cache del servidor, el computador deberá estar equipado con unidad de DVD. Ver anexo 1.

Tener el CD o descargar la imagen ISO del sistema Debian 6.

4.1.2 Instalación del sistema de caching Thundercache

Una vez que poseemos el hardware necesario y la plataforma especificada, podemos comenzar con la instalación de Thundercache 3.1 y realizar la instalación y sus respectivas configuraciones detalladas a continuación.

Figura 3: Comandos instalación thundercache

Es importante que durante la instalación del thunder poner la contraseña del Mysql: **thunder31 32 bits**

Código:

```
1. wget http://www.joemg.host56.com/tc/thunder312-32.sh
2. chmod 777 thunder312-32.sh
3. ./thunder312-32.sh
```

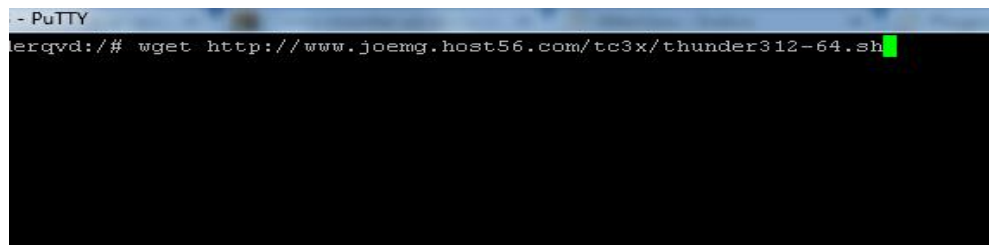
64 bits

Código:

```
1. wget http://www.joemg.host56.com/tc/thunder312-64.sh
2. chmod 777 thunder312-64.sh
3. ./thunder312-64.sh
```

Estos comandos son necesarios para empezar la instalación y configuración de thundercache se deberá utilizar el de 64 bits ya que hay se define que el cache se instalará en un disco duro ya que el otro posee al sistema operativo.

Figura 4: Ingreso de comandos thundercache.



Después de haber realizado los pasos anteriores conectamos el disco duro que será el cache, ingresamos con la terminal Putty como root al sistema y le ponemos los comandos siguientes, que son los encargados de añadir el nuevo disco y de darle permisos de lectura y escritura.

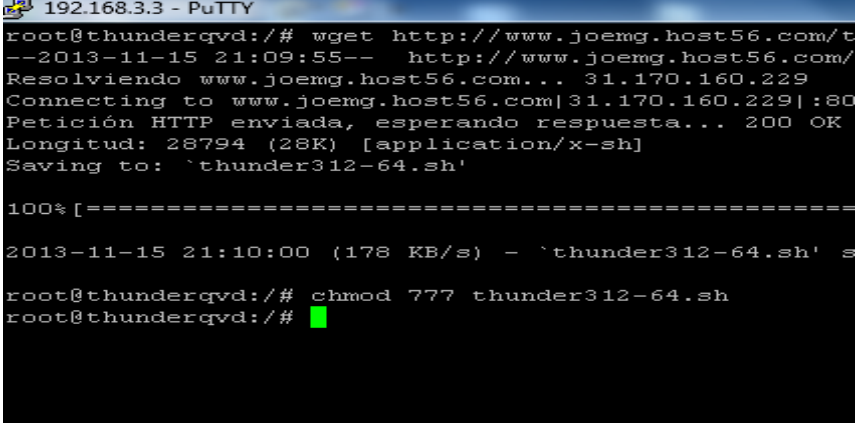
`fdisk -l`

`wget http://www.joemg.host56.com/tc/tcmdisk.sh`

`chmod 777 tcmdisk.sh`

`./tcmdisk.sh`

Figura 5: Permisos de lectura.



```
192.168.3.3 - PuTTY
root@thunderqvd:/# wget http://www.joemg.host56.com/t
--2013-11-15 21:09:55-- http://www.joemg.host56.com/
Resolviendo www.joemg.host56.com... 31.170.160.229
Connecting to www.joemg.host56.com|31.170.160.229|:80
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 28794 (28K) [application/x-sh]
Saving to: `thunder312-64.sh'

100%[=====]
2013-11-15 21:10:00 (178 KB/s) - `thunder312-64.sh' s
root@thunderqvd:/# chmod 777 thunder312-64.sh
root@thunderqvd:/#
```

Figura 6: Instalación squid.

```
192.168.3.3 - PuTTY
|-----|
|                               |
|                               |
|                               |
|-----+-----+
|          INSTALACION THUNDER 3.1.2 - Squid 3.x          |
|                               |
|-----+-----+
|-----> Version 64 bits <-----|
|
/
Obj http://cdn.debian.net squeeze Release.gpg
Ign http://cdn.debian.net/debian/ squeeze/main Translation-en
Ign http://cdn.debian.net/debian/ squeeze/main Translation-es
Obj http://security.debian.org squeeze/updates Release.gpg
Ign http://security.debian.org/ squeeze/updates/main Translation-en
Ign http://security.debian.org/ squeeze/updates/main Translation-es
Obj http://cdn.debian.net squeeze-updates Release.gpg
Ign http://cdn.debian.net/debian/ squeeze-updates/main Translation-en
Ign http://cdn.debian.net/debian/ squeeze-updates/main Translation-es
Obj http://cdn.debian.net squeeze Release
Obj http://security.debian.org squeeze/updates Release
Obj http://cdn.debian.net squeeze-updates Release
Obj http://cdn.debian.net squeeze/main Sources
Obj http://cdn.debian.net squeeze/main amd64 Packages
Obj http://security.debian.org squeeze/updates/main Sources
Obj http://cdn.debian.net squeeze-updates/main Sources/DiffIndex
Obj http://cdn.debian.net squeeze-updates/main amd64 Packages/DiffIndex
Obj http://security.debian.org squeeze/updates/main amd64 Packages
Obj http://cdn.debian.net squeeze-updates/main amd64 Packages
Leyendo lista de paquetes... Hecho
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Seguimos con la instalación de squid que trabaja en conjunto con thundercache.

Figura 7: Usuario de mysql.

```
192.168.3.3 - PuTTY
|-----+-----+
|          Instalacion del Mysql          |
| Es importante que ponga la contraseña del Mysql = thunder31 |
|-----+-----+
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl liblprc-perl mysql-client-5.1 mysql-server-5.1 mysql-server-core-5.1
Paquetes sugeridos:
 libipc-sharedcache-perl libterm-readkey-perl tinyca
Se instalarán los siguientes paquetes NUEVOS:
 libdbd-mysql-perl libdbi-perl libhtml-template-perl libnet-daemon-perl liblprc-perl mysql-client mysql-client-5.1 mysql-server mysql-server-5.1
mysql-server-core-5.1 php5-mysql
0 actualizados, 11 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 22,3 MB de archivos.
Se utilizarán 57,0 MB de espacio de disco adicional después de esta operación.
Ok [Conectando a cdn.debian.net] [Conectando a security.debian.org]
```

Se instala ahora mysql que es otro componente de thundercache habrá que definir el usuario y contraseña para este paso.

Figura 8: Contraseña mysql.

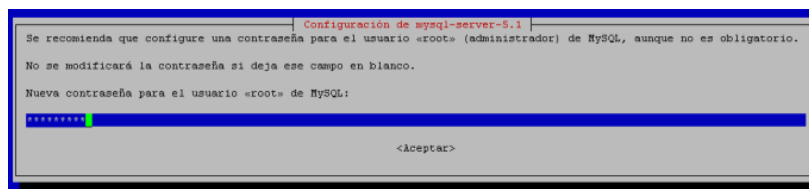
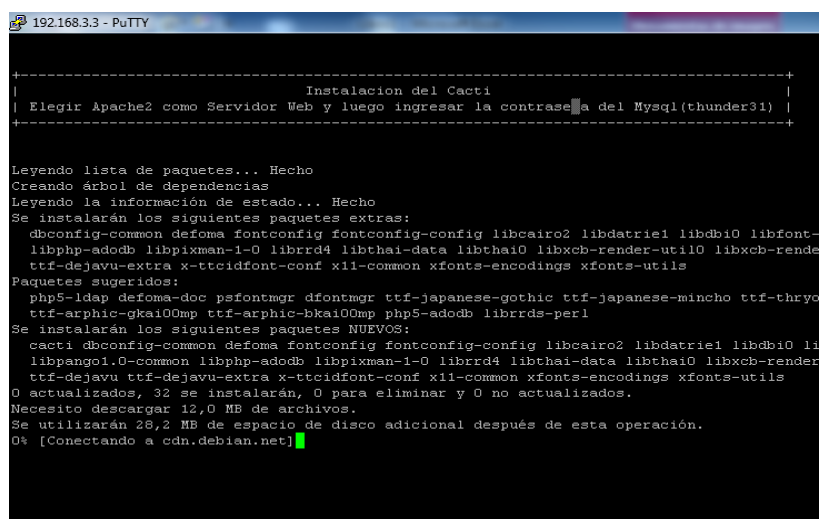


Figura 9: Contraseña cacti.



Otro recurso que también se utiliza es cacti que es una solución para generar gráficos de consumo de red diseñado para tener una buena administración de la red.

Figura 10: Instalación Thundercache finalizada.



En este momento se ha terminado la instalación y configuración de componentes.

Figura 11: Formateo disco.

[illegible]

Se debe formatear el disco duro que será utilizado por el thundercache para que pueda almacenar la información.

Figura 12: Disco agregado thundercache listo.

```
192.168.0.30-P0111
PREPARANDO DISCO DURO: [ /dev/sdb ]

QUE ES LO QUE QUIERES HACER CON EL DISCO DURO: /dev/sdb? ['F'ormatear/'M'ontar/'I'gnorar]: f
1+0 records in
1+0 records out
1048576 bytes (1,0 MB) copied, 0,00507913 s, 206 MB/s
mke2fs 1.41.12 (17-May-2010)
fs_types for mke2fs.conf resolution: 'ext4', 'default'
Calling BLKDISCARD from 0 to 2000398934016 failed.
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=4096 (bitácora=2)
Tamaño del fragmento=4096 (bitácora=2)
Stride=1 blocks, Stripe width=0 blocks
122101760 nodos-i, 488378646 bloques
488378 bloques (0.10%) reservados para el superusuario
Primer bloque de datos=0
Número máximo de bloques del sistema de ficheros=4294967296
14905 bloque de grupos
32768 bloques por grupo, 32768 fragmentos por grupo
8192 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848

Escribiendo las tablas de nodos-i: hecho
Creating journal (32768 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

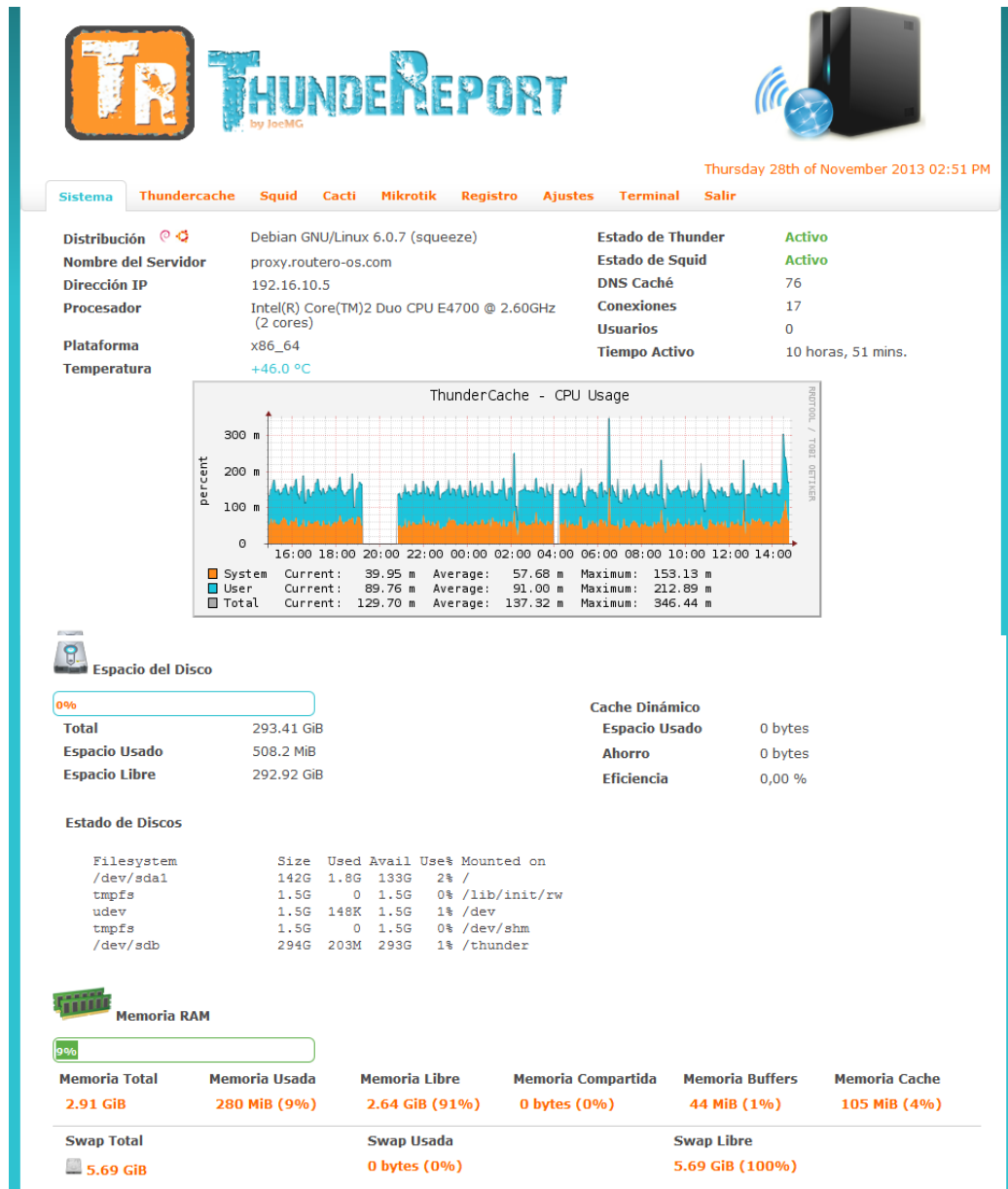
Este sistema de ficheros se revisará automáticamente cada 37 montajes o
180 días, lo que suceda primero.  Utilice tune2fs -c o -i para cambiarlo.

/thunder

+=====+
|                DISCO DURO AGREGADO                |
|                REINICIE SU SISTEMA                  |
+=====+
```

Luego de formatear el disco ya podemos agregarlo y listo estará listo para hacer cache en el disco.

Figura 13: Interfaz gráfica thundercache



Habiendo seguido todos los pasos anteriores entrando por el navegador a la interfaz de administración del thundercache veremos espacio en el disco duro memoria RAM utilizada y muchos datos importantes para el administrador del equipo.

Figura 14: Reporte del cache y sus estadísticas

Thundercache					
Plugins		On - Off			
Ico	Dominios	Archivos	Tamaño	Economía	Hits
4	(http://www.4shared.com) 4shared	3	11.9 MiB	6.84 MiB	1
	(http://www.adobe.com) adobe	7	10.09 MiB	2 bytes	1
	(http://www.avira.com) avira	181	25.54 MiB	3.52 GiB	27.851
	(http://www.static.org) blogspot	612	14.62 MiB	9.65 MiB	1.328
	(http://www.static.org) facebook_photos	3	878 bytes	0 bytes	0
	(http://www.static.org) facebook_profile	76.436	267.51 MiB	679.91 MiB	219.127
	(http://www.static.org) facebook_video	2	12.05 MiB	0 bytes	0
	(http://www.static.org) flickr	1	6.49 KiB	0 bytes	0
	(http://www.static.org) goear	1	4.4 MiB	0 bytes	0
	(http://www.static.org) google_updates	2	56.27 MiB	1.87 GiB	68
	(http://www.keezmovies.com) keezmovies	2	84.12 MiB	0 bytes	0
	(http://www.macromedia.com) macromedia	2	18.88 MiB	0 bytes	0
	(http://www.static.org) microsoftupdate	1	6.63 MiB	0 bytes	0
	(http://www.photobucket.com) photobucket	116	18.15 MiB	2.72 MiB	73
	(http://www.pop6.com) pop6	2	1.1 MiB	0 bytes	0
	(http://www.redtube.com) redtube	3	116.12 MiB	0 bytes	0
	(http://www.sendspace.com) sendspace	1	1.11 MiB	2.23 MiB	2
	(http://www.static.org) softnyx	19	188.67 MiB	596.86 MiB	53
	(http://www.static.org) softnyx_gunbound	1	1.13 MiB	0 bytes	0
	(http://www.static.org) softnyx_rakion	26	51.13 MiB	693.61 MiB	461
	(http://www.static.org) softnyx_wolf	20	47.03 MiB	43.03 MiB	18
	(http://www.softonic.com) softonic	5	30.99 MiB	62.94 MiB	8
	(http://www.sourceforge.com) sourceforge	1	2.87 MiB	0 bytes	0
	(http://www.speedtest.net) speedtest	133	2.35 MiB	2.86 MiB	292

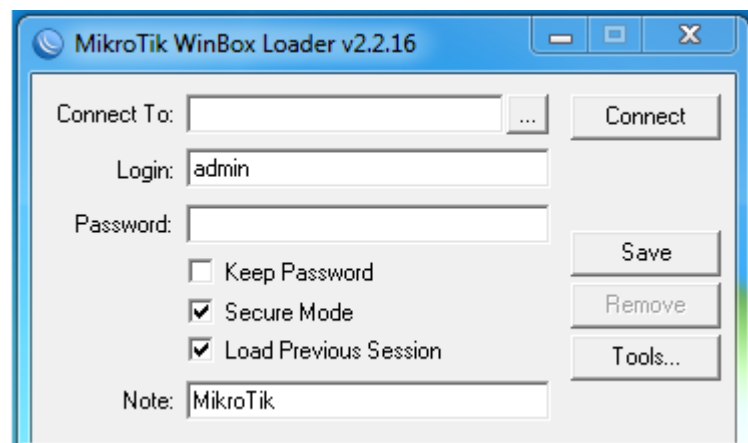
Entre la utilidades que nos da el ThundeReport tenemos la de examinar el contenido que almacena en la cache, donde veremos los dominios, los archivos almacenados del mismo la cantidad de disco que ocupan esos archivos y la economía que ha generado esa información almacenada en cache dependiendo a los accesos que se a echo a la información que esta almacenada.

4.1.3 Instalación de winbox

Esta es un programa ejecutable que nos permite administrar todos los equipos mikrotik en nuestro caso usaremos un routerboard 750 GL que es el equipo indicado para trabajar junto a thundercache. Se requiere que sea descargada es una plataforma libre.

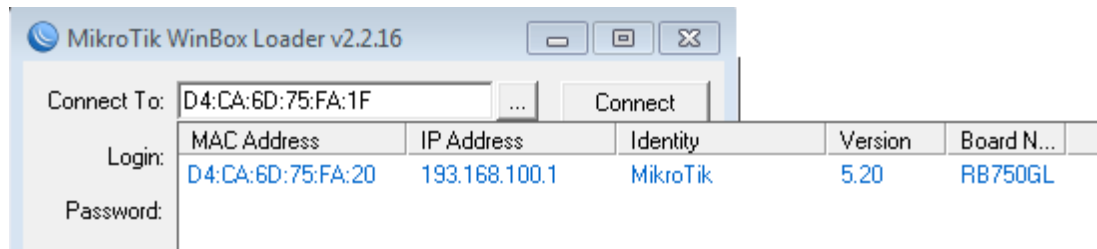
4.1.4 Configuración del Routerboard 750 GI

Figura 15: Conexión rb y winbox



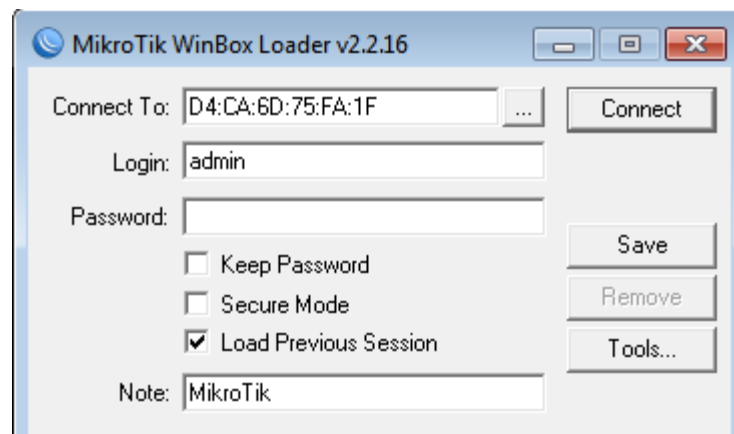
Primer paso conectarse desde el winbox al mikrotik, una vez abierto, hay que presionar el botón con puntos suspensivos (...) para poder escanear los dispositivos que tengan instalado MikroTik RouterOS en la red, en este caso, la imagen muestra sólo un dispositivo, que es el servidor que instalamos previamente.

Figura 16: winbox detecta al rb mediante MAC.



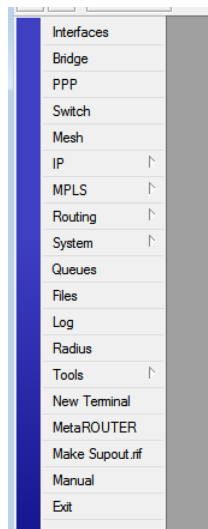
Una vez que se haya colocado la MAC en el cuadro Connect To es hora de presionar el botón Connect para establecer conexión con nuestro servidor.

Figura 17: Conectarse con winbox.



Cuando es la primera vez que nos conectamos al servidor, winbox descarga los módulos necesarios para mostrarnos todas las características del servidor. Y luego nos aparece el menú.

Figura 18: Menú winbox.



Interfaces, donde nos mostrará todas las tarjetas conectadas al servidor, incluyendo las tarjetas wireless, interfaces virtuales como pppoe-client, VLAN, VAP, etc. nos mostrará también, casi en tiempo real, el tráfico que está pasando por estas interfaces.

Wireless, si tuviéramos conectado una tarjeta wireless con chipset atheros, tendríamos la opción wireless para configurar esa tarjeta como un access point, cliente wireless, escanear redes, ver el estado de utilización de un canal, ver los clientes conectados a esta tarjeta, su estado, etc.

IP, aquí encontraremos un submenú con todas las opciones que hagan referencia a IP como por ejemplo:

Addresses, donde asignaremos IP's a las interfaces de red.

Firewall, donde entraremos opciones para bloqueo de IP's, puertos, NAT, marcado de paquetes, etc.

Hotspot, para configurar un hotspot server, y que nuestros clientes tengan acceso a internet mediante un usuario y clave.

Routes, para asignar políticas de ruteo.

Web proxy, donde configuraremos el web cache de Mikrotik, bloqueo de páginas, etc.

System, encontraremos opciones relativas al sistema, como por ejemplo:

Clock, aquí podremos configurar la hora en nuestro servidor.

License, podremos ver el estado de nuestra licencia de uso de Mikrotik RouterOS, así como el SoftID para el posterior licenciamiento.

Password, donde colocaremos una contraseña para asegurar el acceso a nuestro servidor.

Reboot, para reiniciar el servidor.

Resources, aquí aparecerá el estado físico del servidor como, la cantidad de memoria que tenemos, memoria libre, tipo de procesador, velocidad del procesador, espacio del disco duro, espacio libre, tiempo que lleva el servidor encendido, etc.

Shutdown, para apagar el servidor.

Queues, aquí podremos encontrar opciones para poder limitar la velocidad de nuestros clientes, asignar límites globales de velocidad, priorización de servicios, etc.

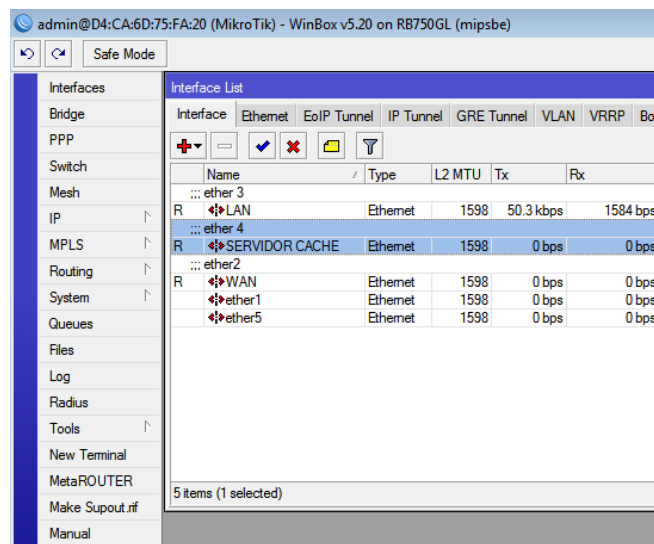
Files, veremos el directorio principal de Mikrotik, donde podremos crear backups de nuestra configuración y también restaurarlos, además de poder ver los archivos "log", la carpeta donde se almacena el portal cautivo de hotspot. **Tools**, encontraremos herramientas generales de Mikrotik, como ping, torch (para escanear conexiones), etc.

New Terminal, que es la consola Mikrotik, donde podremos acceder a todas las opciones y configurarlas por línea de comandos, es lo más práctico cuando se trata de muchas configuraciones, ya que podemos "pegar" listas de comandos para evitar todo el trabajoso proceso de hacer una configuración regla por regla.

Reconocer y nombrar las tarjetas de red conectadas al servidor

Ya en winbox vamos a *Interfaces* y en la pestaña *interface* veremos la lista de tarjetas de red reconocidas por Mikrotik.

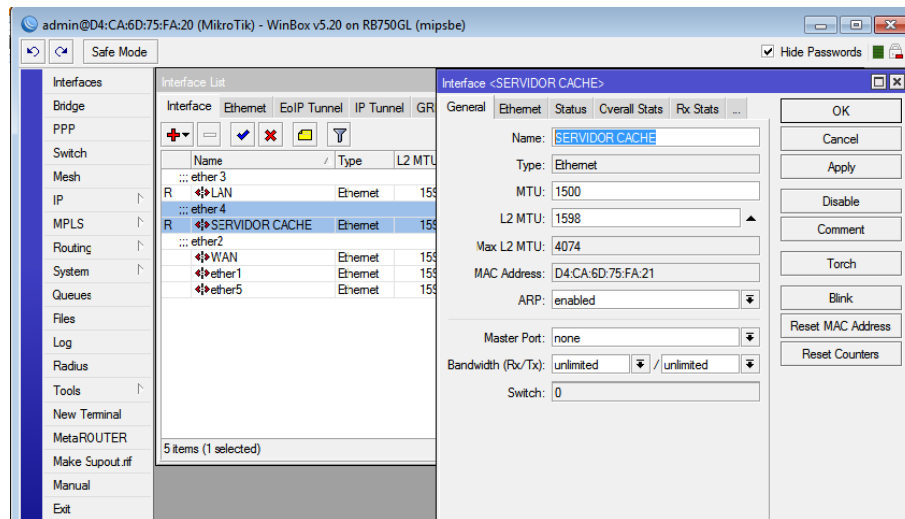
Figura 19: Lista de interfaces winbox.



Interface	Ethernet	EoIP Tunnel	IP Tunnel	GRE Tunnel	VLAN	VRPP	Bor
Name	Type	L2 MTU	Tx	Rx			
R ether 3	LAN	Ethernet	1598	50.3 kbps	1584 bps		
R ether 4	SERVIDOR CACHE	Ethernet	1598	0 bps	0 bps		
R ether2	WAN	Ethernet	1598	0 bps	0 bps		
ether1	Ethernet	1598	0 bps	0 bps	0 bps		
ether5	Ethernet	1598	0 bps	0 bps	0 bps		

5 items (1 selected)

Figura 20: Nombrando las interfaces.



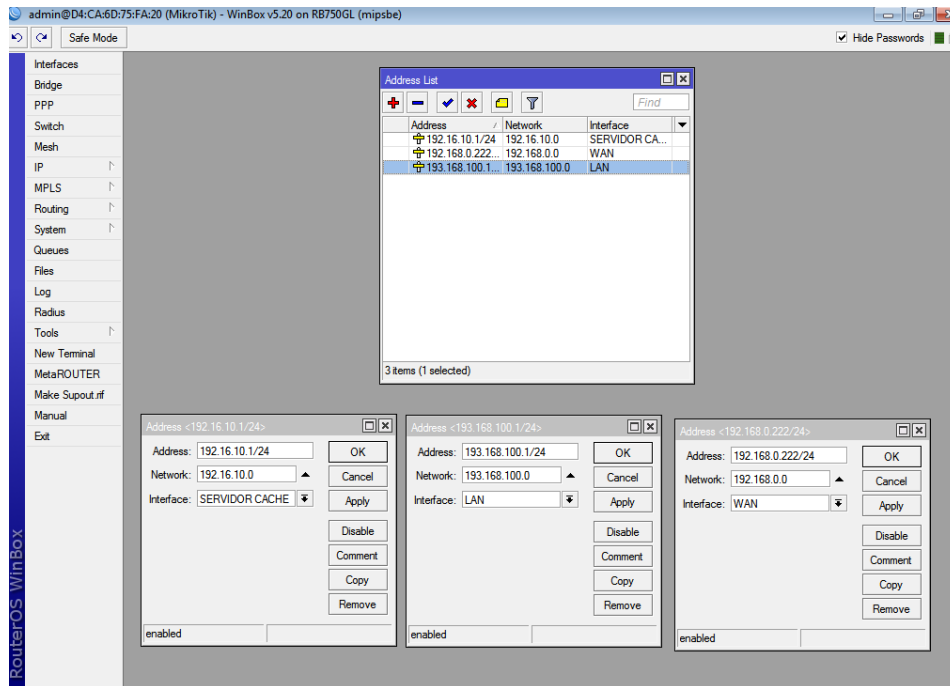
Una vez que sepamos a qué "ether#" corresponde cada una de las tarjetas, podemos cambiarles el nombre fácilmente haciendo doble click encima de su nombre, de la imagen de abajo, se puede ver el ejemplo.

Así tenemos las 2 tarjetas de red ya identificadas y 'nombradas' listas para proceder con la configuración.

Configurar WAN y LAN para conectarse a Internet desde el Servidor

Agregamos un IP a WAN (tarjeta de red WAN o interfaz de red WAN) para que nuestro servidor se pueda comunicar con el router. Para eso nos vamos a **IP -> Addresses** y agregamos una nueva regla (+)

Figura 21: Asignando ip a interfaces.



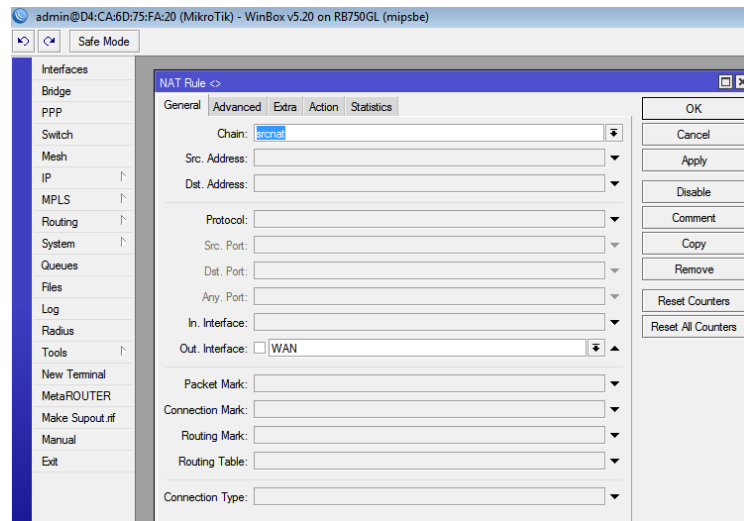
Address, aquí colocaremos el WAN IP del servidor, este IP tiene que estar en el mismo rango de red que la IP de nuestro router.

Interface, seleccionamos a qué interfaz de red asignaremos esta IP, en este caso elegiremos **ether1**, que está haciendo referencia a WAN

Network y **Broadcast**, no es necesario configurarlas manualmente ya que al momento de colocar el "/24" en Address, estas 2 opciones se configurarán automáticamente al momento de hacer click en el botón Apply u OK.

Una vez que tengamos las IP's configuradas para cada tarjeta, tocará hacer el "enmascarado", para eso vamos a **IP -> Firewall -> Pestaña NAT**, y agregamos una nueva regla (+)

Figura 22: Configurando NAT.

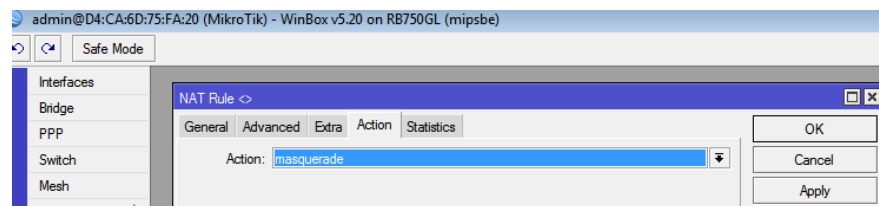


Chain, seleccionamos **srcnat**. Aunque siempre está así por defecto cuando se crea una nueva regla.

Out. Interface, seleccionaremos nuestra interfaz WAN

Pasamos a la pestaña **Action**.

Figura 23: Configurando el mascarado.



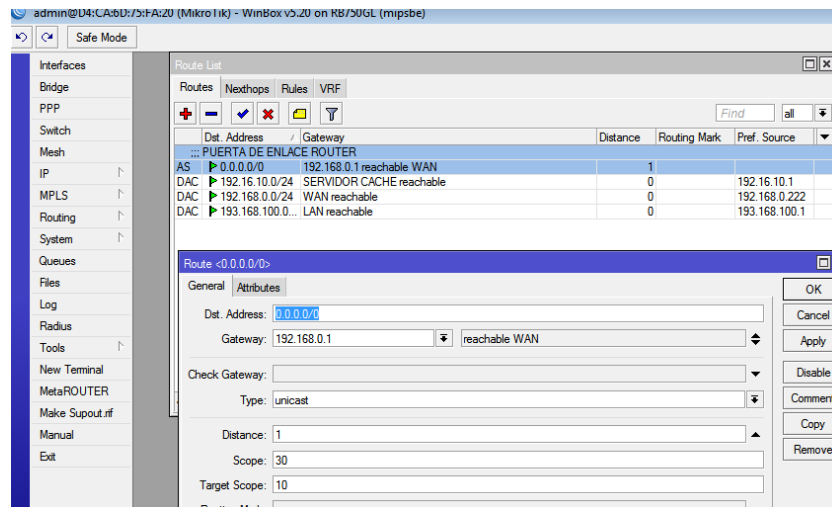
Action, elegiremos **masquerade** que nos permitirá enmascarar nuestras conexiones a detrás de la WAN IP, y así aislar nuestra red LAN.

Ruteo a la puerta de enlace

Especificar el **Gateway** o puerta de enlace donde el servidor se conectará a internet. Ya en la ventana **Route List**, veremos que hay 2 reglas que nosotros no agregamos.

Esto es normal ya que ahí se agregan las rutas de las IP's que asignamos previamente a las tarjetas de red en **Address List**. Para agregar la puerta de enlace que usará nuestro servidor, vamos a la pestaña **Routes** y agregamos una nueva regla (+).

Figura 24: Ruteando la puerta de enlace.

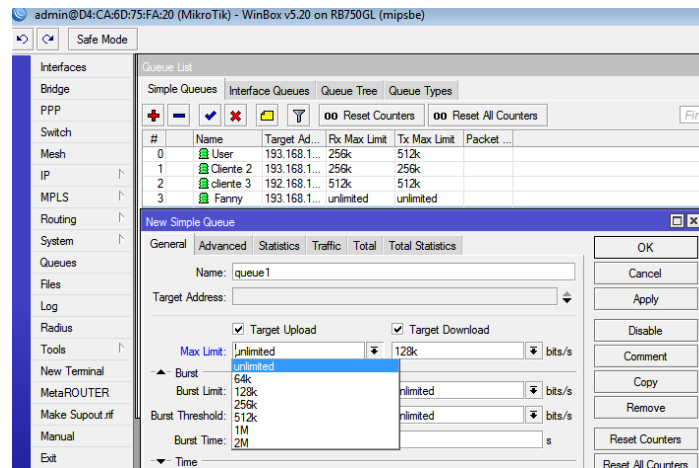


Gateway, aquí sólo colocaremos la puerta de enlace del router (el IP del router), de esta manera le estamos diciendo al servidor de dónde tiene que sacar internet para repartirlo a nuestros clientes.

Reachable quiere decir que Mikrotik comprobó la ruta que colocaste, o sea, la IP está 'viva'.

Agregar clientes a la interface LAN

Figura 25: Agregando clientes y asignando ancho de banda.



Name, aquí colocaremos el nombre del cliente, aunque en realidad puede ser cualquier palabra que nos ayude a identificarlo.

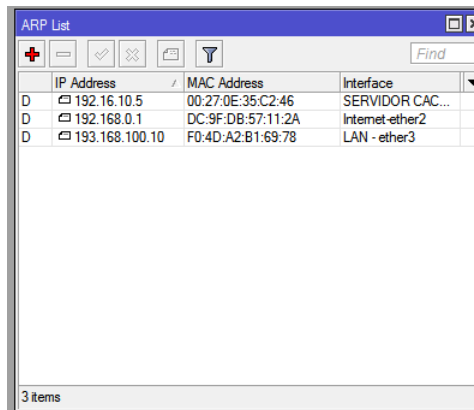
Target Address, especificaremos el IP de nuestro cliente al que queremos limitar el ancho de banda, del ejemplo, el IP es 192.168.10.20

Max Limit, es el lugar donde fijaremos la velocidad máxima de nuestro cliente, tanto de subida (upload) como de bajada (download), en este ejemplo, la subida es de 128k, y la bajada de 512k.

Si bien Mikrotik muestra varias velocidades preestablecidas para escoger, eso no nos impide que podamos asignar una velocidad "a nuestro gusto" tan solo escribiéndola con el teclado.

Amarre ip/mac por arp

Figura 26: Amarre de las interfaces con la MAC.



	IP Address	MAC Address	Interface
D	192.16.10.5	00:27:0E:35:C2:46	SERVERIDOR CAC...
D	192.168.0.1	DC:9F:DB:57:11:2A	Internet-ether2
D	193.168.100.10	F0:4D:A2:B1:69:78	LAN - ether3

3 items

El amarre IP/MAC por ARP es una de las medidas de seguridad más básicas que puede ofrecer el servidor Mikrotik, y consiste en tener una lista de relaciones IP/MAC registradas dentro del servidor; de esa manera, si un intruso con un IP, o MAC, o relación IP/MAC distinto a los ya registrados intentara tener internet, no tendrá respuesta alguna ya que no está en la lista de IP/MAC que registramos previamente.

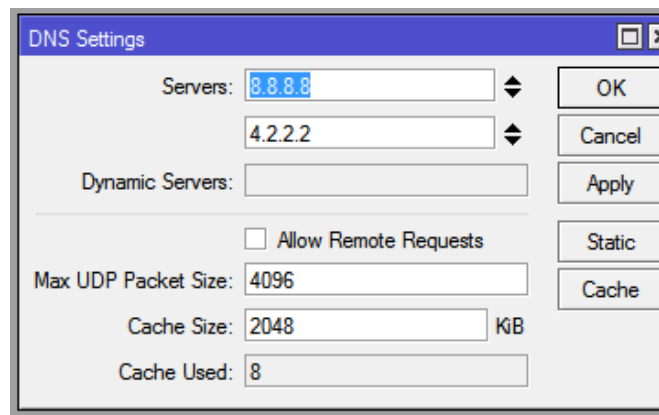
En la imagen de arriba veremos los IP's y MAC's de los dispositivos (PC's, VoIP, Celulares, Routers, etc.) que se agregaron automáticamente a la lista de ARP, y sabemos que fue automáticamente porque tienen la letra "**D**" al lado izquierdo de cada regla.

Se agregan automáticamente ya que tuvieron cierta comunicación con el servidor, podemos suponer que fue porque solicitaron internet al server o viceversa.

Configurar el dns cache de mikrotik

Ese DNS se encarga de resolver nombres de dominio a IP's, por ejemplo, si queremos ver <http://www.google.com>, nuestro PC envía una solicitud al servidor DNS para que resuelva ese dominio, y el servidor DNS le responderá con un IP; paso siguiente, el PC utilizará ese IP para conectarse.

Figura 27: Configuración DNS.



Redireccionar a nuestros clientes a thundercache con mikrotik

Hacemos el enmascarado por rango de red.

Figura 28: Re direccionando a los usuarios.

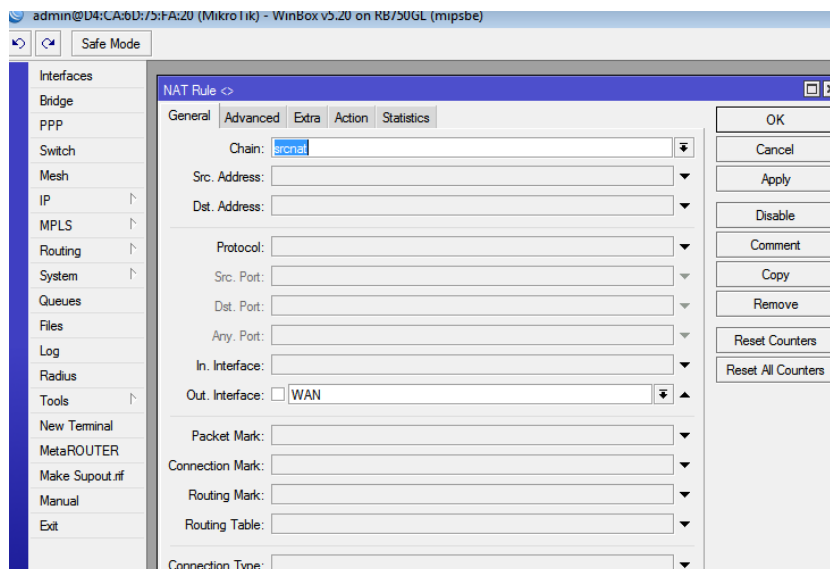


Figura 29: Asignando mascarado.

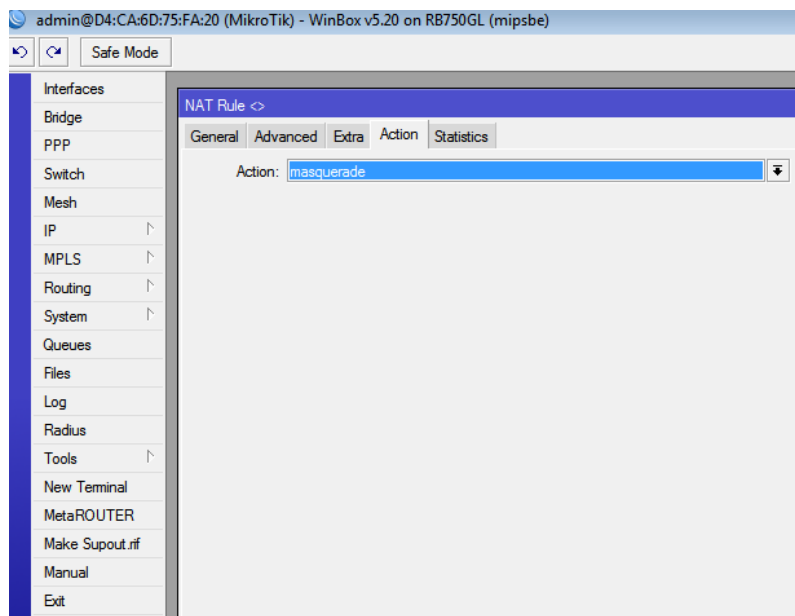
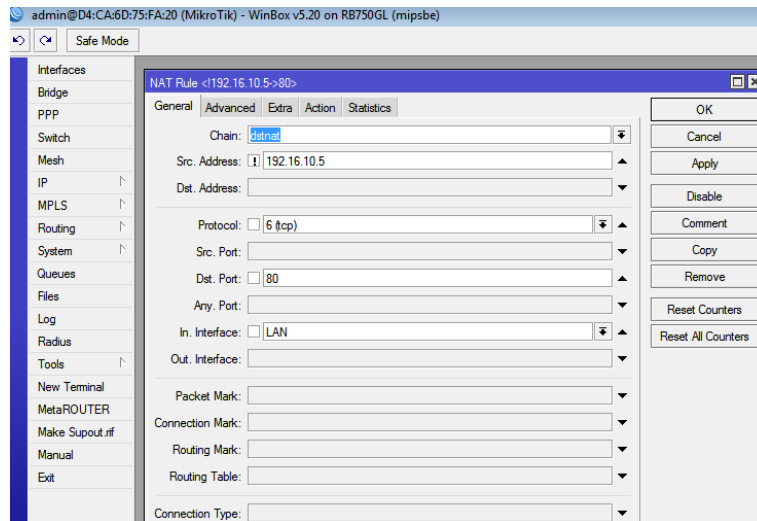
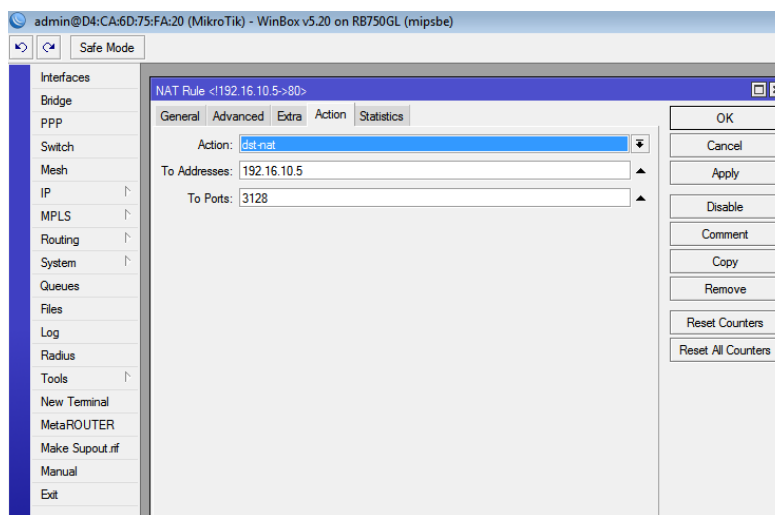


Figura 30: Colocando ip servidor para redirecciones.



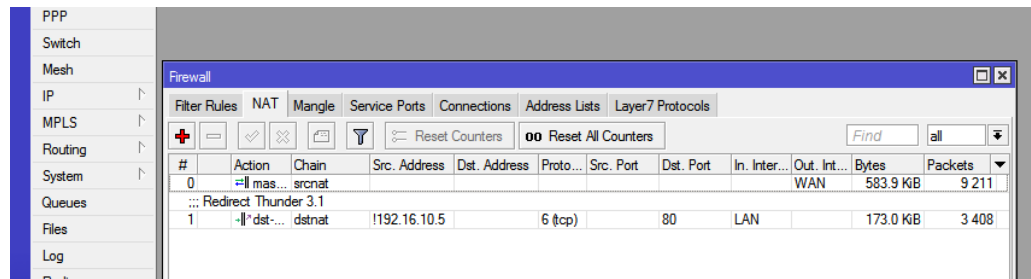
En este gráfico se coloca la ip del servidor cache para que el router se encargue de re direccionar todas las peticiones de la LAN hacia el servidor y no a la WAN.

Figura 31: Re direccionando a los usuarios.



Colocamos el puerto por el que va a escuchar las peticiones el thundercache que es el 3128 un puerto que es de uso exclusivo de servidores cache.

Figura 32: Vista firewall.



4.1.5 Diseño y desarrollo de scripts

Desarrollo de script

El mantenimiento de un servidor es importante para mantener el buen estado del equipo entre los procesos que se hará mantenimiento está el liberar memoria del servidor para evitar que colapse lo que haría que el servidor se vuelva lento.

Para realizar los scripts en debían necesitaremos la terminal putty, o se lo podría hacer mediante manipulación grafica con winscp, se recomienda realizarlo mediante la terminal y así se aprende a manipular comandos debían.

Debemos crear un archivo .sh de scripts, en un directorio del servidor.

Crearemos el siguiente script liberarmemoria.

Debemos guardarlo ponerle el nombre con extensión.sh

A continuación debemos darle permisos al script

Con el comando `chmod 777` le damos los permisos de ejecución al script

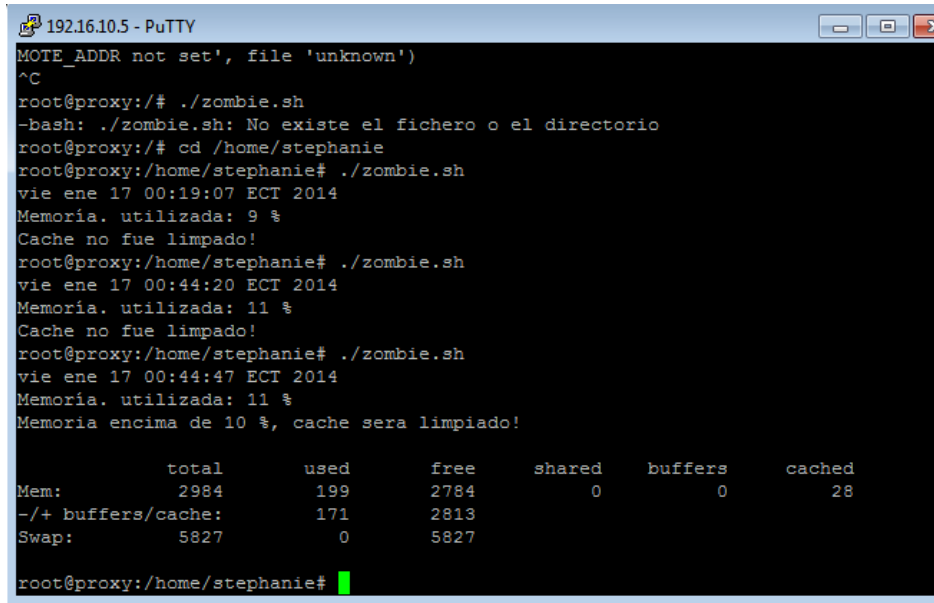
Si no lo hacemos no se podrá ejecutar el script.

Para ejecutar el scripts escribimos en la terminal así, nos vamos al directorio donde se encuentra el script:

```
/home/Stephanie/ liberarmemoria.sh
```

Y se mostrara la ejecución del script en la terminal de putty.

Figura 33: funcionamiento script.



```
192.16.10.5 - PuTTY
MOTD_ADDR not set', file 'unknown')
^C
root@proxy:/# ./zombie.sh
-bash: ./zombie.sh: No existe el fichero o el directorio
root@proxy:/# cd /home/stephanie
root@proxy:/home/stephanie# ./zombie.sh
vie ene 17 00:19:07 ECT 2014
Memoria. utilizada: 9 %
Cache no fue limpiado!
root@proxy:/home/stephanie# ./zombie.sh
vie ene 17 00:44:20 ECT 2014
Memoria. utilizada: 11 %
Cache no fue limpiado!
root@proxy:/home/stephanie# ./zombie.sh
vie ene 17 00:44:47 ECT 2014
Memoria. utilizada: 11 %
Memoria encima de 10 %, cache sera limpiado!

      total      used      free      shared    buffers    cached
Mem:      2984      199      2784         0         0         28
-/+ buffers/cache:      171      2813
Swap:      5827         0      5827

root@proxy:/home/stephanie#
```

A continuación veremos el contenido del script utilizado en la limpieza de memoria. Se encabeza con el Shell y a continuación se hace referencia a la función clean que requiere un parámetro numérico.


```

#!/bin/sh

PATH="/bin:/usr/bin:/usr/local/bin"

porcentaje=10
ramtotal=`grep -F "MemTotal:" < /proc/meminfo | awk '{print $2}`

ramlibre=`grep -F "MemFree:" < /proc/meminfo | awk '{print $2}`

ramusada=`expr $ramtotal - $ramlibre`

ramporcentaje=`expr $ramusada \* 100 / $ramtotal`

date
echo "Memoria. utilizada: $ramporcentaje %";

if [ $ramporcentaje -gt $porcentaje ]
then

echo "Memoria encima de $porcentaje %, cache será limpiado!";
sync
# 'Eliminando' cache:
echo 3 > /proc/sys/vm/drop_caches
echo
free -m
echo

else
echo "Cache no fue limpiado!";
exit $?
fi

```

En este script comenzamos dando un porcentaje de 10%, y almacenamos en una variable la información que se encuentra en un directorio virtual del sistema donde encontramos la memoria total y memoria libre.

Seguido hacemos la operación para obtener el porcentaje de memoria usada es necesario usar `expr` para realizar la operación para obtener el porcentaje de ram utilizada también usamos `expr`.

Debemos realizar una comparación entre el porcentaje que se tiene como referencia y la cantidad usada por el sistema si es mayor se procede a forzar la grabación de información pendiente en la cache, y de esta manera asegurarse de liberar los log con el comando `sync`.

Después hacemos referencia a la ruta donde se indica el manejo de memoria y le damos la opción 3 para que haga una limpieza de `pagecache`, `dentries`, `inodes`.

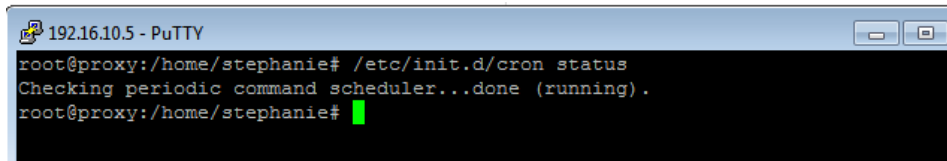
Con el comando `free` mostramos información relativa con la memoria, si en un caso el valor de referencia en este caso 10 es mayor que el porcentaje usado de memoria no se procederá hacer ninguna limpieza.

4.1.6 Automatización script

Después de haber realizado los scripts, debemos automatizarlos.

Para esta tarea debemos utilizar el demonio del sistema llamado `cron` que se encarga de ejecutar tareas `crontab`, pero todas los scripts ejecutados mediante `crontab` quedan en segundo plano por lo tanto no se mostraran resultados en la terminal `putty`.

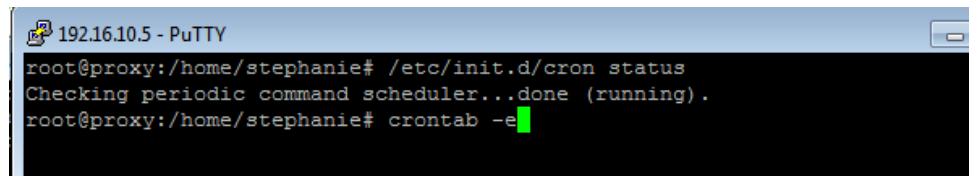
Figura 34: Estado de cron.

A terminal window titled '192.16.10.5 - PuTTY' showing the command `/etc/init.d/cron status` being executed. The output is 'Checking periodic command scheduler...done (running)'.

```
192.16.10.5 - PuTTY
root@proxy:/home/stephanie# /etc/init.d/cron status
Checking periodic command scheduler...done (running).
root@proxy:/home/stephanie#
```

En esta imagen podemos verificar que el demonio cron se encuentra activo ahora podremos automatizar tareas.

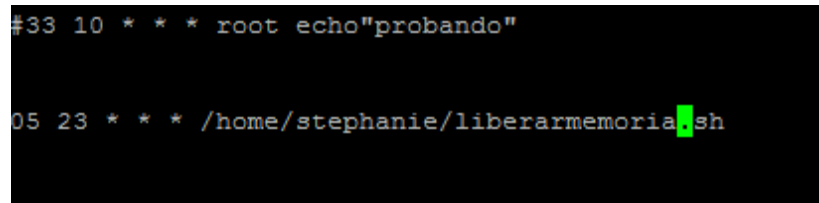
Figura 35: Editando fichero crontab.

A terminal window titled '192.16.10.5 - PuTTY' showing the command `/etc/init.d/cron status` and `crontab -e` being executed.

```
192.16.10.5 - PuTTY
root@proxy:/home/stephanie# /etc/init.d/cron status
Checking periodic command scheduler...done (running).
root@proxy:/home/stephanie# crontab -e
```

`crontab -e` podremos abrir el fichero `crontab` que es donde se programan las actividades que se deseen automatizar.

Figura 36: Fichero crontab.

A screenshot of the `crontab` file content showing two scheduled tasks. The first task is `#33 10 * * * root echo"probando"` and the second task is `05 23 * * * /home/stephanie/liberarmemoria.sh`.

```
#33 10 * * * root echo"probando"

05 23 * * * /home/stephanie/liberarmemoria.sh
```

El `crontab` está formado por 5 variables, empezando desde la izquierda minuto, hora, día del mes, mes, día de la semana.

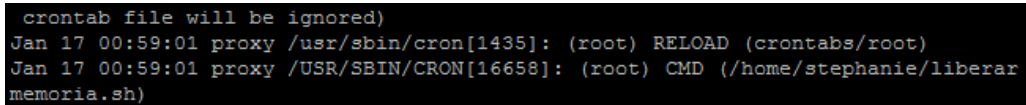
El asterisco significa que el campo toma todos los valores posibles. En este ejemplo el script se ejecutara a todos los 10 de cada mes.

4.1.7 Funcionamiento de las tareas crontab

Como se conoce que todas las tareas que son ejecutadas mediante el fichero crontab se ejecutan en segundo plano, así que una forma de ver si nuestro cron funciona será hacer un seguimiento de los registros los para observar la ejecución del script.

Con el comando `tail -f` y accediendo a la ruta del los de cron `/var/log/syslog` notaremos su ejecución.

Figura 37: Proceso ejecutado por el demonio.



```
crontab file will be ignored)
Jan 17 00:59:01 proxy /usr/sbin/cron[1435]: (root) RELOAD (crontabs/root)
Jan 17 00:59:01 proxy /USR/SBIN/CRON[16658]: (root) CMD (/home/stephanie/liberar
memoria.sh)
```

Le hicimos el seguimiento y notamos que a las 00:59 se ejecutó mi script liberar memoria.

4.2 Comprobación De Hipótesis

4.2.1 Variables Independientes

Dimensión eficiencia

El servidor web cache es una herramienta que aporta eficiencia en una red, al disminuir el tiempo de respuesta a todas Las peticiones web realizadas por los usuarios.

Al ser el servidor el que tiene prioridad al responder a todas las peticiones de los clientes se disminuye el número de peticiones que salen a la web.

4.2.2 Variables Dependientes

Dimensión Download

Al implementar el servidor cache y al hacer uso de su cache se podrá disfrutar de descargas más rápidas, debido a que se hace la descarga de forma local.

4.2.3 Mediciones realizadas en el edificio de tics.

Laboratorios sin servidor web cache

- Mayor consumo de ancho de banda
- Mayor tiempo de espera en las descargas

Análisis

Cuando se utiliza los laboratorios de la unidad de tics sin haberle implementado un servidor web cache en los laboratorios, el tiempo que los usuarios esperan para finalizar sus descargas en alto, y el ancho de banda empleado en estas descargas será elevado.

Para esta experiencia se utiliza la teoría de hipótesis nula, o Test de mediciones apareadas, por cuanto se trata del mismo sujeto medible.

Figura 38: Tiempo de espera usuarios

Descarga	Tiempo de espera sin cache	Tiempo de espera con cache	x	X - \bar{x}	(X - \bar{x}) ²
Video	15	4	11	-2	4
Instalador	35	10	25	12	144
Documento	5	2	3	-10	100
	55	16	39		248

Ho=: La implementación del servidor web cache en los laboratorios de tics no permitirá disminuir el tiempo espera en las descargas de los usuarios.

Ha=: La implementación del servidor web cache en los laboratorios de tics permitirá disminuir el tiempo espera en las descargas de los usuarios.

Aplicando la ecuación,

$$\bar{t} = \frac{\sum t'}{N}$$

$$\bar{t} = \frac{55}{3}$$

Tenemos el tiempo promedio de descargas sin utilizar el servidor web cache.

$$\bar{t} = 18,333 \text{ min.}$$

Ahora con el uso del servidor web cache obtendremos los siguientes datos.

$$\bar{t} = \frac{16}{3};$$

$$\bar{t} = 5.33 \text{ min.}$$

Y, la diferencia promedio de las entradas en los dos momentos.

$$\begin{aligned}\bar{x} &= \frac{\Sigma x}{N} \\ \bar{x} &= \frac{39}{3} \\ \bar{x} &= 13 \text{ min.}\end{aligned}$$

Con este análisis queda comprobada la ventaja de implementar el servidor web cache en los laboratorios de tics de la UTEQ.

Pero, además se aplica la teoría estándar; continuando el proceso, calculamos la desviación estándar de las diferencias de tiempo de espera.

$$\sigma x = \sqrt{\frac{\Sigma (x - \bar{x})^2}{N - 1}}$$

$N - 1$ es el grado de libertad dado: 9.

$$\begin{aligned}\sigma x &= \sqrt{\frac{248}{10 - 1}} \\ \sigma x &= 5,25 \text{ min.}\end{aligned}$$

Este sería el tiempo promedio esperado para que se cumpla la hipótesis nula, es decir, suponiendo que el sistema web cache **no** disminuye el tiempo de espera en las descargas de los usuarios.

Para que la hipótesis se verifique, debemos probar que se rechaza la hipótesis nula, para esto, se comprueba que $\sigma x \leq t$ donde t es la razón observada,

$$t = \frac{\bar{x}}{\frac{\sigma x}{\sqrt{N}}}$$

$$t = \frac{13}{\frac{5,25}{\sqrt{10}}}$$

$$t = \frac{13}{1,66}$$

$$t = 7,83$$

Como la razón observada es de **7,83**, es suficientemente más grande que el tiempo promedio esperado, se rechaza la hipótesis nula y se verifica la hipótesis planteada; luego comparando con la razón que aparece en la tabla T-Student, con 9 grados de libertad y un nivel de significancia del 5% que es: $t = 1.8331$, también se comprueba la hipótesis.

$$6.76 > 1.8331$$

Según los datos estimados, la hipótesis queda comprobada: **La implementación de un servidor web cache reducirá el tiempo de espera en las descargas de los usuarios.**

Figura 39: Velocidad de descarga

	descarga con cache (mbps) (v')	descarga sin cache (mbps) (v'')	x	$X - \bar{x}$	$(X - \bar{x})^2$
Video	3,9	1,76	2.14	0.11	0.0121
Instalador	4	1,83	2.17	0.11	0.0121
Documento	3,6	1,80	1.8	-0.23	0.0529
	11,5	5,39	6.11		0,0771

Ho=: La implementación del servidor web cache en los laboratorios de tics no permitirá aumentar la velocidad de descargas.

Ha=: La implementación del servidor web cache en los laboratorios de tics permitirá aumentar la velocidad de las descargas.

Aplicando la ecuación,

$$\bar{v} = \frac{\sum v'}{N}$$

$$\bar{v} = \frac{11,5}{3}$$

Tenemos velocidad promedio de descargas utilizando el servidor web cache.

$$\bar{v} = 3,83 \text{ mbps}$$

Ahora sin usar el servidor web cache obtendremos los siguientes datos.

$$\ddot{v} = \frac{5,39}{3};$$

$$\ddot{v} = 1.80 \text{ Mbps}$$

Y, la diferencia promedio de las entradas en los dos momentos.

$$\ddot{x} = \frac{\Sigma x}{N}$$

$$\ddot{x} = \frac{6,11}{3}$$

$$\ddot{x} = 2,03 \text{ mbps.}$$

Con este análisis queda comprobada la ventaja de implementar el servidor web cache en los laboratorios de tics de la UTEQ.

Pero, además se aplica la teoría estándar; continuando el proceso, calculamos la desviación estándar de las diferencias de velocidad de descargas.

$$\sigma x = \sqrt{\frac{\Sigma (x - \ddot{x})^2}{N - 1}}$$

$N - 1$ es el grado de libertad dado: 9.

$$\sigma x = \sqrt{\frac{0.0771}{10 - 1}}$$

$$\sigma x = 0.09$$

Para que la hipótesis se verifique, debemos probar que se rechaza la hipótesis nula, para esto, se comprueba que $\sigma_x \leq t$ donde t es la razón observada,

$$t = \frac{\bar{x}}{\frac{\sigma_x}{\sqrt{N}}}$$

$$t = \frac{2,03}{\frac{0,09}{\sqrt{10}}}$$

$$t = \frac{2,03}{0,028}$$

$$t = 72,5$$

Como la razón observada es de **72,5**, es suficientemente más grande que la velocidad promedio esperado, se rechaza la hipótesis nula y se verifica la hipótesis planteada; luego comparando con la razón que aparece en la tabla T-Student, con 9 grados de libertad y un nivel de significancia del 5% que es: $t = 1.8331$, también se comprueba la hipótesis.

$$72.5 > 1.8331$$

4.2.4 Análisis de resultados

Dimensión Eficiencia

Como mostramos en los gráficos el servidor sí fue eficiente ya que al observar el monitoreo de las interfaces se refleja que cuando se realiza la descarga de objetos web guardados se reduce el tiempo de espera de los usuarios y se las descargas son más rápidas. En este caso el servidor cache respondió con más de 3mb a la petición del usuario.

Dimensión funcionalidad

Al realizar la implementación del servidor en el laboratorio 1.04 de la unidad de TICS se comprobó que fue funcional se hizo cache de la navegación en ese laboratorio y con el servidor se disminuyó el tráfico hacia la WAN.

Dimensión Download

Se comprobó que al descargar objetos ya almacenados el cache es capaz de transmitir la información con un ancho de banda muy alto haciendo que se aumente la velocidad de descarga, ya que lo estamos haciendo localmente.

Hipótesis nula

Ho: La implementación del servidor web cache en la unidad de tics no permitirá dar un mejor servicio de navegación a los estudiantes de la UTEQ.

Hipótesis Alternativa

Ha: La implementación del servidor web cache en la unidad de tics permitirá dar un mejor servicio de navegación a los estudiantes de la UTEQ.

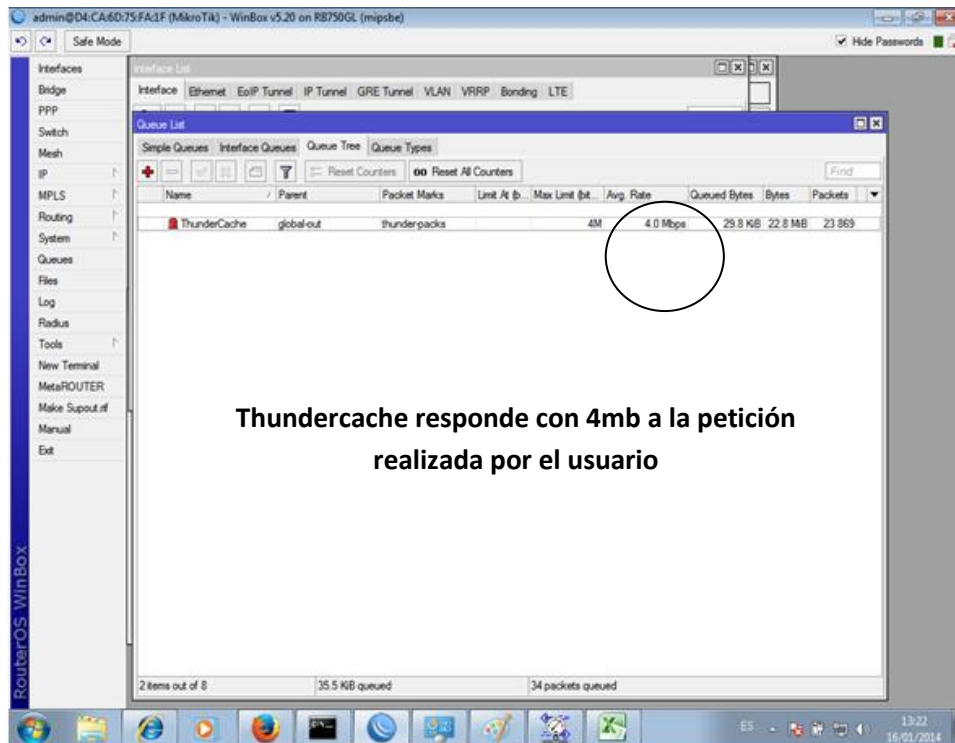
Basado en todos estos resultados se comprueba que la implementación del servidor web cache en los laboratorios de tics será una herramienta que mejorará el acceso a internet de los usuarios de la institución.

4.2.5 Gráficas de rendimiento tomadas en laboratorio de tics

Figura 40: Monitoreo descarga de video

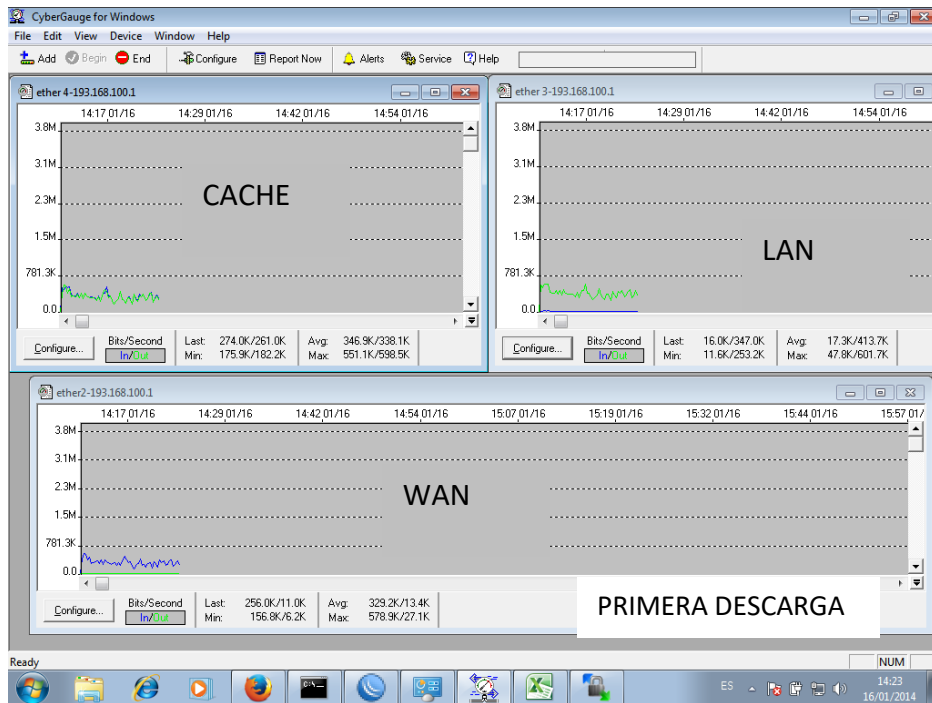


Figura 41: Monitoreo tráfico cache.



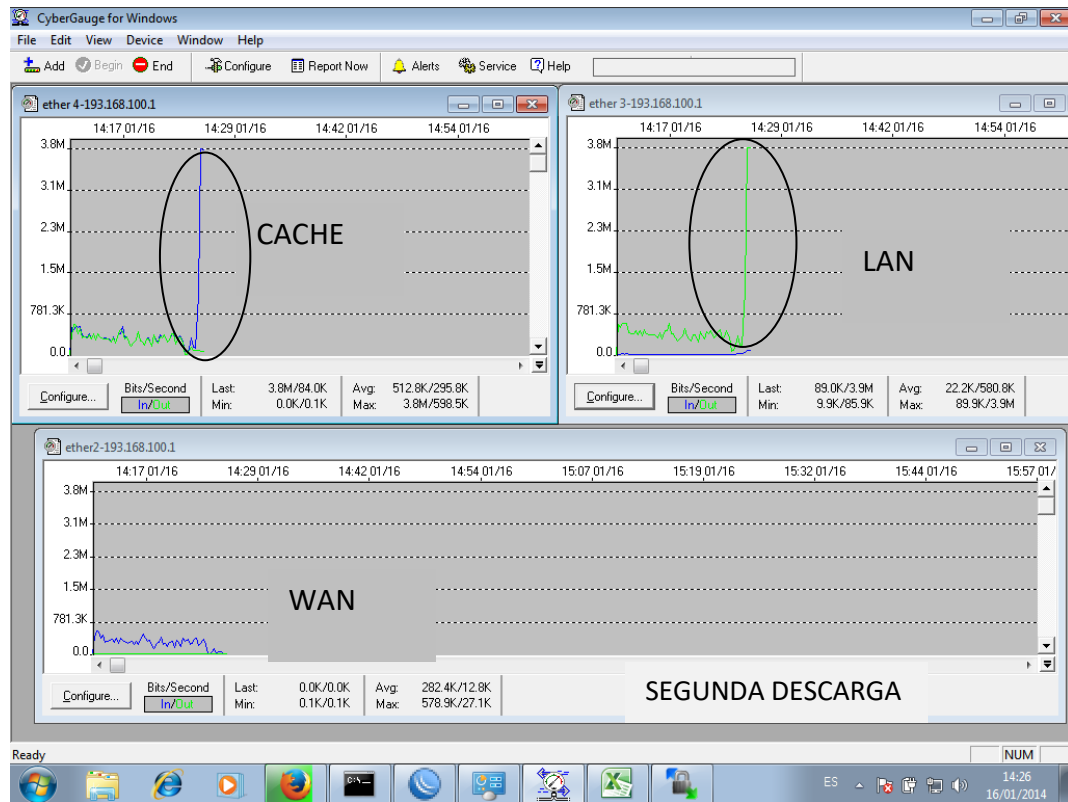
En esta imagen se puede observar que el thundercache está respondiendo a la petición de un usuario con una velocidad de transmisión de 4 Mbps

Figura 42: Monitoreo descarga instalador



En esta primera descarga del instalador veremos que todo el tráfico que circula en las interfaces proviene de la wan, la wan hizo la descarga de hay paso a la cache se realizo la copia y se respondio a la petición del usuario con un ancho de banda de 512kb que es lo que le establecimos a los clientes.

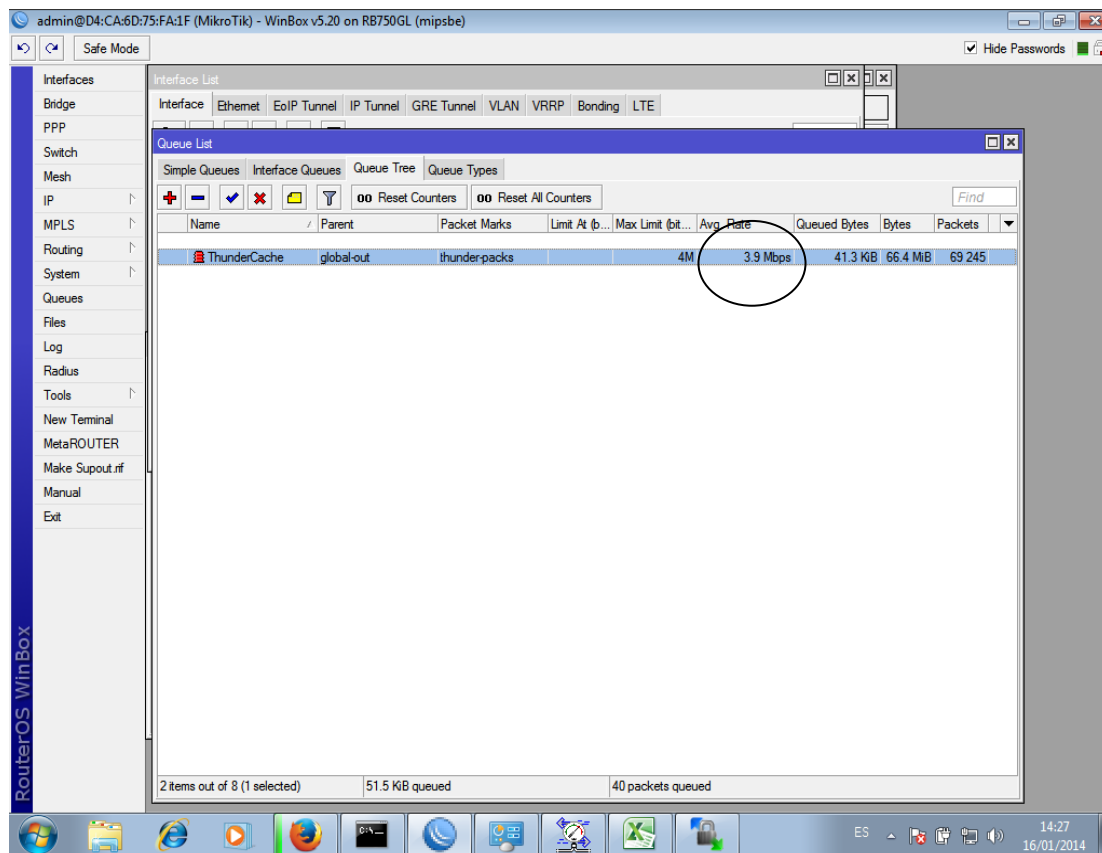
Figura 43: Monitoreo descarga de instalador.



Segunda descarga: En esta descarga ya no habra trafico en la wan la petición del usuario llegará a la cache vemos que a la cache le llego una peticion de mas de 3mb y esta responde a esa peticion llegando a la lan esa gran cantidad de infromación que esta guardara localmente y x eso su transmision es alta.

Viendo el grafico mas abajo se nota que la interface de la cache esla que responde a dicha peticione con todo su limite de ancho de banda que le establecimos 4mb.

Figura 44: Monitoreo tráfico cache.



En esta petición veremos que el thundercache repondió al usuario conn 3,9 mbps así la descarga se realizará de forma mucho más rápida.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5. 1 Conclusiones

La tecnología web cache es una poderosa herramienta para optimizar una red, con la implementación realizada en los laboratorios de la unidad de tics utilizando Debian 6 para la plataforma del servidor y thundercache 3.2 como el sistema de cache, se pudo comprobar lo siguiente:

- Se logró reducir el tiempo de espera en las descargas que realizan los usuarios.
- Se aumento la velocidad de descargas en los laboratorios de la unidad de tics.
- Se disminuyó el consumo de ancho de banda utilizado por los usuarios en los laboratorios de la UTEQ.
- Mediante la implementación y automatización de script se logró liberar la memoria del servidor para evitar mal funcionamiento del equipo.

5.2 Recomendaciones

- Después de haber implementado y observar los beneficios que brinda un servidor cache, se sugiere el uso de esta tecnología y así de esta manera se conseguirá un mejor rendimiento de la red.
- Para implementar el sistema de web cache se debe tener en cuenta que no todas las plataformas GNU/Linux funcionan con thundercache, en esta ocasión se usó debían ya que es la combinación ideal Debian + Thundercache.
- Se recomienda usar un hardware con gran capacidad para evitar problemas en el funcionamiento del equipo.

CAPITULO VI

LITERATURA CITADA

6.1 Bibliografía

Héctor Luis Ávila. (2006). Introducción a la metodología de investigación. Editorial eumed.net. Pág. 21.

Francisco Solsona. Elisa Viso Gurovich, Mauricio Aldazosa. (2007). Manual de Supervivencia en Linux. Universidad Nacional Autónoma de México, Facultad de Ciencias.

Vidal Díaz de Rada. (2009). Análisis de datos de encuestas. Editorial UOC. Etapas en una Investigación con Encuestas. Pág. 19.

Juan Carlos Pérez González. (2009). Programación de Shell scripts bajo Linux. Editorial ENI. Pág. 122.

Diego Redolar Ripoll. (2010). Fundamentos de psicobiología. Editorial UOC. Metodología Cuasi-Experimental. Pág. 57.

Irving Alexander Bermúdez Silva. (2007). Debian GNU/Linux Para El Usuario Final: Editorial ICCO.

Neil J. Salkind. (1999). Métodos de investigación. Editorial Prentice Hall Inc. Pág. 213.

Christopher Negus. Francois Caen. (2008). Ubuntu Linux Toolbox: 1000+ Commands for Ubuntu and Debian Power Users. Published by Wiley Publishing Inc.

6.2 Enlaces web

<http://www.slideshare.net/jcalderon/conf-educatunlsf>

http://www.estebanmendieta.com/blog/wp-content/uploads/Decreto_1014_software_libre_Ecuador.pdf

<http://www.areatecnologia.com/redes-informaticas.htm>

<http://www.debianaddict.org/article62.html>

<http://pplware.sapo.pt/linux/o-que-e-o-crontab/>

<http://www.emezeta.com/articulos/10-comandos-interesantes-para-linux>

<http://helpdeskgeek.com/linux-tips/crontab-howto-tutorial-syntax/>

<http://helpdeskgeek.com/linux-tips/shell-script-backup-directory-linux/>

http://fahwiki.net/index.php/VMware,_Linux_Folding_and_Clock_Drift

http://profesores.fi-b.unam.mx/jlfl/Seminario_IEE/tecnicas.pdf

<http://thundercache.es/recursos/paralelo/redireccion-y-full-cache/>

<http://www.alterserv.com/foros/index.php?topic=7.0>

<http://www.areas.net/comofunciona/conexion/3.htm>

<http://mikrotikxperts.com/index.php/2013-03-28-19-49-36/conocimientos-basicos/159-modelo-osi-y-tcp-ip>

<http://www.areas.net/comofunciona/conexion/3.htm>

<http://mikrotikxperts.com/index.php/2013-03-28-19-49-36/conocimientos-basicos/159-modelo-osi-y-tcp-ip>

<http://blog.desdelinux.net/donde-se-encuentran-los-archivos-de-registro-logs-en-linux/>

<http://www.computerworld.es/archive/la-importancia-de-la-tecnologia-caching>

<http://iie.fing.edu.uy/~vagonbar/unixbas/editorvi.htm>

<http://www.esdebian.org/documentacion/24008/comandos-basicos-vi>

<http://www.linuxfocus.org/Castellano/September1999/article103.html>

<http://dns.bdat.net/documentos/cursos/ar01s12.html>

<http://linuxcomandos.blogspot.com/2008/02/comando-sync-forzar-grabacin-de-datos.html>

CAPITULO VII

ANEXOS

Anexo 1: Instalación Sistema Operativo Debian

Figura 45: Inicio Debian



Figura 46: Selección del país Debian

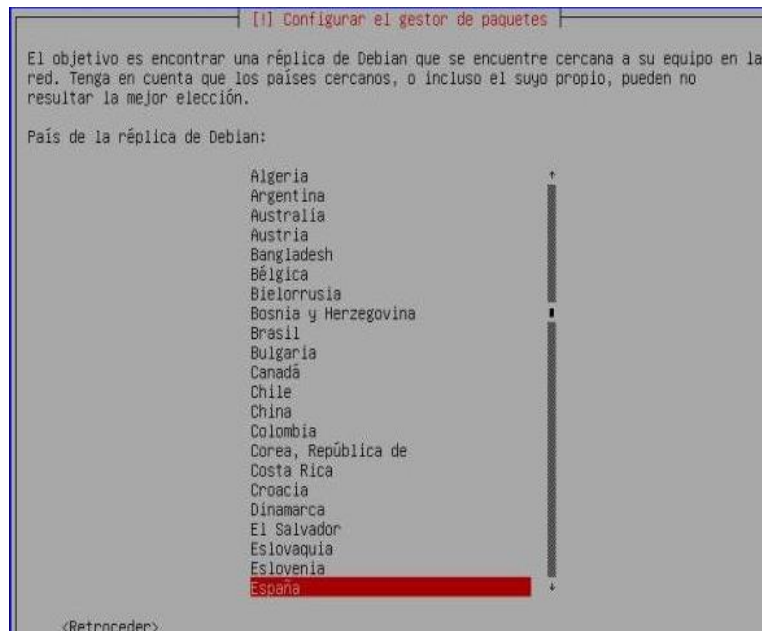


Figura 47: Selección idioma del teclado Debian

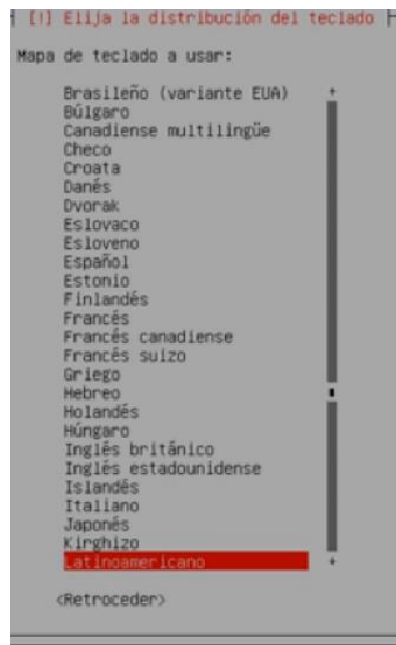


Figura 48: Configuración de red Debian

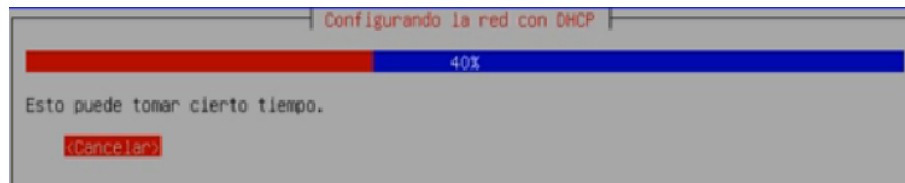
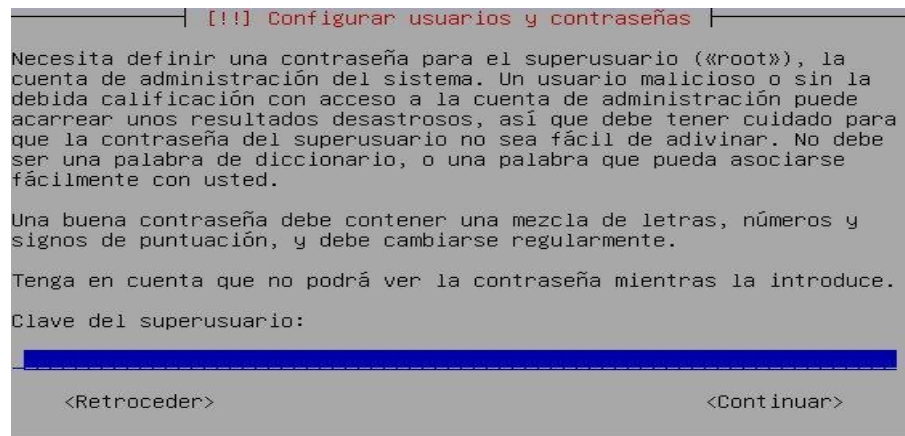


Figura 49: Configuración usuarios y contraseña Debian



El equipo donde residirá el thundercache deberá tener un usuario y contraseña como medidas de seguridad.

Figura 50: Asignación de usuario a Debian

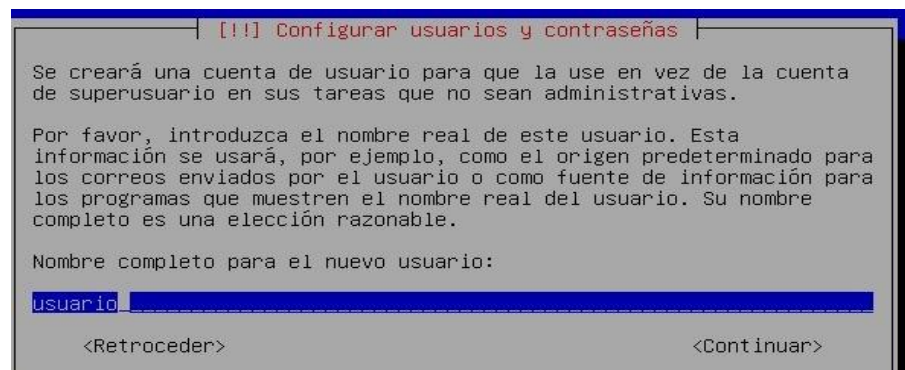


Figura 51: Configurando la red Debian

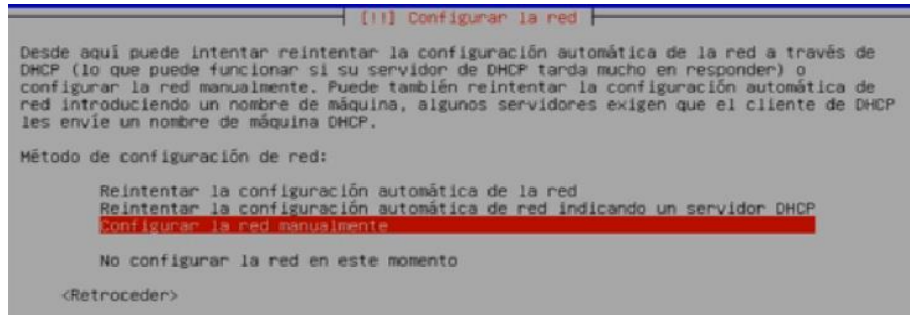
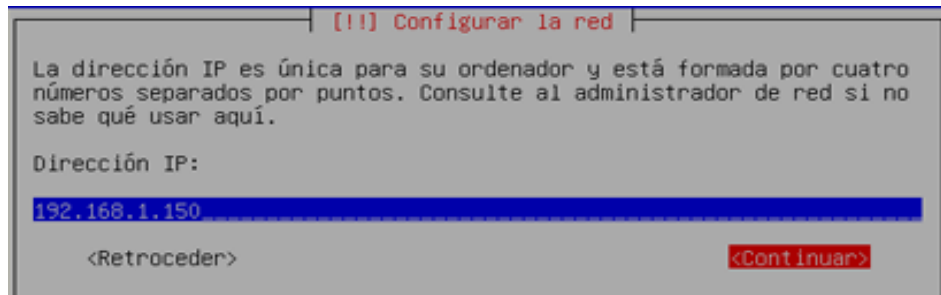


Figura 52: Asignando ip al equipo Debian



También se requiere que el equipo tenga una IP que será necesaria para acceder al equipo cuando sea necesario con apoyo del usuario y contraseña.

Figura 53: Servicio grub Debian

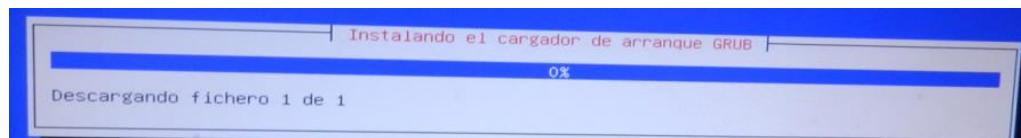
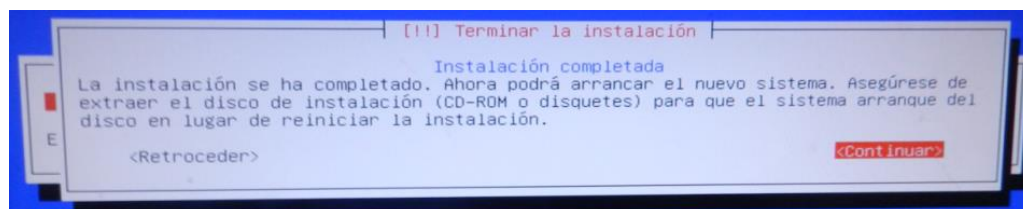
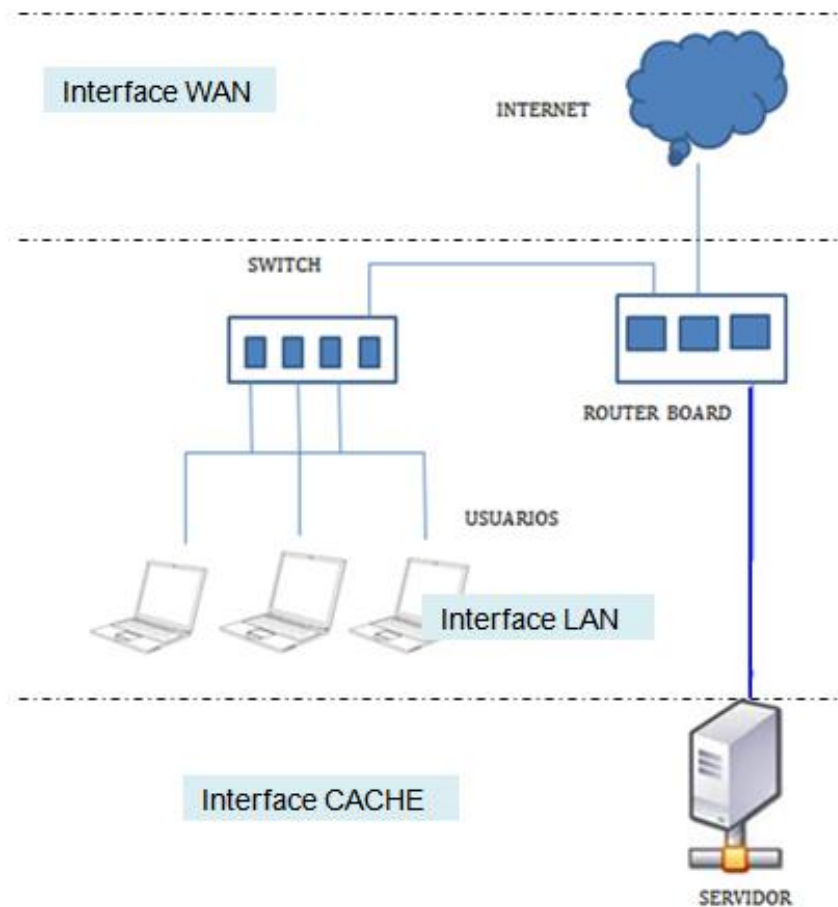


Figura 54: Instalación finalizada Debian



Anexo 2: Diseño de una red con thundercache

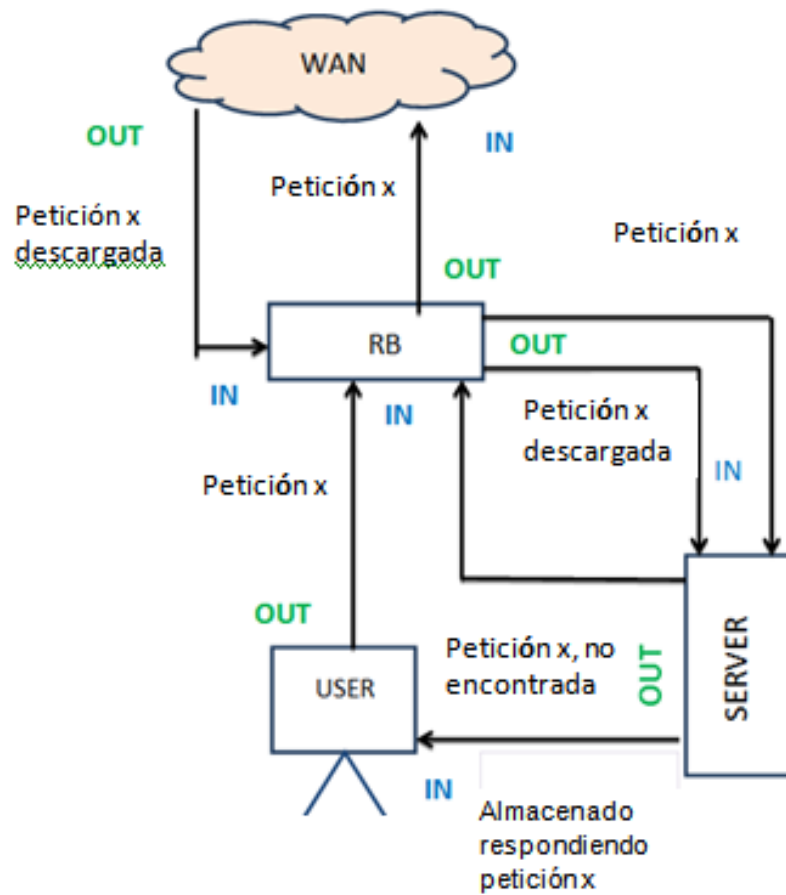
Figura 55: Diseño de la red.



En el siguiente gráfico notaremos 3 interfaces LAN, WAN, CACHE O SERVIDOR, los equipos que se utilizarán están capacitados para esta función, switch para hacer la conexión entre usuarios, rb 750gl que actúa como una pasarela donde todas las peticiones deben pasar para llegar a su destino.

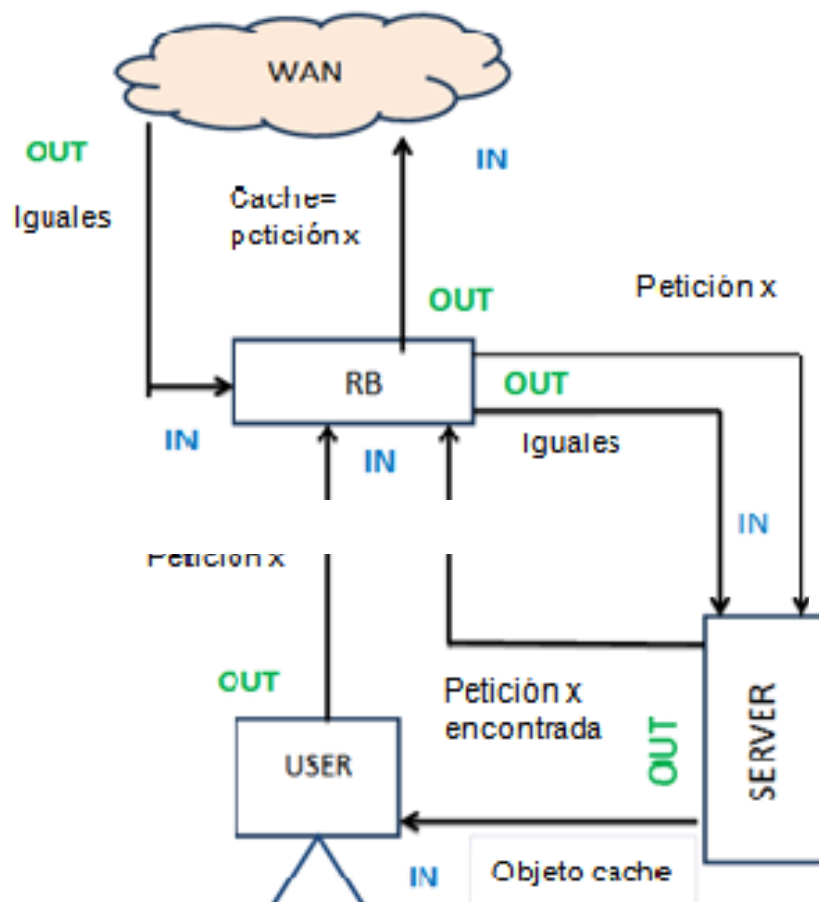
Anexo 3: Funcionamiento de la red cuando no hay cache.

Figura 56: Objeto web no encontrado en cache.

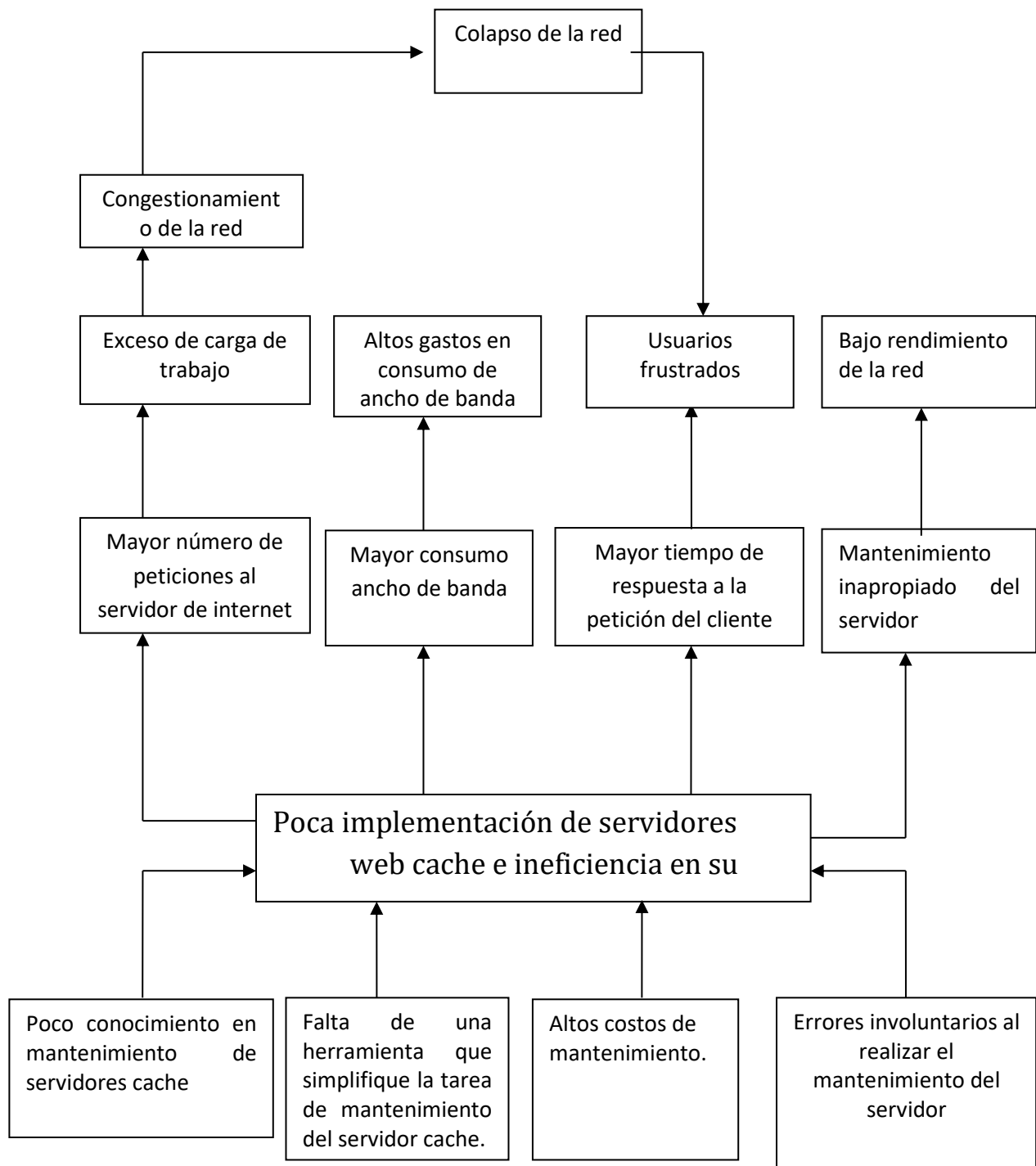


Anexo 4: Funcionamiento de la red cuando hay cache.

Figura 57: Objeto web encontrado en cache.



Anexo 5: Árbol de problemas



Anexo 6: Datos Generales

Datos Personales		
Cédula	Nombres	Apellidos
1205994161	Stephanie Elizabeth	Mora Andrade
Teléfono	Correo Electrónico	
0990391189	tifanimor@hotmail.com	

Datos del Egresado		
Título a obtener	Carrera	Paralelo
Ingeniero en Sistemas	Ingeniería en Sistemas	"A"
Fecha de Egresado		
08/01/2013		

Datos de Tesis		
IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHE BAJO LA PLATAFORMA GNU/LINUX, AUTOMATIZANDO SCRIPPS PARA ÓPTMIMIZAR EL MANTENIMIENTO DEL SERVIDOR.		
Res. Aprobación de tesis	Res. Fecha de sustentación	Fecha Sustentación
		20/01/2014
C.I Tutor	Nombre del tutor	Fecha de Incorporación
	Ing. Carlos Caizaguano	20/01/2014

