UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TRABAJO DE GRADO PREVIO LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TEMA:

"IMPLEMENTACIÓN DE UN SERVIDOR WEB CACHÉ Y DNS CACHÉ PARA LA RED DE INFRAESTRUCTURA DE LA UNIVERSIDAD TÉCNICA DEL NORTE"

AUTOR: LAGUNA PONCE HENRY ANDERSON

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

IBARRA-ECUADOR

2023

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AUTORIZACIÓN DE USO DE PUBLICACIÓN A FAVOR DE LA UNIVERSIDADTÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO	
Cédula de Identidad	040150377-6
Apellidos y Nombres	Laguna Ponce Henry Anderson
Dirección	Leopoldo N. Chávez O5-007 & Calle 2
E-mail	halagunap@utn.edu.ec
Teléfono fijo	(06)2770540
Teléfono móvil	0996326581
DATOS DE LA OBRA	
Título	"IMPLEMENTATACIÓN DE UN SERVIDOR WEB CACHÉ Y DNS CACHÉ PARA LA RED DE INFRAESTRUCTURA DE LA UNIVERSIDAD TÉCNICA DEL NORTE"
Autor	Laguna Ponce Henry Anderson
Fecha	02/05/2023
Programa	■ PREGRADO ☐ POSGRADO
Título	Ingeniero en Electrónica y Redes de Comunicación
Director	MSc. Jaime Roberto Michilena Calderón

UNIVERSIDAD TÉCNICA DEL NORTE

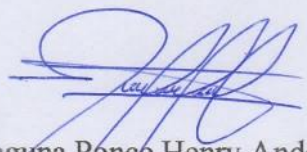
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la universidad en caso de reclamación por parte de terceros.

Ibarra, a los 6 días del mes de junio de 2023

EL AUTOR:



Laguna Ponce Henry Anderson
C.I: 040150377-6

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

MSC. JAIME ROBERTO MICHILENA CALDERÓN, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación denominado: "IMPLEMENTATACIÓN DE UN SERVIDOR WEB CACHÉ Y DNS CACHÉ PARA LA RED DE INFRAESTRUCTURA DE LA UNIVERSIDAD TÉCNICA DEL NORTE", ha sido desarrollado por el señor LAGUNA PONCE HENRY ANDERSON bajo mi supervisión.

Es todo en cuanto puedo certificar en honor a la verdad.

MSc. Jaime Roberto Michilena Calderón

DIRECTOR

AGRADECIMIENTOS

Agradezco a Dios por permitirme guiar este camino de estudio y jamás soltar mi mano, a mis padres por el esfuerzo y apoyo que me brindaron todos los días, familiares y amigos por ser ese eslabón fundamental de ánimo que siempre necesité.

Muchas gracias a la Universidad Técnica del Norte por brindarme el conocimiento del que hoy me siento orgulloso de haber adquirido, a mis maestros por siempre influir en los buenos valores que debe tener un excelente profesional y que siempre me lo llevaré como ejemplo en cada actividad que se me encargue, y finalmente a mi tutor Magister Jaime Michilena por siempre apoyarme en el desarrollo y culminación de este proyecto de manera exitosa.

Laguna Ponce Henry Anderson

DEDICATORIA

Dedico la culminación del estudio de esta carrera, primeramente, a mí, por jamás haberme rendido pese a muchas adversidades, a mi familia, por ser ese eje central de mi formación personal en el ámbito profesional y de valores y finalmente a mis amigos que siempre me acompañaron en este camino.

Laguna Ponce Henry Anderson

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

RESUMEN

El presente tema de titulación parte en base a la búsqueda de sistemas que permitan mejorar la eficiencia de la red, ahorrando costes económicos y de infraestructura es por ello que el análisis parte en la implementación de herramientas para el levantamiento de servidores de DNS y Web caché, los cuales, según la teoría, ayudarían a cumplir este fin.

Para la investigación y desarrollo del proyecto se establece una metodología en cascada, partiendo del análisis con la información teórica de los servicios de DNS y Web caché mediante la revisión bibliográfica, continúa con el diseño levantando información de la situación de la red anterior a la implementación de los servidores del proyecto y mediante un modelo de dimensionamiento disponer los recursos de hardware necesarios para implementar los servicios, se procede con la etapa de implementación donde se realiza el levantamiento de los servidores de DNS y Web caché dentro de la cartera de servicios en la infraestructura de hiperconvergencia, verificando la viabilidad de cada servicio y realizando pruebas de funcionamiento en las zonas de aplicación, finalmente se procede a la etapa de verificación.

Dentro de la verificación se constituye un análisis de costo-beneficio, que junto con los resultados, se obtiene que la implementación del servidor Web caché no es viable en la red, sin embargo el servicio de DNS caché si lo es, obteniendo resultados positivos tanto para la infraestructura de red como para el usuario, los tiempos de respuesta de una consulta se reducen permitiendo al sistema ahorrar consumos de ancho de banda y de recursos de hardware, pues el consumo del servicio en funcionamiento es mínimo, al mismo tiempo mejora características en comparación al servidor antiguo.

ABSTRACT

The present graduation topic is based on the search for systems that allow improving network efficiency, saving economic and infrastructure costs. That is why the analysis starts with the implementation of tools for the deployment of DNS and Web cache servers, which, according to theory, would help achieve this goal.

For the research and development of the project, a waterfall methodology is established, starting with the analysis of theoretical information on DNS and Web cache services through bibliographic review. It continues with the design, gathering information about the network situation before the implementation of the project servers, and using a sizing model to provide the necessary hardware resources to implement the services. The implementation stage proceeds with the deployment of DNS and Web cache servers within the service portfolio in the hyperconvergence infrastructure, verifying the viability of each service and conducting functional tests in the application areas. Finally, the verification stage is carried out.

During the verification stage, a cost-benefit analysis is conducted. Together with the results, it is determined that the implementation of the Web cache server is not viable in the network. However, the DNS cache service is viable, obtaining positive results for both the network infrastructure and the user. The response times for a query are reduced, allowing the system to save bandwidth consumption and hardware resources since the service consumption in operation is minimal. At the same time, it improves features compared to the old server.

CAPÍTULO I. ANTECEDENTES

En el presente capítulo se desarrollarán los antecedentes. El capítulo intenta proporcionar una introducción en el tema de investigación, da a conocer las bases esenciales del proyecto: tema, problema, objetivo general, objetivos específicos, alcance y justificación.

1.1. Tema

IMPLEMENTATACIÓN DE UN SERVIDOR WEB CACHÉ Y DNS CACHÉ PARA LA RED DE INFRAESTRUCTURA DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

1.2. Problema

La Universidad Técnica del Norte es una institución de educación superior que se conforma de once campus. El principal campus “El Olivo” cuenta con dieciséis edificios, entre ellos la biblioteca, auditorios, centros de copias e impresión, salas de clase, laboratorios, etc. y acoge entre docentes, estudiantes y funcionarios en jornada diurna y nocturna un aproximado de 12000 personas (Universidad Técnica del Norte, s.f.). A partir del mes de marzo del 2021, la red de la institución en dicho campus ha venido presentando ciertos problemas de conectividad y de servicios, principalmente con la saturación de los canales, problemas en la red inalámbrica y en la zona de firewall de seguridad perimetral.

Ciertos servicios de red como contenidos web, mail, entre otros, han venido creciendo de forma exponencial en los últimos tiempos, el acceso a dichos servicios por parte las organizaciones o de usuarios particulares ha llevado a que su demanda también crezca (Abbate, s.f.). Esta evolución conlleva a buscar mecanismos de replicación, que garanticen eficiencia y disponibilidad sin que afecte directamente en el consumo de ancho de banda que se maneja actualmente en la infraestructura de la institución. Esto lleva a dos problemas significativos, el aumento de costos de mantenimiento de la red de infraestructura, así como

también el crecimiento en el tráfico de la red (Sánchez, Figueroa, Gamarra, & Mayorga, s.f.), si se tomamos cada parte de estos problemas, entre ellas, el elevado consumo de ancho de banda, el sistema frena los procesos de acceso a la red y a los servicios que un usuario requiere.

Por ende, para la institución, el solicitar el aumento de ancho de banda a sus proveedores no resulta la solución más eficiente para solventar los problemas presentados, es mucho más conveniente indagar en soluciones basadas en gestión de administración o implementación de servicios, de tal forma que esto no conlleve a generar gastos elevados pero que tampoco se vea afectada la calidad del servicio y más bien se aprovechen de mejor manera los recursos existentes y el desarrollo de nuevas tecnologías (Vergara, 2017).

1.3. Objetivos

1.3.1. Objetivo General

Implementar un servidor Web Caché y DNS Caché en la infraestructura de red de la Universidad Técnica del Norte de la ciudad de Ibarra

1.3.2. Objetivos Específicos

- Establecer información teórica sobre los servicios de Web Caché y DNS Caché mediante modelos de dimensionamiento de recursos.
- Establecer parámetros para la implementación de los servicios basados en la arquitectura de hiperconvergencia de Cisco, aplicando una metodología en base al dimensionamiento de los servidores.
- Implementar los servicios de Web Caché y DNS Caché en convergencia con la cartera de servicios existentes en la red.
- Presentar un análisis de relación costo-beneficio en base a los resultados obtenidos a la aplicación de los servicios de Web Caché y DNS Caché.

1.4. Alcance

Para concluir con éxito el desarrollo del proyecto se espera implementar los servicios de Web Caché y DNS Caché, ambos, basados en un diseño de modelos de dimensionamiento de recursos de infraestructura para el número masivo usuarios que se manejan actualmente en la red del campus “El Olivo” de la Universidad Técnica del Norte que actualmente rondan en 12000 entre docentes, estudiantes y funcionarios (Universidad Técnica del Norte, s.f.), esto tomando en cuenta el entorno repotenciado de la infraestructura. De estos servicios se debe comprender y analizar la información teórica que abarca cada uno y así, con el fin de cumplir los objetivos planteados.

La metodología de investigación del proyecto parte del dimensionamiento de los servicios con el establecimiento de los requerimientos de los usuarios y del sistema, la definición de métricas para evaluar los parámetros a ser dimensionados en el servidor, y finaliza con la configuración de los equipos de prueba, lo cual se basa en la elaboración de una propuesta del modelo matemático evaluando primero los resultados para el modelo de colas M/M/1 y luego implementa los resultados para el modelo de colas G/G/1, continúa con la validación de este mediante el análisis de datos de entrada y la estimación de parámetros como, la tasa de llegada de solicitudes de usuarios al servidor y los recursos de acceso, la solución y el análisis de datos salida, esto, debería minimizar el número de recursos de infraestructura necesarios para abastecer la demanda de solicitudes de visita a las páginas Web, entendiéndose que, al minimizar el número de recursos de infraestructura necesarios, se reducen los costos de procesamiento. (escope, 2020).

En base a la información que se recolecte mediante las estadísticas de tráfico y los procesos establecidos para determinar el modelo de dimensionamiento se procederá a ocupar dichos datos para implementar los servicios de DNS y Web caché dentro de la

hiperconvergencia de Cisco en base a su arquitectura junto a los servicios que ya están en funcionamiento como lo es un DHCP, DNS y NTP.

La relación costo-beneficio del proyecto se establecerá mediante los resultados obtenidos al implementar los servicios antes mencionados, de tal forma que, se pueda realizar una comparativa de un antes y después en las estadísticas de tráfico, y así, demostrar si existiese una mejora significativa, o no la hubo en los parámetros de dichas estadísticas, y, por tanto, esto, significaría mayor eficiencia en la red.

1.5. Justificación

En base al Art. 18 del LOES donde especifica que: “El ejercicio de la autonomía responsable permitirá la ampliación de sus capacidades en función de la mejora y aseguramiento de la calidad de las universidades (Asamblea Nacional, 2018).” y con base a la acreditación de la Universidad Técnica Del Norte certificada por el CACES, donde la institución recibió el certificado de acreditación el 26 de Octubre del 2020 en la cual se consolidó el plan de aseguramiento de la calidad UTN, tomando en cuenta que, en uno de sus procesos se detalla el desarrollo tecnológico y la prestación de servicios, entonces, la institución requiere implementar el proyecto presentado para mejorar la eficiencia de la red del campus “El Olivo” y al mismo tiempo solventar los problemas de conectividad que se han venido suscitando, mediante el levantamiento de nuevos servicios.

La necesidad de mejorar constantemente los tiempos de respuesta en las solicitudes de los servicios web de cualquier institución ha provocado que se busquen medios eficientes que cumplan con los requerimientos de su red, al mismo tiempo disminuir el ancho de banda que se usa. A nivel institucional esto es muy importante, ya que disminuir el consumo de ancho de banda significaría la posibilidad de conectar a más usuarios, con una calidad de servicio optima de acuerdo con sus necesidades. Un aproximado de 12000 usuarios, contando entre

estudiantes, docentes y administrativos son quienes se ven beneficiados con el presente proyecto directamente y/o indirectamente usuarios de campus externos pues con los servidores Web y DNS Caché se puede llevar a cabo dicha optimización, ya que es posible obtener de manera local los recursos que se solicitan a una página web (común) o previamente accedida por los usuarios (Gómez, Sepúlveda, & Candela, 2012).

CAPÍTULO II. MARCO TEÓRICO

2.1. Introducción

En este capítulo se va a detallar la información bibliográfica que nos permitirá adquirir las bases fundamentales del trabajo de investigación, la cual otorga validez para su ejecución. Se abarca principalmente con la definición de Redes de Datos, World Wide Web y Sistema de Nombres de Dominio (DNS) así como también la definición de protocolos como HTTP, servidores de caché tanto WEB como DNS, características y funcionamiento, la base de este apartado se establece mediante los recursos bibliográficos que ofrecen varios modelos de dimensionamiento, esto con el fin de establecer los recursos de información teórica para la metodología de investigación propuesta en el diseño.

2.2. Redes de Datos

Desde el inicio de la existencia del ser humano, una de sus necesidades prioritarias ha sido la de comunicarse, por lo que siempre ha buscado y encontrado formas de hacerlo y de esta forma poder transmitir pensamientos y necesidades hacia otros de su especie. Tras el análisis de una línea de tiempo en el desarrollo del hombre se evidencia que la creación de herramientas y tecnología has sido y son fundamentales para establecer una comunicación eficiente y rápida, las redes de comunicación de datos permiten que una persona pueda transmitir y emitir información a sus semejantes hacia cualquier parte del mundo a velocidades casi instantáneas.

2.2.1. Definición de redes de datos

Se define a una red de datos como un conjunto de dos o más elementos que tienen la capacidad de comunicarse entre sí, es decir, que pueden intercambiar información en forma de datos a través de un medio. Las redes a las que nos referimos conectan dispositivos

electrónicos terminales y así también a sus usuarios. Por ejemplo, en un entorno local (LAN[1]) de trabajo, sus usuarios comparten información y recursos que competen dentro de la empresa, y de esta forma aprovechan estos mecanismos de comunicación (Liberatori, 2018).

2.2.2. Clasificación de las Redes

Las redes pueden clasificarse en base a distintos conceptos, su extensión, los dispositivos que las confirman o en base a las tecnologías, podríamos clasificarlas en Redes de Área Local (LAN) y Redes de Área Amplia (WAN) (Liberatori, 2018).

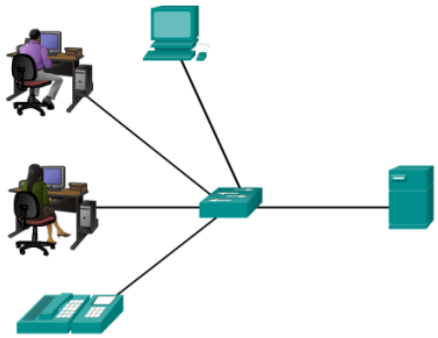
2.2.2.1. Redes de Área Local

Son aquellas que ocupan un área pequeña, por ejemplo, edificios o un conjunto de ellos, por lo general hablamos de redes que funcionan en base a compartir recursos, tales como dispositivos de almacenamiento, impresoras, teléfonos IP, etc. Se caracterizan por mantener una conexión interna, es decir, entre computadoras personales, repetidoras o hubs y conmutadores o switches tal como se muestra en la Figura 1. Los Routers son dispositivos que dan salida a una LAN hacia una WAN (Liberatori, 2018).

[1] LAN: Red de Área Local; por sus siglas en inglés Local Area Network

Figura 1

Topología de una Red de Área Local.



Nota. Adaptada de Cisco Networking Academy. *Exploración de la red*. Recuperado el 3 de junio del 2022 de https://www.uv.mx/personal/angelperez/files/2019/02/CCNA_ITN_Chp1.pdf

2.2.2.2.Redes de Área Amplia.

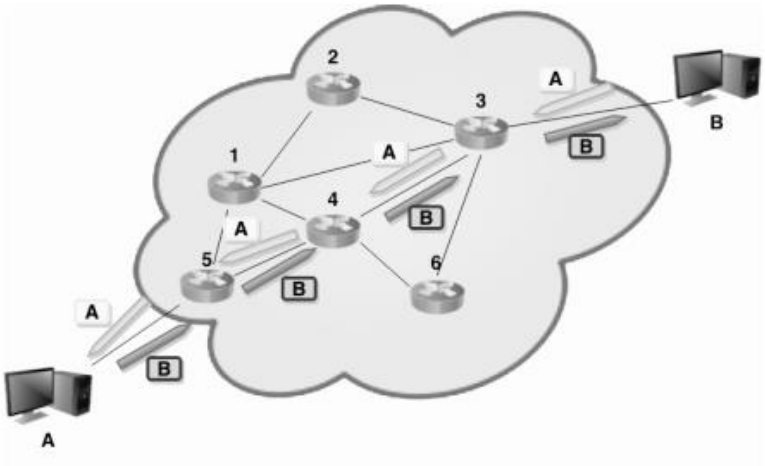
Como su nombre lo especifica, las redes WAN[2] o redes de área amplia son aquellas que cubren áreas geográficamente grandes, estas redes necesitan atravesar rutas públicas y por lo general hacen uso de circuitos que son proporcionados por un proveedor de telecomunicaciones (Liberatori, 2018).

La WAN se conforma por dispositivos de conmutación de paquetes de datos y es, de hecho, su característica principal de configuración. Toda la información se transmite y se encamina a través de nodos cuya función es proporcionar un servicio de conmutación hasta que cada paquete llegue a su destino como se muestra en la Figura 2 (Liberatori, 2018).

Figura 2

Conmutación de paquetes (orientado a conexión).

[2] WAN: Red de Área Amplia; por sus siglas en inglés Wide Area Network.



Nota. Adaptada de Mónica C. Liberatori. *Redes de datos y sus protocolos*. Recuperado el 3 de junio de 2022 de: http://www2.mdp.edu.ar/images/eudem/pdf/redes%20de%20datos.pdf

2.3. World Wide Web

También se la conoce como la web, se conforma de todos los sitios o páginas web públicas a los que un usuario accede mediante un host[3] a través de Internet, todas estas se interconectan por medio de hipervínculos, que, para un usuario, es el enlace donde se obtiene la información de ese sitio o página representada en diferentes formatos como texto, imagen, video o audio, pero que técnicamente, vendría a ser la conexión entre cliente-servidor. No hay que confundir la www[4] con Internet, puesto que es parte de ella (Awati, 2022).

2.2.3. URL

Los URLs (Uniform Resource Locator) son identificadores que permiten acceder a recursos web. De la misma manera en la que los humanos usan direcciones para encontrar ubicaciones, los URLs le ayudan al navegador o a otras aplicaciones Web a encontrar una página o recurso web en la Internet (Guijarro, 2012). Se conforma de los siguientes

[3] Host: dispositivo de usuario final; terminal anfitrión.
[4] WWW: World Wide Web

elementos: protocolo HTTP[5], subdominio, dominio, TLD[6] ruta, parámetro y etiqueta (Pukocz, 2019).

Figura 3

Ejemplo de URL.

Nota. Adaptada de Johnson D. *What is a URL?*. Recuperado el 3 de junio del 2022 de: https://www.businessinsider.com/guides/tech/what-is-a-url

2.2.4. Página y Sitio Web

Una página web es el nombre de un documento o información electrónica adaptada para la World Wide Web y que puede ser accedida mediante un navegador para mostrarse en un monitor de computadora o dispositivo móvil. Dicha información se localiza por lo general en formato HTML o XHTML, y suministra la navegación a otras páginas web mediante enlaces de hipertexto.

Una página web se la consideraría una tarjeta de presentación digital, para cualquier tipo de usuario, empresa u organización, así como una tarjeta de presentación de ideas y de información. De la misma forma, la tendencia orienta a que las páginas web no sean sólo atractivas para sus usuarios, sino también optimizadas para los buscadores mediante el código fuente. Forzar esta doble función puede, sin embargo, crear conflictos respecto de la calidad del contenido.

[5] HTTPS: Hypertext Transfer Protocol
[6] TLD: Top Level Domain

En inglés web site, un sitio web es un sitio (localización) en la World Wide Web que contine documentos (páginas web) organizados jerárquicamente. Cada documento (página web) contiene texto y o gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos. Cada sitio web tiene una página de inicio (en inglés Home Page), que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un navegador.

2.2.5. Funcionamiento de la Web

Cualquier host conectado a la web puede establecerse como cliente o como servidor, de aquí parte el establecimiento de transmisión y recepción de datos, el cliente solicita a los servidores información y estos a su vez transmiten respuestas.

Un cliente por definición es aquel dispositivo que está conectado a Internet y accede a él mediante un navegador o una aplicación que me permita acceder a sitios web, en cambio, un servidor es aquel dispositivo que los almacena. Así, de forma general, cuando el cliente hace una solicitud hacia este servidor, empieza a descargarse una copia del sitio desde el servidor hasta el dispositivo cliente.

Cuando un cliente desea ver un sitio web específico el primer paso es confirmar la existencia de este, después se envía una solicitud hacia su servidor, se procesa la respuesta, se renderiza la página y se muestra el sitio directamente en el cliente (Pardo, 2021).

2.2.5.1.Resolución de URL

Los sitios web se identifican por tener una dirección única, de esta forma cada cliente puede acceder a ellos; al ingresar a un sitio web el cliente solicita un código al servidor, el

cual se proporciona para establecer la conexión entre cliente servidor, este “código” se le conoce como URL[7].

Cuando se obtiene la URL el navegador de internet envía una solicitud de conexión hacia el servidor web del sitio al que se intenta ingresar, este a su vez contiene el código necesario para ser detectado por una dirección IP que al igual que la URL es una dirección única identificada por el protocolo IP[8] el cual proporciona la información para localizar el sitio web.

El proceso de transformar un dominio a una dirección IP empieza por el acceso hacia un servidor específico conocido como DNS[9] cuya función es la de hacer todo este proceso de “conversión” (Pardo, 2021).

2.2.5.2. Envío de solicitud

Cuando se resuelve y se identifica la dirección IP, el navegador envía una solicitud de acceso hacia el servidor que es localizado con esa IP, el protocolo HTTP es aquel que permite transmitir la información de solicitud, pues, establece los procesos de configuración de una solicitud y una respuesta.

Una vez el servidor recibe la solicitud, este envía una respuesta en el mismo formato, no siempre regresa un sitio web, a veces la respuesta es algún archivo o imagen pues hay dominios que dirigen hacia una descarga directa, esto depende del desarrollador (Pardo, 2021).

[7] URL: Uniform Resource Locator
[8] IP: Internet Protocol Access.
[9] DNS: Domain Name System.

2.2.5.3. Análisis de respuesta y visualización del sitio web

Cuando se obtiene la respuesta desde un servidor, el navegador web la procesa mediante el formato establecido por HTTP. El navegador analiza el tipo de información conforme a la información que recibe.

Según el portal Britannica, el código que se maneja para esta respuesta se basa en HTML[10] el cual es el encargado de estructurar la página web, depende de este proceso para que el navegador defina que tipo de información se debería mostrar en pantalla, es decir, información como imágenes, títulos, texto, etc.

2.2.6. Servicios Web

Si bien es cierto, existen múltiples definiciones de lo que son los servicios web, pues es complejo dar una definición exacta de dicho término debido a la vasta información que implica un servicio en la web, sin embargo, se podría decir que de forma general vendría a ser un conjunto de aplicaciones y de tecnologías con la capacidad de operar dentro de la Web, estas tecnologías se enfocan en la transmisión de datos entré sí y de esta forma entregar un servicio, un proveedor ofrece estos servicios mediante procedimientos remotos, entonces los usuarios pueden solicitarlo haciendo un llamado a estos procedimientos mediante la Web.

Este conjunto de sistemas se encarga de establecer la transmisión de información de solicitud y respuesta entre servidores o aplicaciones, actualmente se manejan sistemas compatibles entre diversos lenguajes de programación en los que fueron desarrolladas. En pocas palabras estos servicios permiten “la comunicación de host a host y el intercambio de información mediante paquetes de datos entre servidores y aplicaciones a través de Internet” (Carranza, 2021)

[10] HTML: Hyper Text Markup Language.

2.2.7. Protocolos y estándares

Un estándar es un conjunto de reglas que establecen los requisitos que debe cumplir un producto, un proceso o un servicio, con la finalidad de establecer mecanismos que son la base para permitir que tanto hardware como software sean compatibles entre los sistemas que los utilicen.

Entre los estándares web que más se utilizan hasta la actualidad destacaremos los siguientes:

- Identificador de Recursos Uniforme (URI, URL);
- Hypertext Transfer Protocol (HTTP);
- Lenguaje de Marcado de Hipertexto (HTML);
- Lenguaje Marcado Extensible (XML);
- File Transfer Protocol (FTP);
- Simple Mail Transfer Protocol (SMTP);
- WS-Security (Web Service Security);
- REST (Representational State Transfer);
- GraphQL (arquitectura alternativa a REST).

2.2.7.1. SOAP

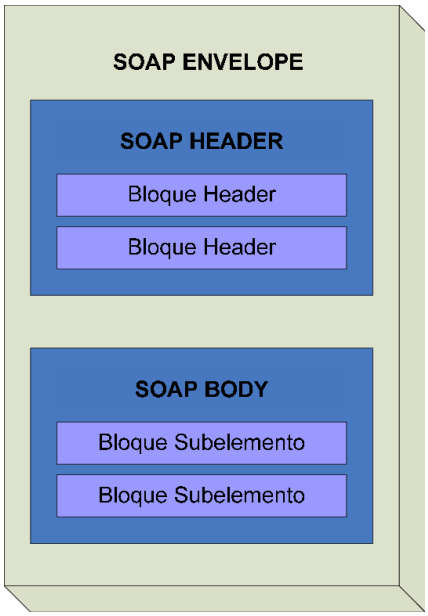
El protocolo SOAP[11] se basa en XML[12] y es el encargado de dar formato para el envío y recepción de mensajes que requieren las aplicaciones para intercambiar información, se diseñó para la comunicación en Internet.

El mensaje SOAP se compone por un Envolve (sobre) el cual tiene un Header (cabecera) y un Body (cuerpo) como lo muestra la figura 4:

[11] SOAP: Simple Object Access Protocol
[12] XML: Extensible Markup Language

Figura 4

Estructura de un mensaje SOAP.



Nota. Adaptada de Pérez J. *Descubrimiento de servicios para la evolución dinámica de sistemas software mediante transformación de modelos*. Recuperado el 4 de junio del 2022 de: https://www.researchgate.net/figure/Figura-8-Estructura-de-un-mensaje-SOAP-La-cabecera-o-header-es-opcional-y-sirve-para-dar_fig3_39657147

2.2.7.2. WSDL

WSDL (Web Services Description Language) también basado en XML es un lenguaje que da los requisitos necesarios para establecer la intercomunicación entre servicios web, describe cómo acceder a ellos.

2.2.7.3.UDDI

UDDI (Universal Description, Discovery and Integration) es un estándar XML es el directorio conde las empresas pueden buscar y registrar servicios web para comprobar la disponibilidad de estos.

2.4. Protocolo HTTP

El Protocolo de Transferencia de HiperTexto es un estándar desarrollado en 1999 por la World Wide Web Consortium, es un protocolo de tipo "cliente-servidor" que establece los intercambios de datos entre los clientes Web y los servidores HTTP. La especificación completa de HTTP 1/0 está establecida en el RFC 1945. Este estándar permite el proceso de realización de peticiones de datos y/o recursos que pueden ser documentos HTML, es la base de todo intercambio de información en la Web, en otras palabras, toda petición es iniciada por el mismo elemento que desea recibir los datos, que se le conoce como cliente o usuario final, que usa una aplicación de navegación Web.

Se basa en los servicios de conexión TCP/IP, y funciona igual que el resto de los servicios de los entornos UNIX[13]: un servidor escucha en un puerto de comunicaciones TCP y aguarda las solicitudes de conexión de los clientes Web. Una vez que se dispone la conexión, el protocolo TCP se ocupa de mantener la comunicación y permitir un intercambio de información libre de errores. (Guijarro, 2012).

2.4.1. Funcionamiento (Conexión TCP)

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, escogiendo un enlace de un documento HTML o digitándola directamente en el campo de búsqueda del cliente Web;
- El cliente Web descodifica la URL, separando sus diferentes partes. Entonces reconoce el protocolo de acceso, la dirección IP del servidor, el puerto opcional (puede ser 80 pero existen más) y el objeto que se necesita del servidor;

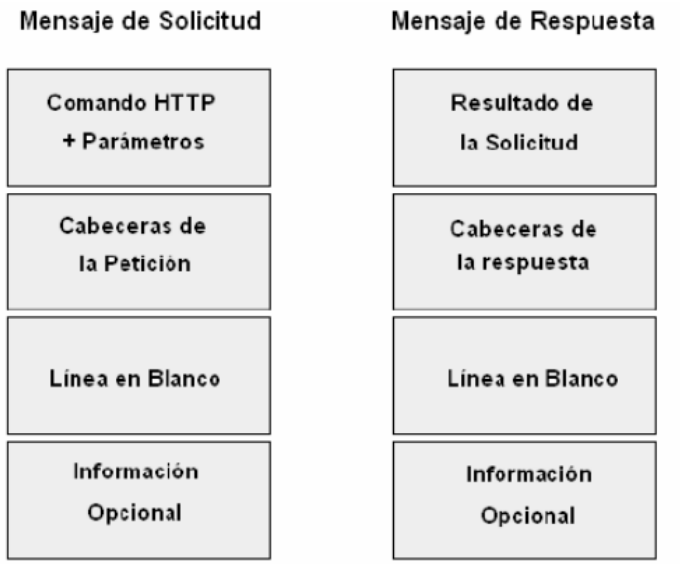
[13] UNIX: Registrado oficialmente como UNIX, sistema operativo portable, multiusuario y multitarea.

- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP que corresponde, se hace la petición. Para ello, se envía un comando, el que sea necesario (GET, POST, HEAD, etc), la dirección del objeto que se requiere (el contenido de la URL junto a la dirección del servidor), la versión del protocolo HTTP que usa (por lo general HTTP/1.0) y un conjunto de parámetros de información, que incluye datos sobre las capacidades del buscador o datos opcionales del servidor;
- El servidor devuelve la respuesta al cliente. Este proceso consiste en un código de estado y el tipo de dato MIME de la información que retorna, seguido de la información requerida;
- Se cierra la conexión TCP;

La conexión con los servidores HTTP se realiza a través de mensajes que se forman por líneas de texto, cada una contiene los distintos comandos y opciones del protocolo. Sólo existen dos tipos de mensajes, uno para realizar peticiones y otro para devolver la respuesta (Donate, 2017). La estructura de los dos tipos de mensajes en forma general se puede apreciar en la figura 5:

Figura 5

Tipos de mensajes de diálogo entre servidores HTTP.



Nota. Adaptada de Donate F. *Transmisión de imágenes de vídeo mediante Servicios Web XML sobre J2ME*. Recuperado el 4 de junio del 2022 de: https://biblus.us.es/bibing/proyectos/abreproy/11214/fichero/TOMO+I%252F05+Capitulo+5+Protocolo+HTTP.pdf

La primera línea del mensaje de solicitud contiene el comando solicitado del servidor HTTP y la respuesta contiene el resultado de la operación, que es un código que indica su éxito o el fracaso. Luego, para ambos tipos de mensajes, hay un conjunto de encabezados (algunos obligatorios, otros opcionales) que condicionan y definen cómo debe comportarse el protocolo.

La separación entre las líneas individuales del mensaje se realiza mediante un par CR-LF (retorno de carro avance de línea). El final del encabezado está marcado con una línea en blanco, seguido posiblemente de datos transportados por el protocolo, como un documento HTML devuelto por el servidor.

2.4.2. Métodos HTTP

HTTP define 8 métodos que indican la acción que desea realizar en el recurso definido. Lo que representa este recurso, ya sean datos preexistentes o datos generados

dinámicamente, depende de la aplicación host. Normalmente, un recurso corresponde a un archivo o resultado de un ejecutable ubicado en el servidor:

- **GET**: se utiliza para recopilar todo tipo de información del servidor. Se utiliza cada vez que se hace clic en un enlace o se ingresa directamente una URL. Como resultado, el servidor HTTP envía el documento correspondiente a la URL.;
- **HEAD**: similar al comando GET y puede usar el mismo encabezado, pero requiere que el servidor solo envíe el encabezado de respuesta. Lo utilizan los administradores de caché de página o los servidores proxy para saber cuándo actualizar la copia almacenada en el archivo;
- **POST:** funciona de manera idéntica al comando HEAD, pero, además, envía datos informativos al servidor, generalmente provenientes de un formulario web para la administración del servidor o la adición de una base de datos (Guijarro, 2012);
- **PUT:** carga un recurso (archivo) en particular es la forma más eficiente de cargar un archivo en el servidor porque el método POST usa un mensaje de varias partes y el servidor decodifica el mensaje. Por el contrario, el método PUT le permite escribir archivos cuando hay una conexión de socket al servidor. La desventaja del método PUT es que no está habilitado en los servidores de alojamiento compartido;
- **DELETE:** Borra el recurso especificado;
- **TRACE:** este método le dice al servidor que reenvíe un mensaje de respuesta en el cuerpo de la entidad con todos los datos recibidos del mensaje de solicitud. Se utiliza para pruebas y diagnóstico;

- **OPTIONS:** devuelve los métodos HTTP admitidos por el servidor para la URL especificada. Esto se puede usar para probar la funcionalidad del servidor web bajo demanda en lugar de en un recurso específico**;**
- **CONNECT:** Este método está reservado para su uso con proxies. Esto permitirá que el servidor proxy se convierta automáticamente en un túnel. Por ejemplo, para comunicarse con SSL (Donate, 2017).

El protocolo HTTP es común en el mundo de Internet, y cada usuario de Internet tiene un navegador web a través del cual puede conectarse a un servidor web desplegado sin hacer nada más que una simple solicitud de una página web. Por lo tanto, se decidió utilizar el protocolo HTTP para la comunicación cliente-servidor (Donate, 2017).

2.5. Sistema de Nombres de Dominio (DNS)

El Sistema de Nombres de Dominio (DNS) es un sistema descentralizado, escalable y distribuido globalmente. Proporciona una base de datos dinámica que combina las direcciones IP de las computadoras, servicios o cualquier recurso conectado a Internet o una red privada con varios tipos de información. Admite tanto IPv4 como IPv6 y la información se almacena como registros de recursos (RR) de varios tipos que pueden almacenar direcciones IP u otra información. Esta información se agrupa en zonas correspondientes a un espacio de nombres o dominio y la proporciona el servidor DNS autorizado de la misma zona.

Básicamente, el DNS es responsable de convertir las direcciones IP de los recursos de la red en nombres fáciles de recordar y legibles por humanos, y viceversa. Esta operación se llama "resolución DNS". Establece un mecanismo amigable para la localización e identificación de activos. La analogía de la guía telefónica se usa a menudo cuando el número relevante se puede ubicar por su nombre o viceversa. En esta comparación, los números

representarán direcciones IP y los nombres representarán registros espaciales de dominio (López, s.f.).

2.5.1. Elementos del DNS

DNS se estructura en tres componentes principales:

- Espacio de dominios de nombres: Consiste en una estructura de árbol jerárquico donde cada nodo contiene 0 o más registros (registros de recursos o RR) con información de dominio. Desde el nodo raíz en el nivel superior, las ramas forman las regiones antes mencionadas, que pueden contener uno o más nodos o dominios, que a su vez se pueden dividir en subdominios a medida que asciende en la jerarquía;
- Servidores de Nombres: Estos son los servidores responsables de mantener y proporcionar información sobre el espacio de nombres o dominio. Por un lado, están los servidores que almacenan información completa para uno o más conjuntos de espacios de nombres (dominios) y son responsables de ellos. Dicen que son servidores competentes de estas zonas o dominios. Por otro lado, hay otro tipo de conjuntos de almacenamiento del servidor de diferentes dominios que reciben consejos con los servidores competentes apropiados (búsqueda recursiva). Esta información se mantiene temporalmente en la localidad (caché) y la extiende periódicamente. Son los servidores de caché. A través de los servidores de nombres y sus interacciones, se logra la distribución y redundancia del espacio de nombres de dominio. Con la organización de estos servidores de nombres y su capacidad para conectar, distribuir y redundar el espacio de nombres de dominio.

- Resolvers: estos son servidores de caché o programas cliente que se encargan de enviar las solicitudes necesarias y recuperar la información solicitada para proporcionar al usuario solicitante (López, s.f.).

2.5.2. Espacio de Dominios de Nombres, Jerarquía y Sintaxis

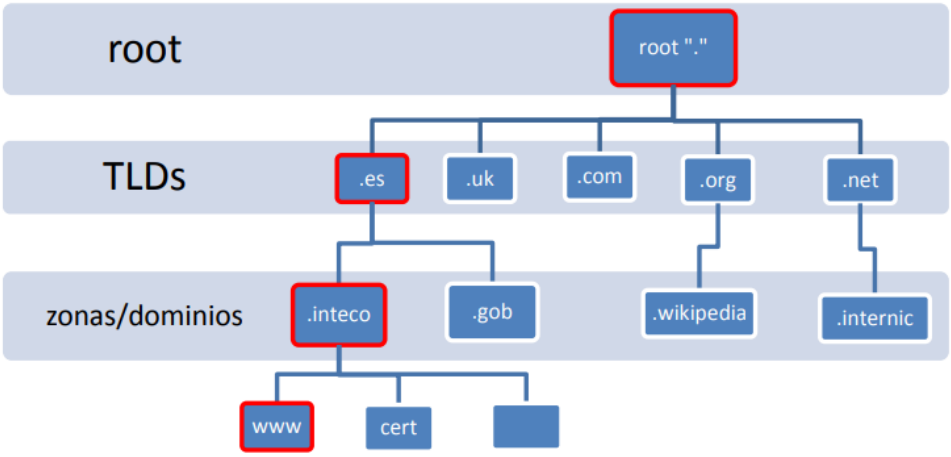
2.5.2.1. Estructura Jerárquica

El DNS consta de espacios de nombres de dominio organizados en una jerarquía similar a un árbol donde los nodos están interconectados, cada uno de los cuales representa un nivel de espacio de dominio. El nivel superior de toda la jerarquía es el dominio raíz o directorio raíz representada por "." (punto). Solo un nivel por debajo es un dominio de nivel superior o TLD. Son, a su vez, nodos padres de otros niveles inferiores, conocidos como TLD de segundo nivel. Luego, la jerarquía continúa hasta que se alcanza el último nodo que representa el recurso. Un nombre que consta de una cadena completa se denomina nombre de dominio completo (FQDN).

Una zona es parte de un espacio de nombres de dominio que tiene derechos administrativos delegados a un servidor DNS que actúa como "autoridad" para esa sección o dominio. Este servidor se denomina servidor autorizado para la zona. La jerarquía comienza con la zona raíz "." es el nivel superior. Aunque no se muestra normalmente, cada nombre de dominio completo termina con un "." indica el final de un espacio en el área principal. Por ejemplo, "www.example.com" es en realidad "www.example.com", donde el punto a la derecha al final representa la región principal. Este nombre de dominio completo se denomina nombre de dominio completo (FQDN).

Figura 6

Jerarquía de dominios DNS.



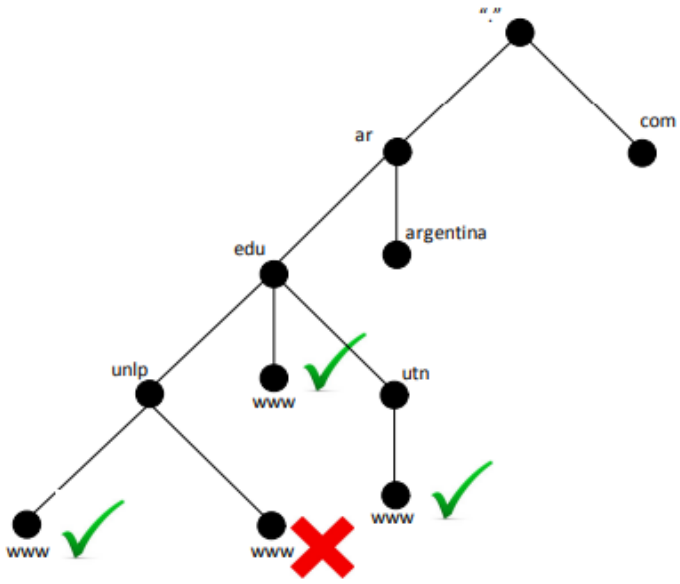
Nota. Adaptada de A. López. Guía de Seguridad en Servicios. Recuperado el 4 de junio del 2022 de: https://galleton.net/index.php/es/libros-pdf/libros-de-computacion/item/17653-guia-de-seguridad-en-servicios-pdf-antonio-lopez-padilla

2.5.2.2. Espacio de nombres

Si hay independencia entre caminos, se puede repetir el nombre. Este es el caso de la etiqueta www, que a menudo se repite como el nombre del nodo en las hojas, pero en diferentes rutas de árboles en la jerarquía de nombres. La Figura 6 ilustra esta situación. Allí, puede ver el caso del mismo nombre en la misma ruta, esto no está permitido; y una instancia repetida del mismo nombre en otro camino. Este último solapamiento de nombres está permitido dado que se cumple la independencia de caminos:

Figura 7

Inconsistencia de nombres en el sistema DNS.



Nota. Adaptada de Río N. (2015). Diseño e implementación de una solución de administración de tráfico de red basada en DNS y chequeos de disponibilidad.

Los nombres de dominio, incluidos los servidores de nombres, están estructurados de forma lógica, a diferencia de la organización de las direcciones IP que normalmente sigue a la organización física o geográfica. Este tipo de organización permite que dos servidores dependientes de un mismo dominio (por ejemplo, www1 y www2 en ejemplo.com.ar) estén en redes diferentes, incluso geográficamente separadas. Además, otros dominios, llamados subdominios, pueden estar separados de cada dominio. Esta estructura define el árbol del sistema de nombres. Una manera fácil de determinar si un dominio es parte de otro es usar una estructura de nombres jerárquica. Si se encuentra algún punto común en la ruta raíz, podemos concluir que ambos subdominios son parte de un dominio. Por ejemplo, el dominio ejemplo.com.ar es un subdominio del dominio com.ar.

Cada dominio también puede denominarse subdominio de nivel N, donde N es la distancia medida por el número de nodos en el árbol que deben viajar para llegar a la raíz desde ese nodo. Por lo tanto, se ha determinado que la raíz "." (Punto) tiene nivel 0. Los dominios de nivel 1 provienen del directorio raíz y su número es limitado.

Este tipo de dominio se usa comúnmente para delegar la administración de la base de datos a países, organizaciones y otros fines que contienen nombres lógicamente relacionados con actividades específicas. Aunque el sistema de nombres no contiene ninguna regla sobre cómo se deben nombrar los dominios, hay una serie de dominios de primer nivel que se han tomado como referencia para construir el árbol. (Río, 2015)

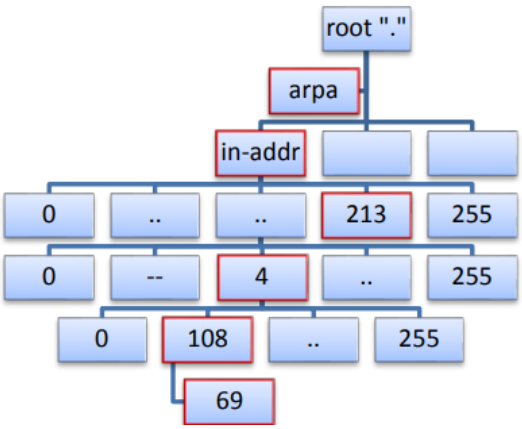
2.5.2.3. Espacio de dominio de direcciones IN-ADDR.ARPA

En DNS, el dominio in-addr.arpa se utiliza para definir el espacio de direcciones IP. Este dominio garantiza que la dirección IP se vuelva a traducir al nombre correspondiente, lo que facilita su búsqueda en Internet.

Los subdominios en in-addr.arpa están estructurados con hasta 4 etiquetas (IP versión 4), cada una de las cuales representa un octeto de una dirección IP. Por ejemplo, la información sobre la dirección IP 213.4.108.69 estaría en el dominio 69.108.4.213.in-addr.arpa. Obsérvese como se sigue el criterio jerárquico en la Figura 8.

Figura 8

Dominio 69.108.4.213.in-addr.arpa.



Nota. Adaptada de López A. (s.f.). Guía de Seguridad en Servicios

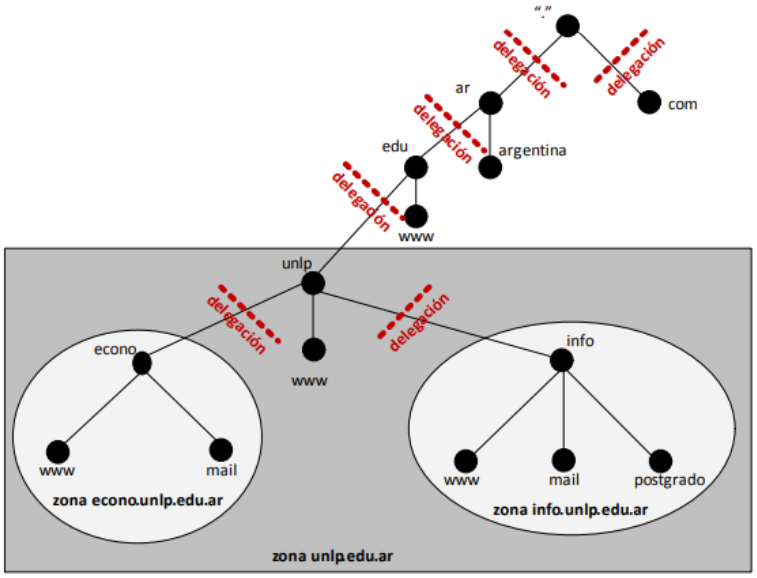
2.5.3. Servidores DNS y zonas

Toda la información sobre zonas, dominios y registros de permisos es administrada por el servidor de nombres de dominio. En la mayoría de los casos, cada servidor de nombres contiene una parte de la base de datos de resolución, denominada zona DNS, de la que es responsable. Se considera que un servidor tiene autoridad para una región en particular si controla una parte de la base de datos correspondiente al dominio representado por esa zona. Los servidores DNS pueden ser proxy para una o más zonas. Además, tiene información que lo vincula con otros servidores de arbitraje, lo que le permite reconocer otros nombres para los que no tiene autoridad ni información en su propia base de datos. Cada dominio DNS se puede dividir en diferentes zonas.

Cada región corresponde a un subdominio del dominio anterior. Además, las zonas pueden ser administradas por el mismo servidor DNS que tendrá autoridad sobre ellas, o pueden ser delegadas a nuevos servidores DNS que autogestionarán parte de la facilidad de resolución autorizada y tiene autoridad de resolución de nombres. La figura 9 muestra un ejemplo de diagrama de autorización de dominio y zona:

Figura 9

Esquema de dominios de DNS y delegación de zonas.



Nota. Adaptada de Río N. (2015). Diseño e implementación de una solución de administración de tráfico de red basada en DNS y chequeos de disponibilidad.

La especificación DNS establece dos tipos de servidores de nombres: servidores primarios y servidores secundarios. El servidor DNS principal tiene información sobre los permisos para la zona especificada porque el administrador del servicio especificado ingresó la configuración manualmente. Este es el servidor donde se realizan los cambios en los registros de diferentes resoluciones. El servidor secundario recibe información de permisos del servidor principal. En este último caso, el administrador no actualiza la información de configuración y permisos obtenida del servidor primario en un concepto conocido como transferencia de zona.

Siempre que haya un cambio en la zona principal, se notificará a la subregión para que actualice la información de permisos en la parte de la base de datos que ha cambiado en comparación con la zona principal. Además, el servidor secundario sondeará periódicamente al servidor principal para determinar si se han producido cambios en la región sobre la que no se ha notificado. En este caso, el servidor secundario moverá la zona modificada del servidor primario. Puede tener tantos subservidores como desee configurar (Río, 2015).

La información de permisos se puede almacenar en el servidor mediante diferentes métodos, según el software utilizado. Microsoft proporciona software compatible con el servicio DNS que almacena la configuración en el registro de Windows. Otra implementación del protocolo DNS es la proporcionada por el software BIND, que será el tema central de este artículo. BIND almacena datos en archivos en el sistema de archivos donde se ejecuta el servicio. Cada archivo es un área de permiso y tiene un diseño similar al de la imagen a continuación:

Figura 10

Configuración de zona de DNS en servidor BIND.

```
ejemplo.com.ar.          IN SOA  dns1.ejemplo.com.ar. root.ejemplo.com.ar. (

199609260  ; serial

28800      ; refresh (8 hours)

7200       ; retry (2 hours)

2419200    ; expire (4 weeks)

86400      ; minimum (1 day)

)
```

Nota. Adaptada de Río N. (2015). Diseño e implementación de una solución de administración de tráfico de red basada en DNS y chequeos de disponibilidad.

La primera fila de la tabla indica que el servidor de nombres es la mejor fuente de información para la zona ejemplo.com.ar. Esto significa que el servidor dado tendrá autoridad para esta zona. La primera columna de la fila contiene el nombre de dominio completo, seguido de "." (Punto) denota su dependencia de la raíz. La segunda columna contiene la palabra reservada IN, que indica que el dominio es un dominio de Internet. La tercera columna se refiere al tipo de registro, que en este caso es SOA (Autorización de inicio) y está precedida por el nombre del servidor DNS que tiene autoridad para la zona, que en este caso

parece ser dns1. por ejemplo .com.ar. Finalmente, el campo root.example.com.ar se traduce a la dirección de correo electrónico root@example.com.ar, que es la dirección de correo electrónico para escribir si desea contactar al administrador de la zona DNS. Seguido a esta primera línea se encuentran 5 (cinco) valores los cuales son (Jeftovic, 2015):

- Serial;
- Refresh;
- Retry;
- Expire;
- Mínimum;

Los datos mencionados, definen lo que se conoce como encabezado de un archivo de zona de DNS.

2.5.4. Tipos de registros

Cada línea en el archivo de zona de servidores DNS identifica lo que se llama un registro de recursos (RR). Varias líneas, como se discutió en la sección anterior, definen los ajustes de configuración generales para el comportamiento de la zona. Otros definen registros que permiten asociaciones entre nombres y valores. Los nombres definidos en el archivo de zona no distinguen entre mayúsculas y minúsculas, por lo que los administradores pueden definirlos como mejor les parezca.

En general, por convención, se utilizan todas las letras minúsculas. Aunque los registros se presentan en un orden específico en el DNS RFC, esto no es obligatorio. Los tipos de registro se pueden mezclar en un archivo de zona, aunque a menudo se definen convenciones, como agrupar todos los tipos de registro y especificar el orden alfabético por grupo. Cada tipo de registro tiene un propósito y uso específico (Río, 2015).

Los tipos más comunes de registro que pueden definirse son:

- SOA (Start Of Authority o Autoridad de la zona): contiene información sobre el servidor DNS principal de la zona, la dirección de correo electrónico del administrador del dominio, el número de serie del dominio y la actualización o la hora de actualización. La información de este registro se describe en el apartado anterior;
- NS (Name Server o Servidor de Nombres): define la relación que existe entre un nombre de dominio y un servidor de nombres que contiene información sobre un dominio en particular. Cada dominio puede estar asociado con uno o más servidores de nombres. Especifique qué servidores de nombres autorizados para la zona;
- MX (Mail Exchange o registro de intercambio de correo): Asocia un nombre de dominio con una lista de servidores de intercambio de correo para ese dominio. Le permite equilibrar la carga y priorizar el uso de uno o más servicios de correo electrónico;
- A (Address o dirección): Este registro se usa para traducir nombres de servidores a direcciones IPv4;
- CNAME (Canonical Name o Nombre canónico): Se utiliza para crear un nombre de host o alias para un servidor en un dominio. Se utiliza cuando varios servicios (como correo e Internet) se ejecutan en un único servidor con la misma dirección IP.;
- PTR (Pointer o indicador): También conocido como registro inverso, funciona en orden inverso a los registros A, traduciendo las direcciones IP en nombres de dominio.

2.6. Servidor de Caché

Un servidor de almacenamiento en caché es un servidor o servicio web dedicado que actúa como un host local de páginas web u otro contenido web. Al colocar la información previamente solicitada en la memoria caché o temporal, un servidor de caché acelera el acceso a los datos y reduce las necesidades de ancho de banda de una empresa. Los servidores de caché también permiten a los usuarios acceder a contenido sin conexión, incluidos archivos multimedia u otros documentos. El servidor de caché a veces se denomina mecanismo de caché.

Un servidor de caché es casi siempre un proxy, es decir, un servidor que representa al usuario al interceptar sus solicitudes de Internet y administrarlas para el usuario. Por lo general, esto se debe a que los recursos corporativos están protegidos por un firewall. Este servidor permite solicitudes salientes, pero filtra todo el tráfico entrante. Los servidores proxy ayudan a hacer coincidir los mensajes entrantes con las solicitudes salientes. Por tanto, también puede almacenar archivos recibidos para su posterior recuperación por parte de cualquier usuario. Servidores proxy y almacenamiento en caché de usuarios; Todas las solicitudes y respuestas de Internet provienen de una ubicación específica en Internet. (El proxy no es completamente invisible, su dirección IP debe especificarse como parámetro de configuración a un navegador u otro programa que admita el protocolo (TechTarget, 2019).

2.6.1. Caché DNS

La caché de DNS (también conocida como caché de resolución de DNS) es un almacenamiento de DNS temporal en los dispositivos (computadoras, teléfonos inteligentes, servidores, etc.) del dominio visitado (un registro para la dirección IPv4, un registro AAA para la dirección IPv6).) etcétera). Almacena estos registros en función de su vida útil (TTL).

Cada vez que visite el sitio web, su dirección se almacenará en esta base de datos temporal para facilitar las visitas posteriores. Básicamente, el almacenamiento en caché de DNS es la forma que tiene un dispositivo de tratar de ahorrar tiempo y esfuerzo y evitar largas búsquedas de DNS al responder consultas de DNS con una entrada de DNS que ya está en el caché de DNS temporal.

La caché del servidor DNS almacena información sobre las consultas de DNS durante un período de tiempo específico llamado TTL (tiempo de vida) de cada entrada de DNS. Los servidores de caché optimizan el uso de la red al reducir el tráfico DNS de Internet porque mantienen los registros solicitados y pueden servirlos directamente sin repetir consultas recursivas. Además, reducen la carga en los servidores autorizados, especialmente en la zona raíz o servidor (Pramatarov, 2021).

2.6.2. Caché Web

El almacenamiento en caché del servidor proxy o el almacenamiento en caché web reduce el uso de ancho de banda y mejora la velocidad y la confiabilidad del sitio web al proporcionar un punto de presencia para uno o más servidores de contenido internos. Los proxies de almacenamiento en caché se pueden configurar como un proxy inverso o un proxy directo que proporciona un punto de acceso a un servidor web interno o externo responsable de reducir los tiempos de solicitud y respuesta. El proxy inverso es la configuración predeterminada.

El servidor proxy intercepta las solicitudes de datos del cliente, extrae la información solicitada de las computadoras que alojan el contenido y devuelve ese contenido al cliente. Por lo general, las solicitudes de documentos se alojan en un servidor web y se envían mediante HTTP (protocolo de transferencia de hipertexto). Estos servidores web también se conocen como servidores de origen o servidores de contenido. Sin embargo, puede configurar

el servidor proxy para que funcione con otros protocolos, como el Protocolo de transferencia de archivos (FTP).

El servidor proxy almacena el contenido en caché en el caché local antes de enviar el contenido al solicitante. Los ejemplos de contenido en caché incluyen páginas web estáticas. El almacenamiento en caché permite que el servidor proxy responda a solicitudes posteriores del mismo contenido sirviéndolo directamente desde el caché local, mucho más rápido que volver a descargarlo desde el host de contenido (IBM, 2022).

Existen 3 tipos de almacenamiento en caché los cuales se les considera como caché web que son:

- Caché del Navegador*:* es creado por el usuario (User-Agents). Este caché se llama caché privado porque lo crea un solo navegador. Este caché se crea cuando un usuario realiza una solicitud del navegador a un sitio web, y en esa primera solicitud se recopilan todos los datos sin necesidad del servidor;
- Caché del Intermediario: este tipo de caché lo crea un proveedor de servicios de Internet (ISP), que es un intermediario cliente-servidor. Esto se denomina caché compartida (caché de proxy de reenvío) porque admite la misma vista de página para varios usuarios;
- Caché del Servidor: La caché del servidor también se conoce como caché del puerto (caché de proxy inverso). Este tipo de almacenamiento en caché es diferente del almacenamiento en caché privado del usuario o del almacenamiento en caché público que depende directamente del host de la red. Puede usar diferentes tipos de cachés de puerto, como usar un CDN o paquetes como Varnish Caché. (Deyimar, 2022).

2.6.2.1.Control de los caches web.

El protocolo HTTP define tres mecanismos básicos para controlar las cachés:

Frescura: permite que la respuesta se use sin volver a verificar el servidor original y puede ser controlada tanto por el servidor como por el cliente. Por ejemplo, el encabezado de respuesta Expires contiene la fecha de vencimiento del documento, y la directiva Cache-Control: max-age le dice al caché cuántos segundos debe ser válida la respuesta.

Validación: se puede usar para verificar si la respuesta almacenada en caché es buena después de que caduque. Por ejemplo, si la respuesta tiene un encabezado Última modificación, la memoria caché puede realizar una solicitud condicional utilizando el encabezado If-Modified-Since para ver si la página ha cambiado.

Invalidación: generalmente es un efecto secundario de otra solicitud que pasa por el caché. Por ejemplo, si la URL asociada con la respuesta almacenada en caché se solicita posteriormente con una solicitud POST, PUT o DELETE, la respuesta almacenada en caché se invalidará. (MDN contributors, 2022).

CAPÍTULO III. ANÁLISIS DE SITUACIÓN ACTUAL Y DISEÑO

En el presente capítulo se realiza el análisis de la situación actual de la red, enfocando los apartados pertinentes al caso de investigación, es decir, la topología de red, direccionamiento, VLAN's, servicios, etc. Finalmente mediante el desarrollo del diseño en base a un modelo de dimensionamiento.

3.1. Situación Actual de Red

Este apartado contiene la información de la situación actual de la red institucional de la Universidad Técnica del Norte, se presenta la topología Física y Lógica del data center, se hace una auditoría de los servicios de DNS levantados actualmente.

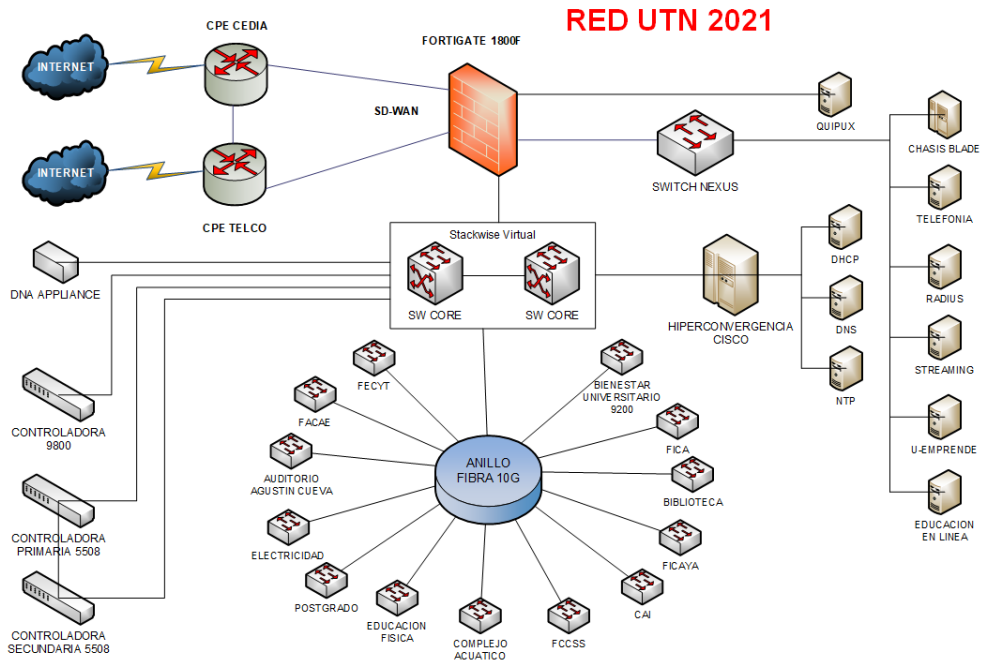
A continuación, se detallan las consideraciones más importantes para establecer los antecedentes y especificar el análisis previo al diseño y levantamiento de nuevos servicios, en este caso del caché de DNS y Web.

3.1.1. Topología física y equipos existentes.

Esta topología detalla la disposición física donde se encuentran los equipos de red existentes como se muestra en la Figura 11 donde se menciona los elementos de hardware conectados entre sí.

Figura 11

Topología Física UTN.



Nota. Adaptada de Dirección de Desarrollo Tecnológico e Informático UTN

3.1.1.1. Descripción

Actualmente, la Universidad Técnica del Norte mantiene conexión hacia Internet mediante dos proveedores, un enlace de fibra óptica que provee CEDIA[14] y el proveedor principal TELCONET.

La institución cuenta con un cuarto de telecomunicaciones, dentro del departamento de desarrollo tecnológico e informático ubicado en el edificio central, desde allí se interconectan las diferentes facultades por medio de enlaces directos de fibra óptica.

La red se dispone mediante una jerarquización de capas independientes, una capa de núcleo, capa de distribución y capa de acceso, facilitando de esta forma la administración del tráfico interno de la institución.

La capa de core se ubica directamente en el data center y se compone de los dos switches de núcleo principales, en la parte de distribución cada switch se encuentra en cada

[14] CEDIA: Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia.

facultad o edificio donde los puntos de tráfico de datos final ya se encuentran distribuidos en los switches de acceso.

Como parte fundamental cabe mencionar que los servicios de DHCP, DNS y NTP están siendo administrados mediante un sistema HCI[15] establecido por equipos CISCO.

3.1.2. Distribución lógica de segmentos de red.

En la siguiente tabla se establece la distribución de los segmentos de red de capa tres, se indica cómo los equipos se comunican entre sí dentro de la red universitaria mediante la configuración lógica y la distribución de VLAN's.

Tabla 1.

Distribución de VLAN's UTN.

N°	DESCRIPCIÓN	VLAN	DIRECCIÓN IP	MASCARA DE SUBRED	GATEWAY
1	EQUIPOS-ACTIVOS	1	172.16.x.x	255.255.254.0	172.16.x.x
2	DMZ	2	10.24.x.x	255.255.255.0	10.24.x.x
3	CORE-FW	3	172.16.x.x	255.255.255.0	172.16.x.x
4	EQUIPOS-WIRELESS	4	172.16.x.x	255.255.254.0	172.16.x.x
5	CCTV	6	172.16.x.x	255.255.255.0	172.16.x.x
6	RELOJES-BIOMETRICOS	7	172.16.x.x	255.255.255.0	172.16.x.x
7	TELEFONIA-IP-ELASTIX	8	172.16.x.x	255.255.252.0	172.16.x.x
8	AUTORIDADES	12	172.16.x.x	255.255.255.0	172.16.x.x
9	DDTI	14	172.16.x.x	255.255.255.0	172.16.x.x
10	FINANCIERO	16	172.16.x.x	255.255.255.0	172.16.x.x
11	COMUNICACION-ORGANIZACIONAL	18	172.16.x.x	255.255.255.0	172.16.x.x
12	ADMINISTRATIVOS	20	172.16.x.x	255.255.255.0	172.16.x.x
13	ADQUISICIONES	22	172.16.x.x	255.255.255.0	172.16.x.x
14	U-EMPRENDE	24	172.16.x.x	255.255.254.0	172.16.x.x
15	AGUSTIN-CUEVA	26	172.16.x.x	255.255.255.0	172.16.x.x
16	BIENESTAR-DOCENTES	28	172.16.x.x	255.255.255.0	172.16.x.x
17	BIENESTAR-ADMINISTRATIVOS	30	172.16.x.x	255.255.255.0	172.16.x.x
18	CLUBES-UTN	32	172.16.x.x	255.255.255.0	172.16.x.x
19	NATIVA	39	----------	----------	----------

[15] HCI: Infraestructura Hiperconvergente.

20	FICA-LABORATORIOS	40	172.17.x.x	255.255.254.0	172.17.x.x
21	FICA-WIRELESS	42	172.17.x.x	255.255.255.0	172.17.x.x
22	FICA-ADMINISTRATIVOS	44	172.16.x.x	255.255.255.0	172.16.x.x
23	FICAYA-LABORATORIOS	48	172.17.x.x	255.255.254.0	172.17.x.x
24	FICAYA-ADMINISTRATIVOS	52	172.16.x.x	255.255.255.0	172.16.x.x
25	FECYT-LABORATORIOS	56	172.17.x.x	255.255.254.0	172.17.x.x
26	FECYT-ADMINISTRATIVOS	60	172.16.x.x	255.255.255.0	172.16.x.x
27	FACAE-LABORATORIOS	64	172.17.x.x	255.255.254.0	172.17.x.x
28	FACAE-ADMINISTRATIVOS	68	172.16.x.x	255.255.255.0	172.16.x.x
29	FCCSS-LABORATORIOS	72	172.17.x.x	255.255.254.0	172.17.x.x
30	FCCSS-ADMINISTRATIVOS	76	172.16.x.x	255.255.255.0	172.16.x.x
31	POSTGRADO-LABORATORIOS	80	172.17.x.x	255.255.254.0	172.17.x.x
32	POSTGRADO-ADMINISTRATIVOS	84	172.16.x.x	255.255.255.0	172.16.x.x
33	CAI-LABORATORIOS	88	172.17.x.x	255.255.254.0	172.17.x.x
34	CAI-ADMINISTRATIVOS	92	172.16.x.x	255.255.255.0	172.16.x.x
35	BIBLIOTECA-LABORATORIOS	96	172.17.x.x	255.255.254.0	172.17.x.x
36	BIBLIOTECA-DOCENTES	98	172.16.x.x	255.255.255.0	172.16.x.x
37	BIBLIOTECA-ADMINISTRATIVOS	100	172.16.x.x	255.255.255.0	172.16.x.x
38	EDUROAM	128	172.20.x.x	255.255.224.0	172.20.x.x
39	WIRELESS-EVENTOS	160	172.21.x.x	255.255.248.0	172.21.x.x
40	SERVICE DNA	192	172.23.x.x	255.255.240.0	172.23.x.x
41	COPIADORA	201	172.24.x.x	255.255.255.0	172.24.x.x
42	HX-INBAND-MGMT	202	172.25.x.x	255.255.255.0	172.25.x.x
43	HX-STORAGE-DATA	203	172.25.x.x	255.255.255.0	172.25.x.x
44	HX-VMOTION	204	172.25.x.x	255.255.255.0	172.25.x.x
45	HX-VM-NETWORK	205	172.25.x.x	255.255.255.0	172.25.x.x
46	CLÚSTER DNA	208	172.23.x.x	255.255.240.0	172.23.x.x
47	IPS-PUBLICAS	211	-	-	-

Nota. Adaptada de Dirección de Desarrollo Tecnológico e Informático UTN

3.1.3.1. Descripción

La red universitaria se segmenta en 45 subredes distribuidas en cada una de las instancias del campus Universitario, cada Facultad cuenta con una subred de distribución

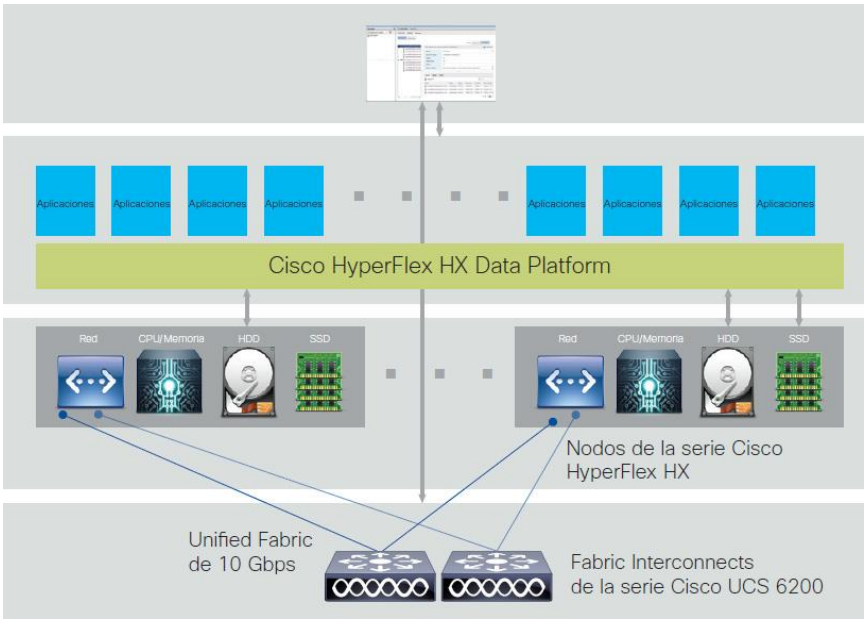
correspondientes a Laboratorios y Administrativos, a excepción de la FICA donde se establece una VLAN para su conexión inalámbrica.

3.1.3. Arquitectura hiperconvergente

Por naturaleza, la arquitectura de una plataforma en un sistema HCI tiene enfoque en el almacenamiento definido por software, actualmente la Universidad maneja la administración de los servicios DHCP, NTP y DNS mediante este sistema, la arquitectura se observa en la Figura 12:

Figura 12

Arquitectura de un sistema Hiperconvergente implementado por CISCO.



Nota. Adaptada de CISCO. (2016). *Si usted utiliza la virtualización de servidores, sufre silos de almacenamiento.*

Toda la estructura se monta en base a los conmutadores CISCO UCS 6454, que permiten una conexión estructurada de 54 puertos, este es un conmutador de canal de fibra, FCoE y 10/25/40/100 Gigabit Ethernet de una unidad de bastidor (1RU) que ofrece un rendimiento de hasta 3,82 Tbps y hasta 54 puertos. El conmutador tiene 28 puertos Ethernet

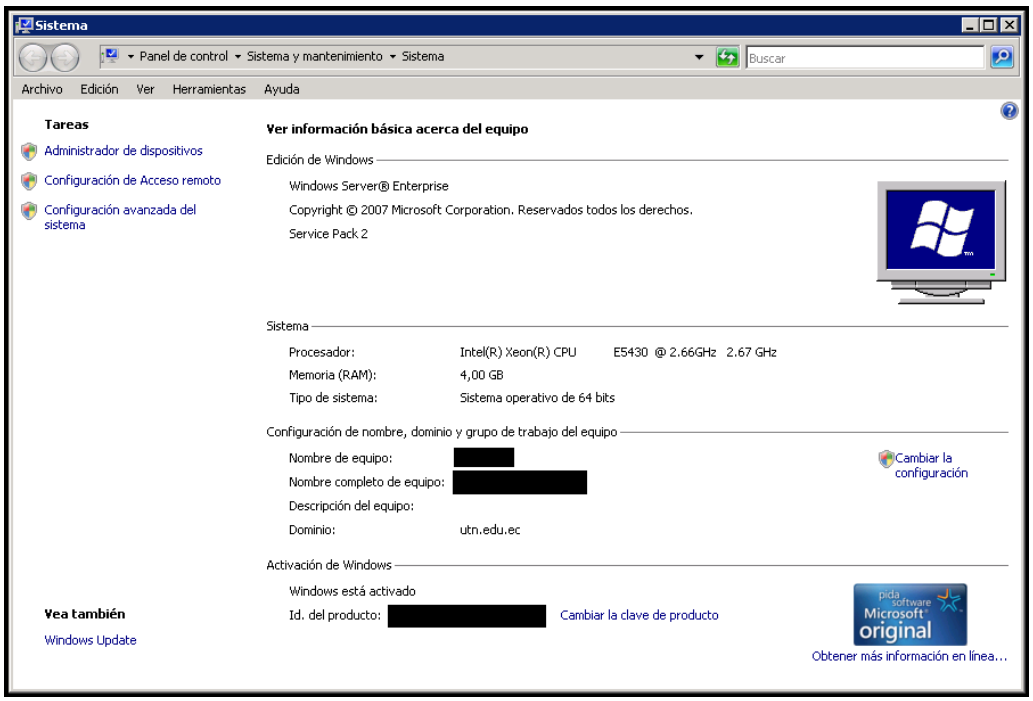
de 10/25 Gbps, 4 puertos Ethernet de 1/10/25 Gbps, 6 puertos de enlace ascendente Ethernet de 40/100 Gbps y 16 puertos unificados que pueden admitir puertos Ethernet de 10/25 Gbps u 8/ Puertos de canal de fibra de 16/32 Gbps. Todos los puertos Ethernet son compatibles con FCoE (CISCO, 2021).

3.1.4. Servidor DNS

La red de Infraestructura de la Universidad Técnica del Norte mantiene en funcionamiento actualmente un servicio de DNS mediante la Herramienta administrativa "DNS" montada en el sistema operativo Windows Server 2008.

Figura 13

Información de sistema operativo Windows Server 2008 donde está montado el servidor DNS.



Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

Cabe mencionar que actualmente, el servidor de DNS está instalado mediante un rol de Servicios de dominio de Active Directory, tanto el servidor maestro como esclavo, con direcciones IPv4 estáticas: 172.16.x.x. y 10.24.x.x

Para la configuración del servidor se ha tomado en cuenta los conceptos de: zona de búsqueda directa y zona de búsqueda inversa. Identificaremos cada uno de ellos, cómo están configurados y que registra cada uno.

3.1.4.1. Estructura DNS en Active Directory

En el servidor de DNS que se maneja actualmente se trabaja con la configuración de Microsoft Active Directory donde usa DNS para permitir que estaciones de trabajo y servidores ubiquen los servicios que se ejecutan en el espacio de nombres de Active Directory.

La localización de servicios de Active Directory dependen directamente de la disponibilidad de la zona DNS, en este caso se integra las zonas y los registros dependiendo del tipo de zona, pues el directorio puede integrar las zonas y registros de recursos como objetos de este. En este caso únicamente nos centraremos en la información con respecto a las zonas de búsqueda directa e inversa, ya que es la información que se utilizará en el desarrollo de este proyecto de tesis.

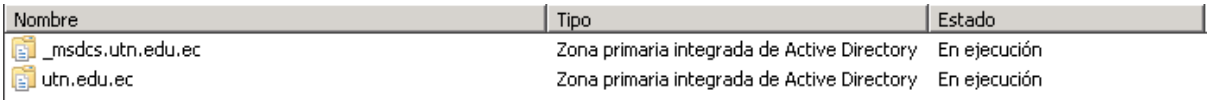
3.1.4.2. Zona de búsqueda directa

Una zona de búsqueda directa es simplemente una forma de resolver nombres de host en direcciones IP. Es decir, nos permite crear zonas primarias y secundarias o stub zones. Aquí es donde están establecidos los distintos tipos de registros como lo son: A, CNAME, MX, SRV, TXT, etc. Para asociarlos a una dirección IP, posteriormente, algún host previamente registrado, podrá realizar consultas al DNS, el cual devolverá la dirección IP en la que está alojado.

La configuración de nuestro servidor DNS está establecida por dos zonas primarias integradas de Active Directory como lo muestra la figura 14:

Figura 14

Zonas de búsqueda directa para servidor DNS UTN.

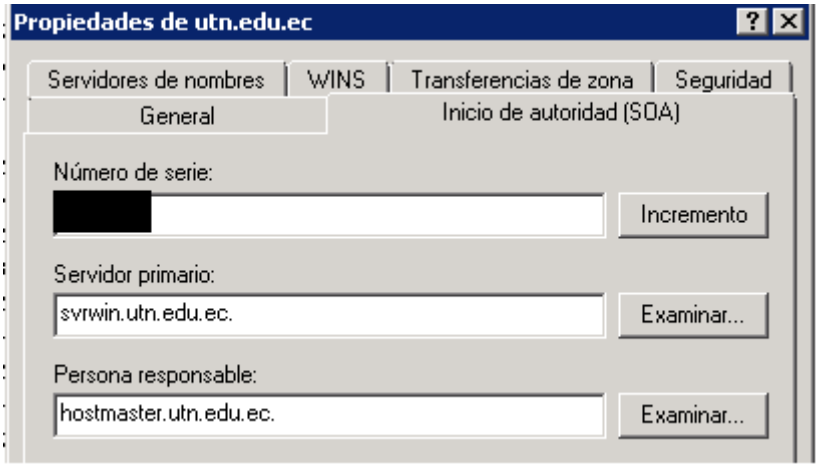


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

En este caso nos enfocamos en la zona primaria “utn.edu.ec” pues aquí se establecen cada uno de los registros tanto de host, administradores y servidor de nombre, en primer lugar está configurado un registro de Inicio de Autoridad (SOA) (Figura 15) el cual establece la información del administrador sobre el dominio, en este caso está establecido por el servidor primario de nuestro servidor “SVRWIN” con administrador “hostmaster” los cuales están ya registrados como dominios dentro del directorio de Active Directory:

Figura 15

Propiedades del registro SOA.

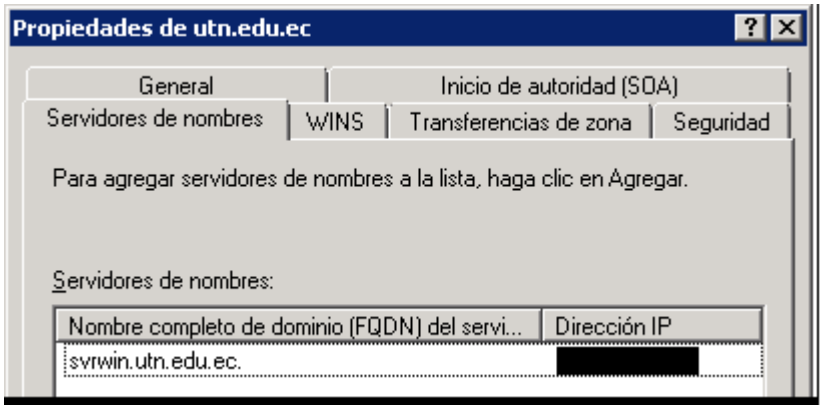


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

También se establece el servidor del Servidor de nombres (NS) donde se especifica directamente la dirección IP de dicho servidor mediante el subdominio del mismo, en este caso del servidor primario:

Figura 16

Propiedades del registro NS.

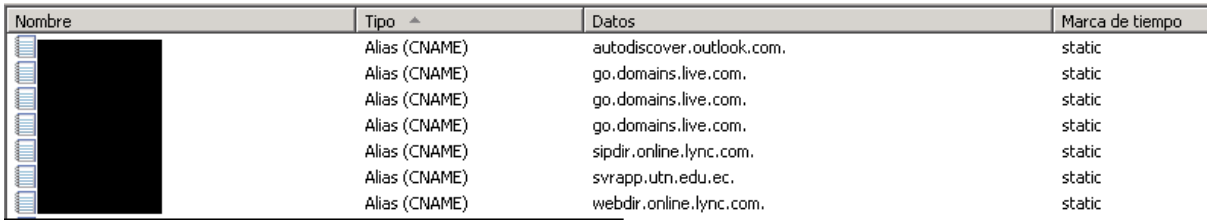


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

Así también como parte de la configuración del servicio DNS, se establecen los registros de CNAME los cuales reenvían un dominio o subdominio a otro dominio, cabe recalcar que este registro no proporciona una dirección IP, sino que realizan consultas a dominios externos, en este caso a servidores: live, Lync y Outlook. (Figura 17).

Figura 17

Registros CNAME.

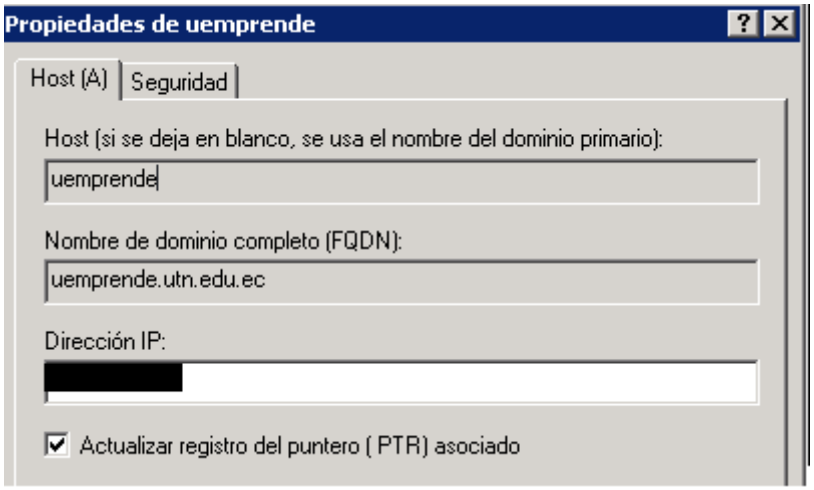


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

El servidor también configura una aproximado de 600 registros para host (A) los cuales contienen la dirección IP de dominios que pertenecen directamente a la institución, para este caso tomaremos el ejemplo del departamento de la "UEmprende" que se configura con su dominio como lo muestra la Figura 18:

Figura 18

Propiedades del registro A de la "uemprende".



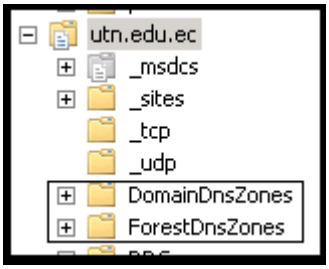
Nota. Adaptado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

Cabe recalcar, que cada departamento tiene su propia distribución de dirección IP para su registro A en el servidor DNS, como lo especifica la Tabla 1, donde se puede apreciar dicha información de direccionamiento, pero la mayoría de los registros de este tipo configurados en la zona de búsqueda "utn.edu.ec" pertenecen a host como PC, Impresoras o dispositivos que no tienen relevancia en el estudio del caso.

Finalmente, dentro de la zona, están configuradas las particiones que corresponden tanto a "DomainDnsZones" como a "ForestDnsZones" (Figura 19).

Figura 19

Particiones DomainDnsZones y ForestDnsZones.

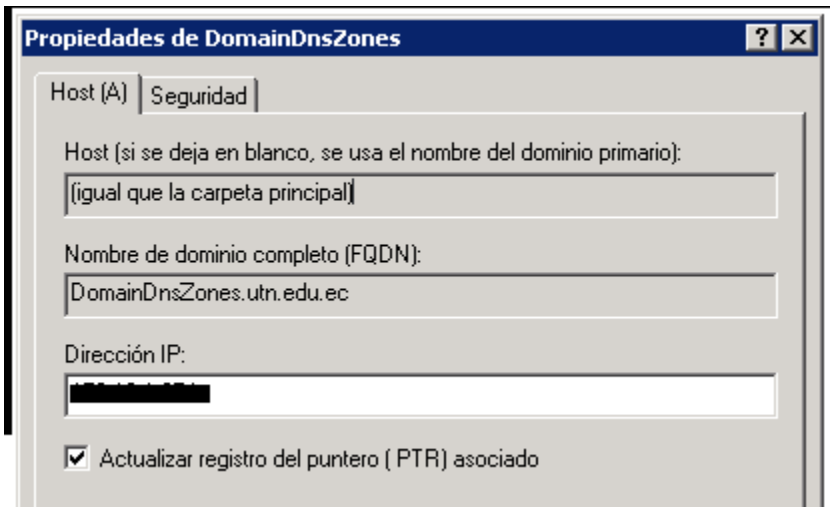


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

La partición de “DomainDnsZones” almacena la zona DNS del dominio y son únicos para cada dominio y todos los controladores de dominio que son servidores DNS en un dominio reciben una réplica de esta partición, en este caso se establece el registro A para la dirección IP del servidor DNS (Figura 20)

Figura 20

Propiedades del Registro A de la partición DomainDnsZones.



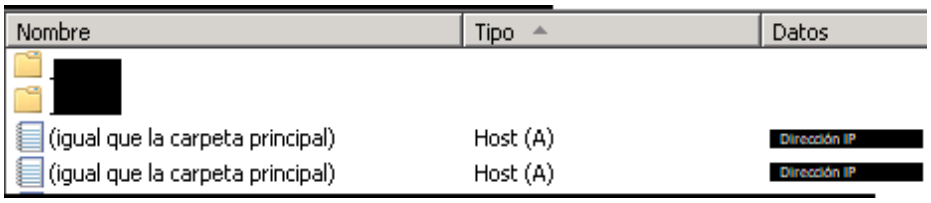
Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

En cambio, la partición “ForestDnsZones” Contienen los detalles de todos los servidores DNS que se ejecutan en los controladores de dominio del bosque. Las zonas DNS almacenadas en la partición del directorio de la aplicación se replican en todos los servidores

DNS que se ejecutan en los controladores de dominio del bosque, en este caso al tener dos servidores DNS, aquí se configuran los dominios a los registros A tanto de servidor maestro como esclavo (Figura 21),

Figura 21

Registros A de la partición ForestDnsZones.



Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

3.1.4.3. Zonas de búsqueda Inversa.

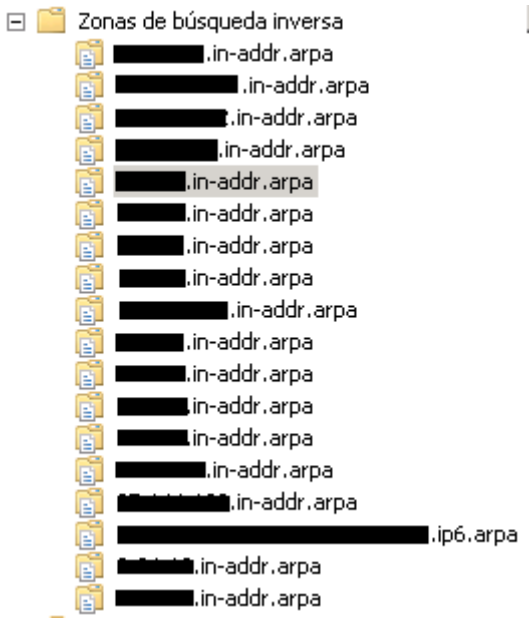
La zona inversa, nos permitirá por el contrario generar registros PTR los cuales establecen una dirección IP a un nombre. Las consultas inversas definen de que dominio es la IP consultada, es decir, se consigue lo contrario de la habitual búsqueda de DNS de reenvío, en la que se consulta al sistema de DNS para que devuelva una dirección IP.

Las búsquedas DNS inversas consultan a los servidores de DNS por un registro PTR (puntero); si el servidor no tiene un registro PTR, no puede resolver una búsqueda inversa. Los registros PTR almacenan las direcciones IP con sus segmentos invertidos, y le añaden ".in-addr.arpa". Por ejemplo, si un dominio tiene una dirección IP de 192.0.2.1, el registro PTR almacenará la información del dominio en 1.2.0.192.in-addr.arpa.

En nuestro servidor tenemos registros PTR para cada registro en las zonas de búsqueda directa, dicho de otra forma, para cada registro asignado a un segmento de red específico, existe su zona de búsqueda inversa (Figura 22).

Figura 22

Zonas de búsqueda inversa a cada segmento de red de los registros de Zona de búsqueda directa.

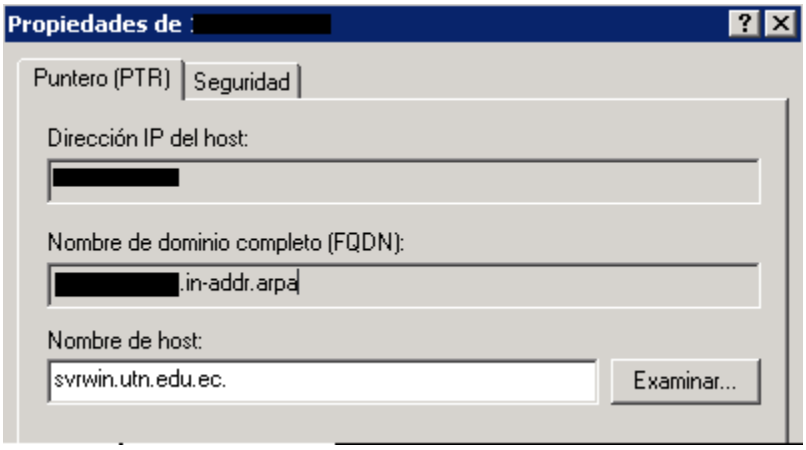


Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

Tomaremos como ejemplo, el registro PTR para el registro A del servidor DNS primario donde se especifica el dominio “x.x.x.x.in-addr.arpa” con nombre de host “svrwin.utn.edu.ec” (Figura 23).

Figura 23

Propiedades del registro PTR del host del servidor DNS primario.



Nota. Adaptada de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte

3.2. Diseño de servidor de caché

Para establecer los procesos de configuración de servidores DNS caché y Web caché, primero hay que entender su funcionamiento, ambos servicios están basados en una lógica similar, el usuario realiza consultas, en el caso de DNS caché, el sistema verifica si hay almacenada información con respecto a nombres de dominio, y en el caso de Web Caché, se verifica si hay contenido de páginas web almacenadas en caché, para devolverlas al usuario con mayor velocidad, ahorrando ancho de banda y permitiendo mayor eficiencia en la navegación web con tiempos de respuesta cortos.

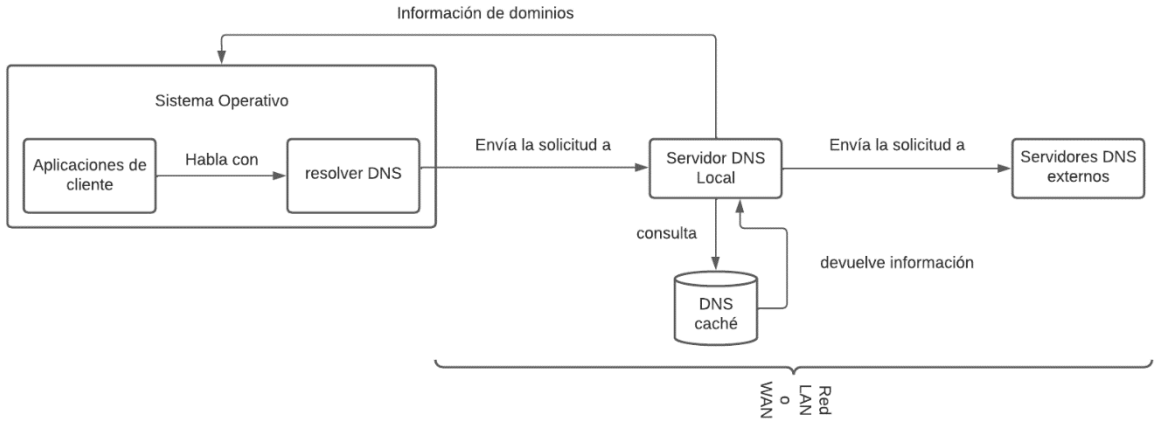
3.2.1. Arquitectura de servidor DNS caché

En el caso del DNS se establecen distintos tipos de caché, el que analizaremos es el caché de servidor, antes ya se mencionaba cómo se realiza el proceso de resolución de dominios, desde el cliente hacia los servidores DNS locales y externos, en la UTN, se manejan dos servidores de DNS, maestro y esclavo con exactamente la misma configuración a excepción de su dirección IP, el almacenamiento de caché sirve para identificar futuras consultas que previamente ya han sido identificadas por el sistema, de tal forma que la consulta posterior se la realiza directamente a dicha memoria de caché, es entonces que el

sistema resuelve la información de los dominios que han sido almacenados, caso contrario la consulta se extiende a servidores externos, al mismo tiempo que la almacena, una vez que recibe una respuesta así como lo muestra la Figura 24:

Figura 24

Proceso de resolución de nombres de dominio con consulta en caché.



Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

En nuestro caso, basaremos el método investigativo para la implementación de este servicio en software libre, y el diagrama antes presentado muestra la resolución de DNS en un sistema Linux, dándonos la ventaja de tener servicios que actúan como un solucionador de DNS.

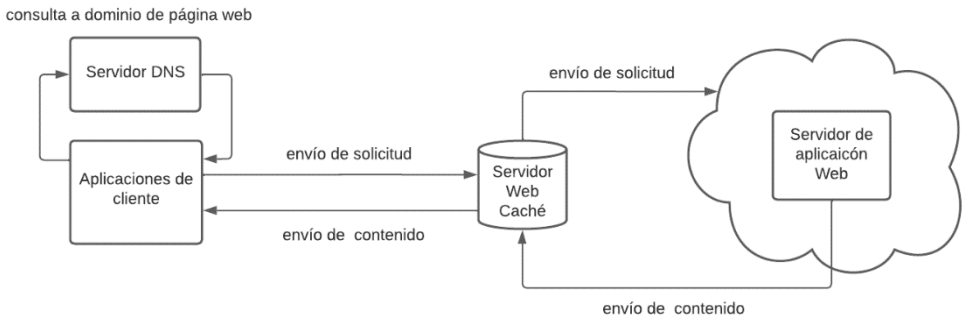
3.2.2. Arquitectura de servidor Web caché.

Como se mencionó anteriormente la lógica de funcionamiento de un Web caché es similar a la del caché de DNS, debido a que el proceso siempre empieza con una consulta hacia el servidor de DNS para identificar el dominio de un sitio web al que el cliente desea acceder, éste devuelve su respuesta con la dirección IP de dicho sitio, entonces la aplicación del cliente envía una solicitud hacia internet con esa información, aquí entra en juego nuestro

servidor de Web caché, pues primero la solicitud se procesa mediante este servicio, si en el almacenamiento de caché existe la información requerida por el cliente, este envía el contenido directamente para su visualización dando un "acierto de caché", si el Web cache no tiene el contenido solicitado o el contenido está obsoleto o no es válido, transfiere la solicitud al servidor web de la aplicación, devolviendo nuevamente la información requerida hacia el servidor Web caché y finalmente envía el contenido al cliente y almacena una copia de la página, este proceso se identifica en la Figura 25.

Figura 25

Proceso de solicitud y respuesta de un servidor de aplicación web con almacenamiento en caché de servidor.



Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

3.2.3. Dimensionamiento de servidores

En este apartado se establece los parámetros, métricas y procesos que se deben cumplir para satisfacer los requerimientos necesarios en el levantamiento de los servidores DNS y Web caché, y así, mantener un correcto funcionamiento.

Los servicios antes mencionados cumplen con la característica de tener similares procesos de funcionamiento, por tal motivo, el modelo de dimensionamiento que se

desarrollará a continuación es aplicable para los dos casos, así mismo, parte de la objetividad del proyecto es la utilización de software libre, por lo que los componentes, métricas y características que se van a desarrollar están basadas en sistemas operativos de Linux, con software que corresponde a esta especificación.

Para dimensionar los servidores se debe tener en cuenta principalmente el número de usuarios, y también se deben tomar en cuenta parámetros adicionales como las aplicaciones utilizadas por los usuarios, principalmente aplicaciones web, los procesos del sistema, el requerimiento de almacenamiento en disco duro y disponibilidad de memoria.

3.2.3.1. Metodología.

En un principio, se estableció el diseño en base a modelos matemáticos de evaluación de colas, sin embargo, debido a que en el estudio del caso no sería lo más eficiente aplicar modelos de simulación para la optimización del sistema, sino que la teoría es más aplicable en datos estadísticos por la naturaleza del servicio en almacenamiento en caché, la metodología de diseño para dimensionar nuestros servicios se basa en un modelo de aplicación directa, se comienza levantando dos servidores de prueba, de los cuales se va a recopilar información de consumo de recursos de hardware mediante un virtualizador en un periodo de ocho a quince días, de esta forma se procede al establecimiento de los requerimientos del sistema, la definición de métricas para evaluar los parámetros a ser dimensionados en los servidores, todo en base a un análisis de los datos estadísticos seleccionados de los servidores de prueba y finaliza con la configuración de las máquinas virtuales, este último punto será explicado a fondo en el “Capítulo 4” de nuestro tema de estudio.

3.2.3.2. Requisitos.

Los requisitos que deben cumplir los servidores para el presente proyecto son.

- Procesamiento de las peticiones simultáneas de los usuarios.
- Convergencia de los servidores maestro y esclavo.
- Escalabilidad en los servicios.
- Cada servidor manejará por separado sus procesos, disposición de almacenamiento y memoria.
- Los servidores se manejarán bajo licencias de software libre, mediante sistemas basados en Linux.

3.2.3.3. Métricas.

Los parámetros para considerar se basan en los siguientes criterios:

- Desde el punto de vista del usuario se toma en cuenta los tiempos de respuesta de las aplicaciones y del acceso a sitios web.
- Desde el punto de vista del servidor se debe evaluar parámetros de consumo del CPU, los procesos en cola de ejecución, la cuantía de memoria libre y la utilización del disco duro. Es decir, los parámetros que nos permitan establecer el consumo de recursos y los probables cuellos de botella que se puedan dar.
- De acuerdo con la carga específica hay que vincular los puntos de vista del usuario y del servidor, puesto que la relación entre ellos es mutua. La actividad de carga específica que hay en el servidor depende directamente de la ocupación de los usuarios, y es aquí donde se toma en cuenta el uso y disponibilidad de recursos.

3.2.3.4. Recopilación y Análisis de datos.

Como se hizo mención anteriormente, el primer paso para aplicar nuestra metodología es el levantamiento de servidores de prueba, en este caso se hace uso de dos servidores DNS

Caché, los cuales mantienen las mismas configuraciones, pero con diferente enfoque ya que uno es un servidor “maestro” y el otro “esclavo”.

Así también, se toma en cuenta el requisito de que el servidor debe estar basado en Linux, esto para ocupar mucho menos recursos, pero sin perder la operabilidad y eficiencia del servicio.

Previa investigación, la herramienta que se pone en funcionamiento es “Unbound” pues es la que más se apega a cumplir los requisitos que ocupa la metodología de dimensionamiento, los requerimientos del sistema y métricas. Cabe mencionar que los servidores únicamente se levantaron para pruebas del proceso actual de dimensionamiento, por lo tanto, no se especifica procesos de instalación y configuración del mismo, pues lo más relevante para cumplir el objetivo del capítulo presente es el análisis de las gráficas y estadísticas, los datos se obtienen aplicando el servicio a zonas específicas de VLAN que están configuradas dentro de la topología lógica de red.

Las gráficas que se analizan a continuación están tomadas del virtualizador “Zabbix” de los servidores de prueba, todos levantados en una máquina virtual con Sistema Operativo “Alma Linux”. Se analizan los tres parámetros principales para el dimensionamiento de un servidor, CPU, memoria y almacenamiento, para lo cual Zabbix nos permite virtualizar gráficas estadísticas para cada uno, finalmente se ha identificado como “DNSCACHE1” y “DNSCACHE2” a los servidores de prueba, tanto al principal como al secundario respectivamente.

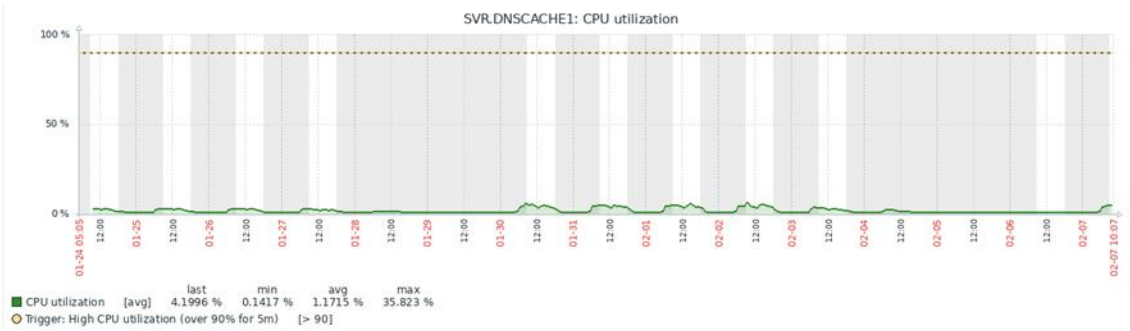
3.2.3.5. CPU

La CPU donde se levantaron los servicios mantiene un rendimiento máximo de 3.5 GHz especificada por el fabricante, esta frecuencia de reloj es la base en la cual vamos a fundamentar el análisis estadístico de los servidores, cabe mencionar que el periodo de

recolección de datos es exactamente de catorce (14) días, en primera instancia verificamos las gráficas del uso de CPU del DNSCACHE1 en la figura 26:

Figura 26

Uso de CPU del servidor 1.



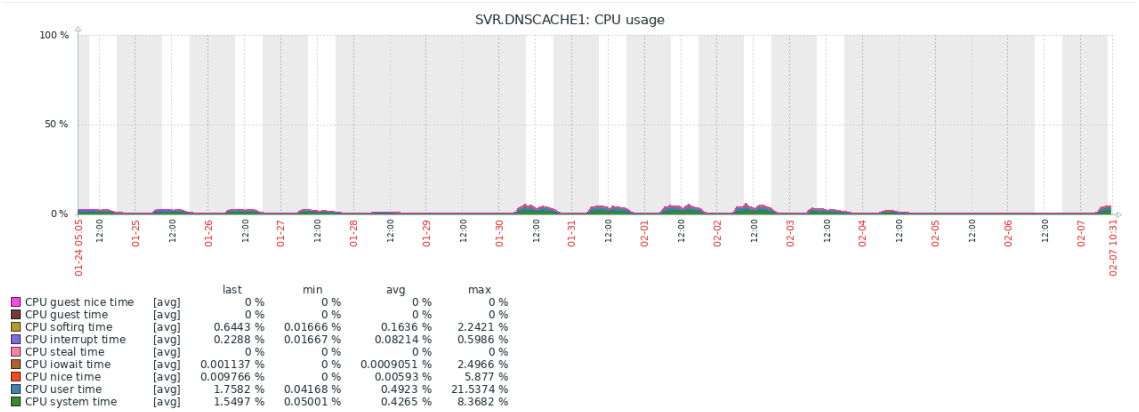
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

En primer lugar, Como se puede observar, el uso del CPU es prácticamente bajo, si tomamos de referencia los días de calendario, es lógico pensar que los días de mayor uso de este recurso es en días laborables, es por ello que la gráfica nos muestra cinco (5) picos de uso de CPU en un periodo semanal. Algo fundamental a tomar en cuenta es que el promedio en porcentajes es que el sistema usó un 1,18% en el trascurso de los catorce (14) días, pero hay que tomar en cuenta que en algún punto del tiempo hubo un pico donde se usó un 35.82% del CPU, un dato que hay que considerar tomando en cuenta el tamaño de la red de la institución.

De una forma más específica podemos analizar la Figura 27, donde se especifica más a detalle el uso de la CPU con este servidor.

Figura 27

Uso de CPU del servidor 1.



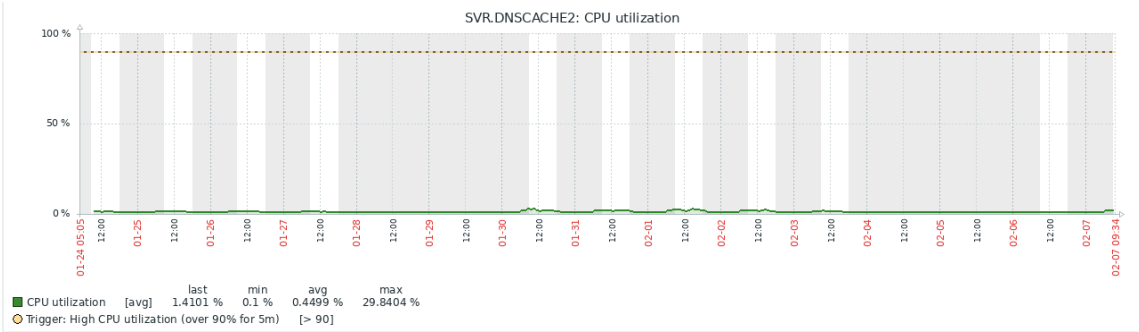
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Aquí se puede observar los datos de uso de CPU en el tiempo de uso donde se promedia un 0,49% pero nos dice que el uso máximo es del 21,53% así como también del tiempo del sistema de la CPU donde se tiene un promedio del 0,42% pero asciende al 8,37% en su valor máximo de consumo que son los valores más significativos que se tomará en cuenta para el dimensionamiento.

Ahora bien, vamos a identificar los valores que nos brinda la estadística del uso de CPU del servidor DNSCACHE2 en la figura 28:

Figura 28

Uso de CPU del servidor 2.

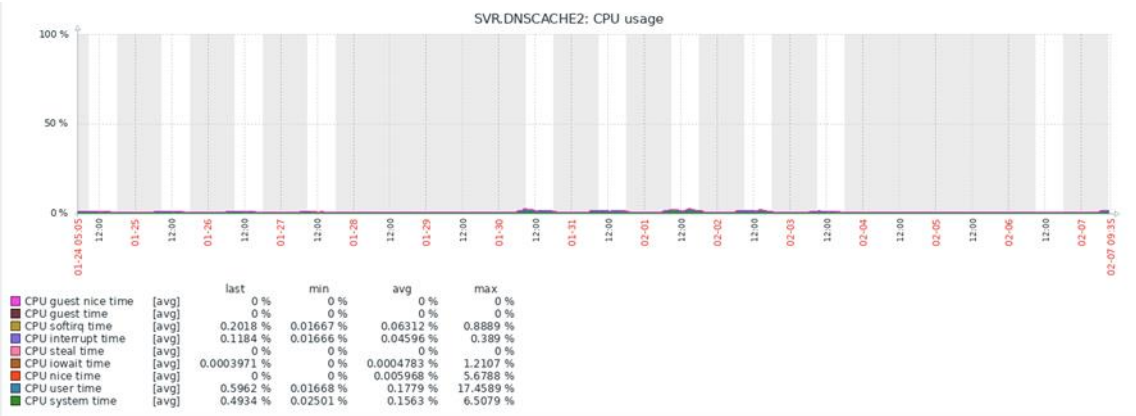


Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Al ser un servidor secundario o “esclavo” el uso de la CPU es mucho menor al servidor 1, es por ello que en esta gráfica notamos que el uso promedia un 0,45% aunque el valor máximo al que podría llegar es del 29,85%, verificamos más a detalle también otros datos se usaron del recurso en la Figura 29.

Figura 29

Uso de CPU del servidor 2.



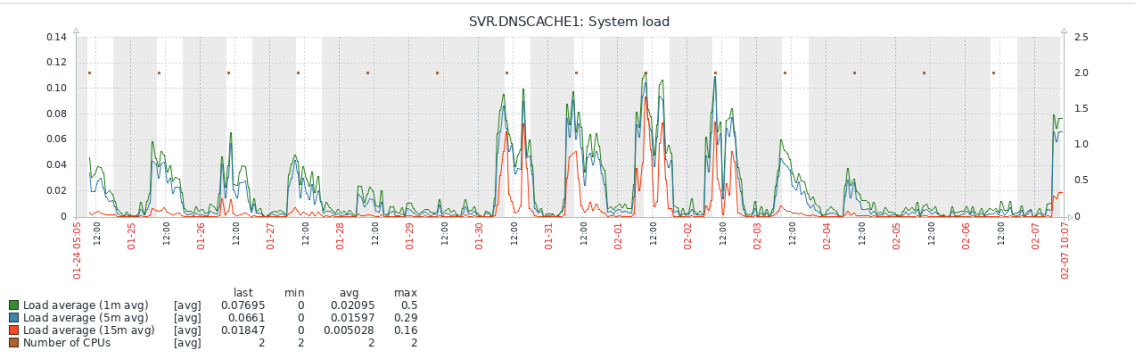
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Aquí también los datos de tiempo de uso de CPU promedian un 0,18% con un pico máximo del 17,5% y el tiempo del sistema ronda el 0,16% con picos que pueden llegar al 6,51% de uso de los recursos.

También se toma en cuenta un parámetro fundamental para establecer el uso de la CPU, en este caso notar en la figura 30 donde la carga del sistema del servidor 1 hace uso de dos (2) núcleos, dando a conocer que el mismo podría trabajar con ese número de procesadores sin problemas dada la información de porcentajes de consumo.

Figura 30

Carga del sistema del servidor 1.

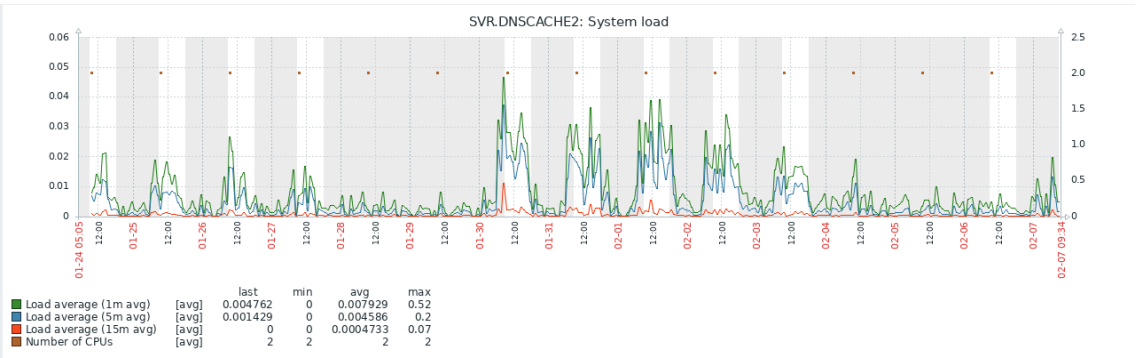


Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Y si el servidor principal DNSCACHE1 hace uso de estos recursos es lógico deducir que el DNSCACHE2 también lo hace, incluso en menor cantidad, al ser un servidor de respaldo como lo muestra la figura 31:

Figura 31

Carga del sistema del servidor 2.



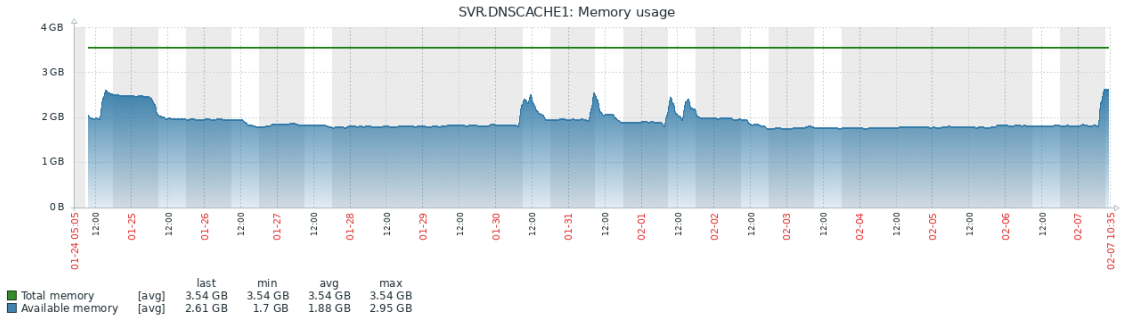
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

3.2.3.6. Memoria RAM.

La máquina virtual está configurada en base a cuatro (4) GB de memoria RAM de las cuales según la Figura 32, la memoria real que usa el sistema es de 3,54 GB.

Figura 32

Uso de memoria del servidor 1.



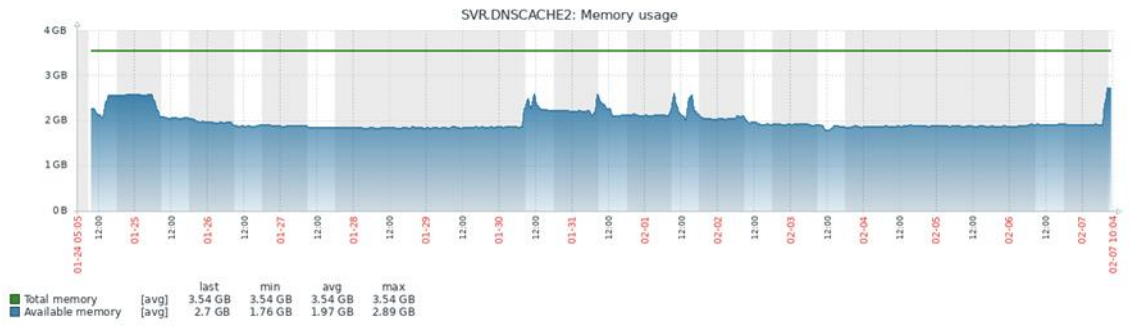
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Como se puede observar lo mismo sucede con la memoria, los picos de uso de este recurso se acentúan en los días laborables dando un consumo de 1,88 GB en promedio y un valor máximo de 2,95 GB por lo que el sistema no ocupa el 100% de la capacidad a la que está configurado.

Lo mismo sucede con el DNSCACHE2 en la figura 33 donde se puede apreciar los mismos parámetros, en este caso la diferencia es mínima o similar. Aunque la configuración para ambos servidores es la misma, en el CPU se nota la diferencia de procesamiento de procesos el uno del otro, pero el uso de memoria no difiere.

Figura 33

Uso de memoria del servidor 2.



Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Los valores de uso de memoria al igual que en el servidor 1 son similares, la memoria total ronda en 3,54 GB donde el promedio es de 1,97 GB en 14 días con un consumo máximo de 2,89 GB.

Para el dimensionamiento de nuestros servidores, esta característica no interfiere en los resultados finales, puesto que la diferencia es mínima en el periodo de tiempo establecido y la comparativa final se basará en un promedio.

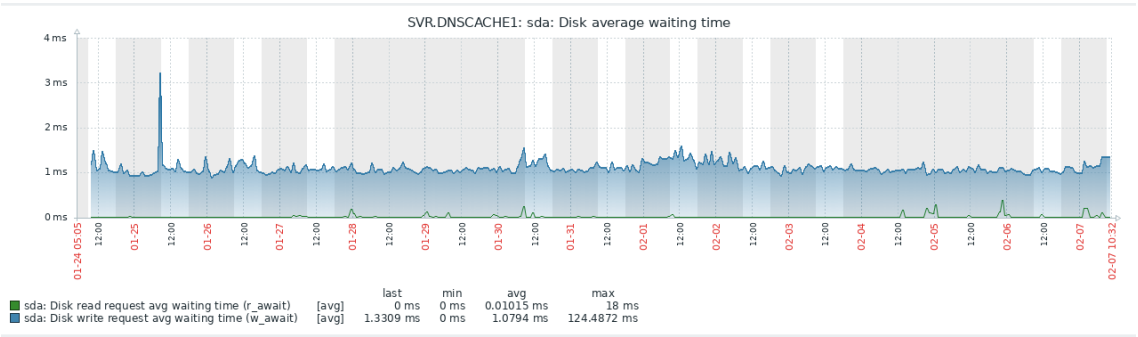
3.2.3.7. Disco Duro

El parámetro de uso de disco duro es el más importante a considerar puesto que nuestros servidores se enfocan principalmente en almacenamiento, no por ser una memoria de caché significa que no se le dé el tratamiento adecuado a la información que se almacena, pues, estos datos refieren el principal hilo de funcionamiento de la DNS y Web caché.

Aunque no es tan relevante, si es importante verificar la información del tiempo de espera promedio de solicitud de lectura y escritura del disco como lo muestra la figura 34:

Figura 34

Tiempo de espera promedio de solicitud de lectura y escritura del disco en el servidor 1.



Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Como se puede observar en la gráfica, el tipo de lectura es mucho menor al de escritura con un promedio de 0,01 ms con picos que pueden llegar hasta los 18 ms y el tiempo de escritura que promedia 1,08 ms con picos de hasta 124,49 ms. La misma herramienta nos permite identificar que en el tiempo de recolección de datos el disco duro usa casi la mitad de su capacidad como lo muestra la figura 35:

Figura 35

Espacio de disco duro utilizado en el servidor 1.

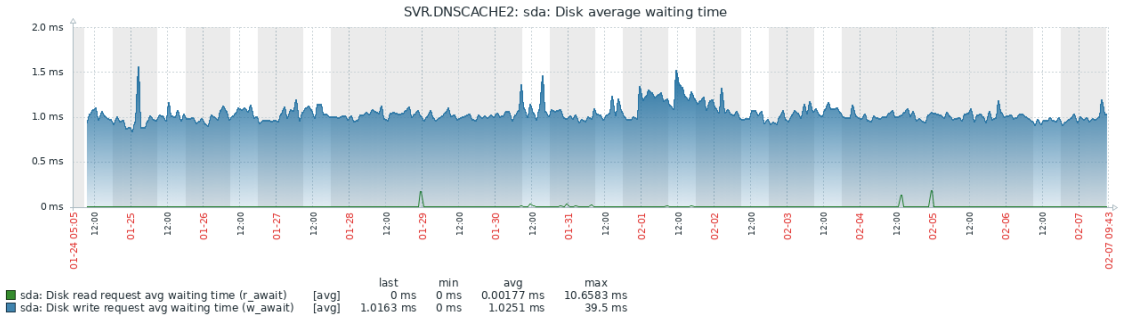
Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Se puede apreciar que el valor de disco disponible en el sistema es de 16,4 GB de los cuales se usa 8,13 GB, es decir, ocupa el 49,56%.

Finalmente, el uso del disco duro para el DNSCACHE2 difiere en los datos, pues es un servidor secundario a lo cual los tiempos de lectura y escritura promedio reducen al tener menos carga como lo muestra la figura 36:

Figura 36

Tiempo de espera promedio de solicitud de lectura y escritura del disco en el servidor 2.



Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Se nota que, en este caso, el servidor 2 reduce el promedio de tiempos de lectura a 0,001 ms con picos que pueden llegar a los 10,65 ms y en el caso de escritura se promedia a 1,03 ms con el pico más alto que se establece en 39,5 ms

De la misma forma, el espacio de uso del disco también se reduce como lo muestra la figura 37:

Figura 37

Espacio de disco duro utilizado en el servidor 2.

Nota. Tomado de Departamento de desarrollo tecnológico e informático de la Universidad Técnica del Norte.

Del total de disco duro disponible que es de 16,4 GB el sistema utiliza 6,55 GB, es decir un 39,93%.

3.2.3.8. Resumen y establecimiento de recursos.

En base a los datos obtenidos se procede a la realización de una tabla comparativa donde se constituye la estimación final del dimensionamiento de recursos para los servidores, en base a datos promediados de los parámetros que se consiguieron en el análisis de las gráficas, con una regla de tres se puede obtener la frecuencia de reloj a la cual trabajaría el sistema en los diferentes casos, recordemos que el sistema puede trabajar hasta los 3,5 GHz.

$$\frac{\%\ Uso\ máximo * 3{,}5\ GHz}{100\%} = Frecuencia\ de\ reloj\ de\ uso\ promedio \qquad [1]$$

Tabla 2

Tabla comparativa de uso promedio de CPU.

	CPU					
	DNSCACHE1			**DNSCACHE2**		
	Utilización de CPU	**Tiempo de uso de CPU**	**Tiempo de sistema de CPU**	**Utilización de CPU**	**Tiempo de uso de CPU**	**Tiempo de sistema de CPU**
Promedio	1,18%	0,49%	0,42%	0,45%	0,18%	0,16%
Uso máximo	35,82%	21,53%	8,37%	29,85%	17,5%	6,51%
Frecuencia	1,25GHz	0,75GHz	0,29GHz	1,04GHz	0,61GHz	0,22GHz

Nota. Adaptada de Dirección de Desarrollo Tecnológico e Informático UTN.

De estos datos podremos promediar las frecuencias de reloj para establecer un parámetro global de uso de CPU en GHz.

Para el caso de memoria se proyecta el mismo proceso como lo muestra la Tabla 3, donde se comparan los datos obtenidos para su promedio final en base a un total de memoria utilizable de 3,54 GB.

Tabla 3

Uso promedio de memoria RAM para los servidores 1 y 2.

	RAM	
	DNSCACHE1	**DNSCACHE2**
Promedio	1,88 GB	1,97 GB
Uso máximo	2,95 GB	2,89 GB
Total de memoria	3,54 GB	3,54 GB

Nota. Adaptada de Dirección de Desarrollo Tecnológico e Informático UTN.

Finalmente se hace la comparativa de datos de uso de Disco Duro para cada servidor.

Tabla 4

Uso promedio de Disco Duro para los servidores 1 y 2.

	Disco Duro	
	DNSCACHE1	**DNSCACHE2**
Total	16,4 GB	16,4 GB
Uso	8,13 GB	6,55 GB
Porcentaje	49,56%	39,93%

Nota. Adaptada de Dirección de Desarrollo Tecnológico e Informático UTN.

Ahora bien, con estos datos podremos promediar cada uno de los parámetros para así establecer el dimensionamiento de los servidores DNS y Web Cache, por tanto, los recursos mínimos de hardware que necesitarían los servidores son:

- CPU: Procesador de 2 núcleos con frecuencia de reloj a 2 GHz.
- Memoria RAM: 4 GB.
- Disco Duro: 15 GB.

3.2.4. Selección de software.

En base a los requerimientos establecidos, se selecciona el software, cumpliendo parte de la metodología de investigación para dimensionar los servicios a implementar, con la realización de pruebas directamente con el software, y una vez realizadas las pruebas se asientan los requerimientos de hardware definitivos para los servidores.

El software que cumple con los requerimientos para el servidor DNS Caché es "Unbound" pues según su origen, no requiere de muchos recursos de Hardware y cumple a cabalidad con cargas grandes de procesamiento. Para el caso de la Web Caché, se usará la herramienta "Apache Traffic Server" pues de la misma forma, cumple con los requerimientos antes establecidos. Este apartado se desarrolla a mayor detalle en el siguiente capítulo mediante la comparativa de características del software con respecto a los requisitos mencionados en el diseño.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS DE FUNCIONAMIENTO

En el presente capítulo se desarrolla el procedimiento a detalle de la instalación y configuración de cada máquina virtual donde se levantan los servicios a implementar, así como también de las herramientas que se usarán para dicho fin, esto, en base a los datos obtenidos en el diseño.

Finalmente, se establecen pruebas y resultados que ayudarán en el análisis para concluir satisfactoriamente con el desarrollo del presente tema de investigación.

4.1. Sistemas Operativos

Si bien es cierto, Linux dispone de una vasta línea de distribuciones que nos permiten instalar servidores basados en software libre, lo primero que viene a nuestra mente al escuchar sistema operativo es “Windows” o “MacOS”, sin embargo, en la actualidad varias son las razones por las que se debería optar por un sistema libre, es decir, aquel que permite su modificación y redistribución mediante la libertad del usuario de hacer lo que sea con el software, he aquí la principal razón, y es que los servidores que se requiere implementar en base a nuestro alcance y justificación deben cumplir la característica de aplicase mediante una “Licencia Libre”, de esta forma las configuraciones se podrán hacer y deshacer sin el problema de violar derechos de terceros, además de que en el apartado económico, el uso de una licencia de este tipo no requiere un gasto pues su distribución es totalmente gratuita.

En base a estas consideraciones se optó por el uso del Sistema Operativo “AlmaLinux” pues, en primer lugar, ofrece características especiales como su distribución gratuita, no requiere tasas ni recargos, así como también el nulo uso de contratos de servicio o restricciones en su registro. Además de que es un SO que está enfocado a un uso empresarial pues es una distribución de Linux muy bien lograda y estable en todo sentido para este fin,

por tanto, se puede usar directamente para la construcción de arquitecturas seguras, operativas y eficientes.

Para la implementación de los servidores que compete al presente tema de tesis, AlmaLinux posee los complementos y soporte necesario para su correcto funcionamiento, esto en base a la documentación de desarrolladores tanto del sistema operativo como de las herramientas administrativas que se van a levantar.

Figura 38

Logo identificativo de AlmaLinux.

Nota. AlmaLinux OS Foundation. (s.f.). Logo AlmaLinux. Recuperado el 15 de febrero, 2023, de https://almalinux.org/

El sistema operativo se instala en una máquina virtual dentro de la configuración del sistema hiperconvergente de Cisco, por lo que se pidió directamente a la parte administrativa del data center ubicado en el departamento de desarrollo tecnológico e informático de la UTN, proceda a la asignación de recursos para las máquinas virtuales necesarias en base al dimensionamiento previsto en el diseño. El sistema operativo para instalar es la versión mínima por lo que no requiere gran cantidad de recursos de hardware. La instalación y configuración se especifica a detalle en el Anexo 1.

4.2. Servidor Web Caché

Los sistemas y aplicaciones informáticas distribuidas confiables se han convertido en la piedra angular de las empresas líderes, especialmente en la automatización y administración de procesos comerciales de misión crítica y en la prestación de servicio al

cliente. Como desarrollador y administrador de sistemas de estos sistemas y aplicaciones, proporciona una variedad de soluciones de tecnología de la información (TI) para asegurarse de tener los sistemas más eficientes disponibles.

Esto incluye tareas tales como diseñar, probar e implementar estrategias de rendimiento, confiabilidad, disponibilidad y escalabilidad del sistema/aplicación para brindar niveles de servicio satisfactorios a los usuarios finales. El almacenamiento en caché es uno de los muchos métodos de entrega de aplicaciones básicos pero efectivos en los que puede confiar. Antes de continuar, echemos un vistazo rápido a lo que es el almacenamiento en caché, dónde y/o cómo se puede usar y sus beneficios.

Este tipo de servicios se pueden implementar de distintas maneras, por ejemplo: el almacenamiento en caché se puede usar del lado del cliente, como en el caché del navegador o en el caché de la aplicación (o en el modo fuera de línea). La mayoría de los navegadores modernos vienen con implementaciones de almacenamiento en caché HTTP.

También puede llevar a cabo el almacenamiento en caché en sus servidores de origen o back-end. Hay diferentes formas de almacenar en caché a nivel de servidor, como un proxy inverso con almacenamiento en caché para imágenes, documentos, scripts, etc o el almacenamiento de bases de datos que proporciona acceso en memoria a datos de uso común, como filas de bases de datos requeridas, resultados de consultas y otros procesos.

El almacenamiento en caché también se puede implementar a nivel de red en una LAN o WAN mediante un servidor proxy. Un ejemplo común de este tipo de caché es una CDN (red de entrega de contenido), que es una red distribuida globalmente de servidores proxy web, que, de hecho, es la implementación que más se acerca a los requerimientos de este tema de investigación.

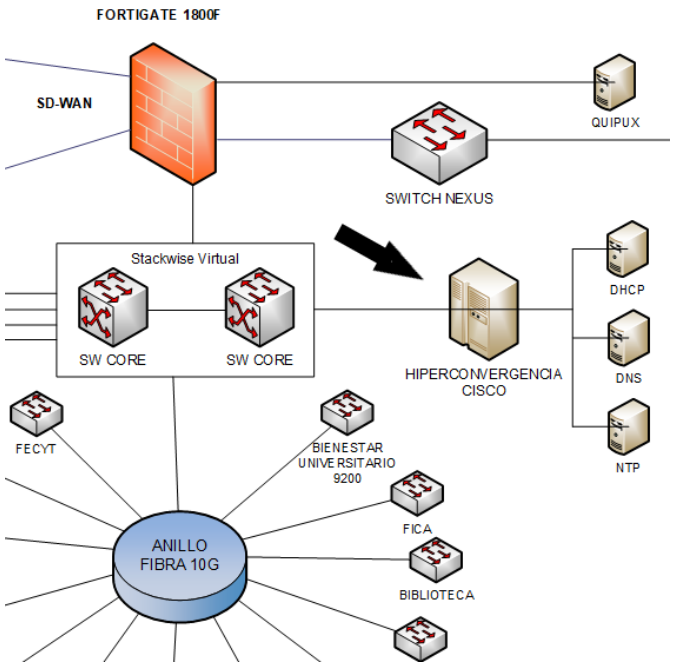
4.2.1. Antecedentes

Para el presente proyecto se analizarán en primer lugar, los antecedentes que nos permitirán establecer la viabilidad en la implementación de un servidor Web Caché o Proxy Caché en la infraestructura actual de la red de la institución.

Como lo muestra la figura 39, la configuración de red se basa en un sistema HCI (Sistema Hiperconvergente) de Cisco, el cual consta de equipos especializados que se especifican en el apartado de Situación actual de red, donde serán levantados los servicios en el presente tema de tesis, además se puede observar que cada uno de los departamentos y facultades se conectan a los switches de core mediante un enlace de Fibra Óptica de 10 Gbps obteniendo salida a Internet mediante la conexión de un Firewall, en este caso un Fortigate 1800F.

Figura 39

Configuración de red de servicios y salida a Internet de la UTN.



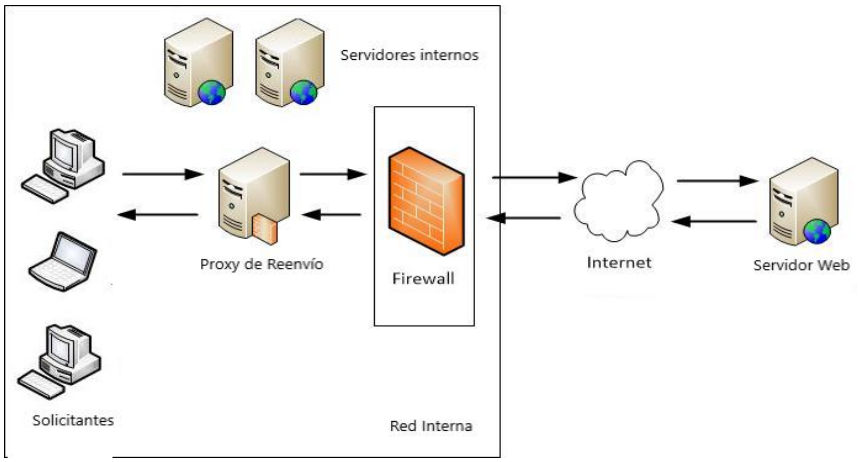
Nota: Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN.

Ahora analizamos la arquitectura de un servidor Proxy Caché o Web caché que se encuentra en la Figura 25, pues como se mencionó anteriormente un servidor de este tipo se puede implementar de distintas formas, pero, en nuestro caso se enfoca directamente hacia mejorar la eficiencia de nuestra red interna por lo que la única configuración de Proxy que se requiere es la de un Servidor de "reenvío" o "transparente", cada configuración será desarrollada a detalle en los siguientes apartados.

Por tanto, si analizamos el funcionamiento de un servidor de Proxy de reenvío como lo muestra la figura 40, este se debe configurar como un intermediario entre la red interna y la internet.

Figura 40

Funcionamiento de un Servidor Proxy de reenvío.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN.

Nuestro servidor debe cumplir este requerimiento de configuración para que funcione correctamente, pero el sistema actual de red no permite su implementación por las siguientes razones:

- Todo el trabajo de filtrado de paquetes se encarga directamente el Firewall Fortigate 1800F.

- La administración de máquinas virtuales para los servicios está a cargo del sistema HCI de Cisco por lo que enrutar una máquina de ellas para filtrado de todos los paquetes de datos que ingresan hacia anillo de Fibra Óptica de 10Gbps sería una carga muy grande que podría provocar cuellos de botella.
- El diseño actual de la red no permite una conexión de este tipo, se requeriría el rediseño de la misma para que sea posible, pero los administradores de la red no lo recomiendan.

Por tanto, la implementación de un Servidor Web caché o Proxy Caché no es viable para la red de infraestructura de la Universidad Técnica del Norte.

Sin embargo, se puede implementar este servidor para fines de pruebas futuras, o bien, establecer una opción de configuración en caso de un rediseño.

4.2.2. Software

En base a una vasta investigación bibliográfica y experiencias documentadas de profesionales en el campo de redes, se puede apreciar que existen múltiples herramientas que permiten configurar un servidor proxy caché con licencias abiertas, pero, una de las más destacadas a nivel empresarial fue “Apache Traffic Server” pues a más de contar con una amplia documentación de configuración posee características especiales que nos servirán para cumplir con los objetivos del tema.

Figura 41

Logo identificativo de Traffic Server.

Nota. Adaptado de Apache (s.f.). Logo Apache Traffic Server. Recuperado el 15 de febrero, 2023, de https://docs.trafficserver.apache.org/index.html

Según sus desarrolladores, Apache Traffic Server es un servidor caché de proxy o caché web de alto rendimiento que ayuda a mejorar la eficiencia y el rendimiento de una red al guardar en caché mucha de la información a la que se accede con frecuencia en los bordes de dicha red. Esto acerca el contenido físico al usuario final, lo que garantiza una entrega más rápida y reduce el consumo de ancho de banda. Principalmente este software está diseñado para mejorar el manejo de contenido para empresas y proveedores de servicio de Internet (ISP) maximizando la eficiencia en el consumo de ancho de banda.

El software presentado soporta cuatro tipos de configuración para un proxy:

- Servidor de Origen.
- Proxy Inverso.
- Proxy de reenvío.
- Proxy transparente.

En este caso, se procede a la configuración del proxy, tanto de reenvío como el transparente, dicho de otra forma, vamos a establecer dos escenarios donde se procede a simular una implementación de este servidor en la red de la institución con estas dos configuraciones, dejando como prueba el funcionamiento del mismo para futuras implementaciones.

4.2.2.1. Instalación y primeros pasos.

La instalación de la herramienta se detalla en el Anexo 2 del presente documento, además se incluye la configuración general e instalación de paquetes adicionales.

4.2.2.2. Archivos de configuración y comandos básicos.

Una vez que el servidor está instalado, se presentan los archivos de configuración principales que se encuentran en el directorio "*/etc/trafficserver/*" los cuales pueden ser configurables mediante herramientas como "vi" o "cat" este apartado se detalla en el Anexo 2 del presente documento, se recomienda la revisión de esta documentación.

Mediante el comando "*systemctl start | restart | stop trafficserver*" se puede iniciar, recargar y parar el servicio de trafficserver respectivamente.

4.2.3. Escenarios de configuración

Se planteó como objetivo la implementación de servidor web caché para reducir costos de ancho de banda y mejorar la eficiencia de la red, por tanto, las configuraciones de Traffic Server que cumplen con este objetivo son las de un proxy de reenvío y transparente, debido a que:

El Servidor de Origen genera el contenido que desea transmitir (y posiblemente almacenar en caché) utilizando Traffic Server. En una configuración de proxy directo, el servidor de origen puede ser cualquier servidor externo al que el cliente proxy intente conectarse. En una configuración de proxy inverso es donde los servidores de origen suelen ser un conjunto conocido de servidores y puede usar Traffic Server como una capa de almacenamiento en caché para mejorar el rendimiento.

Estas opciones no nos sirven por que van dirigidas al lado del servidor y no del cliente, que es lo que se requiere, es por ello, que se presentan los dos escenarios los cuales se van a configurar paso por paso y documentados en las guías de administrador.

4.2.3.1. Caché Proxy Web de reenvío.

Un proxy de reenvío se puede configurar como un instrumento central en una infraestructura para acceder a Internet y se puede usar junto con el almacenamiento en caché

para reducir el consumo general de ancho de banda. Un proxy de reenvío actúa como un guardián entre los navegadores de los clientes en la red y todos (o algunos, según su configuración) los sitios web que visitan esos clientes.

El proxy de reenvío recibirá la solicitud HTTP, aplicará las reglas de modificación o filtrado de solicitudes que haya configurado y reenviará la solicitud a su sitio de destino en consecuencia. La respuesta se devolverá a través del mismo proxy, posiblemente almacenada en caché y/o modificada y devuelta al cliente original.

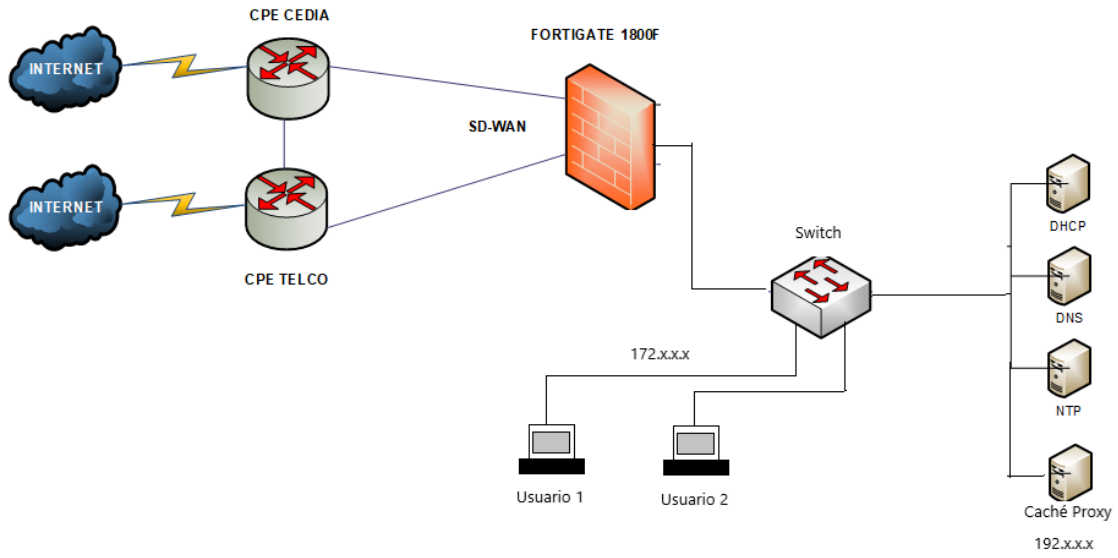
Para este servicio cada cliente debe configurarse para usar un proxy de reenvío. Los navegadores de los clientes comprenderán que están utilizando un servidor proxy y configurarán sus solicitudes HTTP en consecuencia. Esto da como resultado que el comando HTTP original se emita con un URI completo que contiene el nombre de host de destino, es decir, este proceso es visible para el usuario en cuanto a la configuración se refiere, se pierde el sentido de invisible para el usuario, pero tiene sus ventajas, como la rapidez en la implementación.

- *Topología*

Como se mencionó al inicio del apartado, esta implementación no es viable para una aplicación directa en la red de la institución, pero se adaptó la simulación de los posibles escenarios en caso de un rediseño, primeramente, se plantea una red de forma local que simula clientes con peticiones hacia Internet mediante un puerto LAN que fue suministrado por la universidad de manera que este puerto es la salida de la WAN, pero son interceptados por nuestro servidor Proxy Caché como lo muestra la figura 42:

Figura 42

Topología Escenario 1.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN.

Como se puede observar, para este escenario la configuración parte de la máquina virtual montada dentro del sistema Hiperconvergente, que conecta directamente dos clientes hipotéticos mediante un switch a los cuales se les ha asignado un direccionamiento clase B: 172.x.x.x.

- *Configuración del escenario*

Para este escenario se va a implementar un servidor proxy de reenvío es por ello que los clientes se tendrán que configurar tanto con la dirección IP del servidor como también del puerto en el que trabaja (192.x.x.x:xxxx) para que el reenvío se realice directamente hacia dicho servidor, esta configuración se detalla en el Anexo 3.

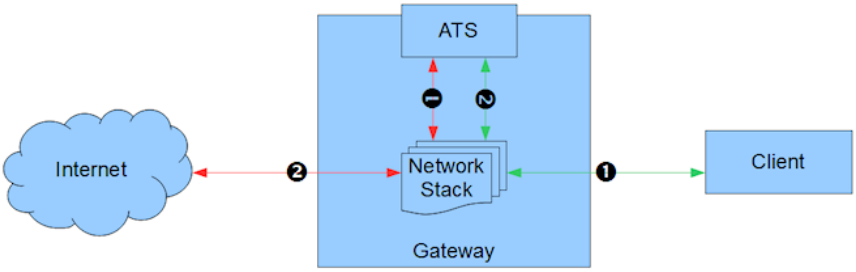
4.2.3.2. Caché Proxy Web transparente

Esta configuración es la que más se acerca al objetivo principal de la implementación de un servidor web caché, puesto que el proxy transparente tiene la capacidad de interceptar conexiones entre cliente-servidor sin ser visible, es decir, para el usuario esto es totalmente

invisible el proceso de captura y almacenamiento de caché, esto lo podemos comprender de mejor manera analizando la figura 43:

Figura 43

Flujo de tráfico básico Traffic Server como proxy transparente.



Nota. Adaptado de Adaptado de Apache (s.f.). Logo Apache Traffic Server. Recuperado el 15 de febrero, 2023, de https://docs.trafficserver.apache.org/admin-guide/configuration/transparent-proxy.en.html

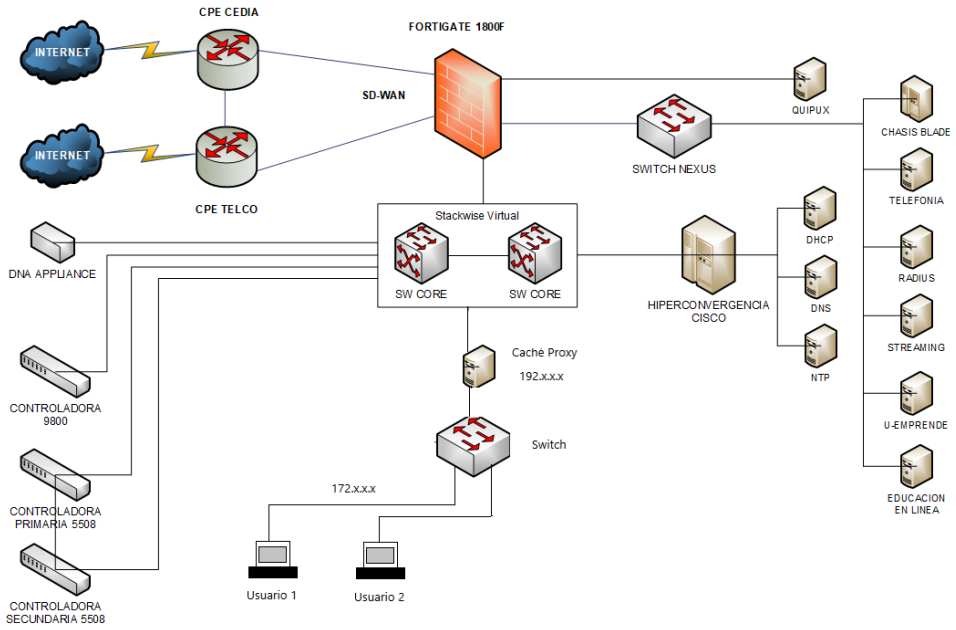
Existe una puerta de enlace por la cual todo el tráfico pasa del cliente hacia Internet, esta puerta es la encargada de establecer un empalme entre Traffic Server y los flujos seleccionados de ese tráfico, la configuración de parte del cliente es nula, lo único que se realiza es un enrutamiento de su tráfico hacia Traffic Server de forma interna de tal forma que trabaje como un interceptor, aquí realiza el filtrado correspondiente y al mismo tiempo el almacenado en la caché al igual que un proxy de reenvío.

- *Topología*

Para este escenario se establece al Traffic Server como un enrutador Linux, es decir, tenemos dos interfaces, una de entrada y otra de salida, los clientes se ubican en la red 172.x.x.x/26 y el enrutador en la red asignada por la administración 192.x.x.x, entonces la configuración de red quedaría:

Figura 44

Topología Escenario 2.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN.

- *Configuración del escenario*

Para esta configuración, se hace uso de la herramienta "*iptables*" muy bien conocida por los profesionales en el ámbito de las redes, pues nos permite, mediante reglas, establecer un enrutamiento de los paquetes de datos a través de un servidor Linux, esta configuración se podría llamar una especie de firewall, pero configurable mediante líneas de comando y reglas de aceptación o denegación. Este proceso se detalla en el Anexo 3.

4.2.4. Pruebas y Resultados

Las pruebas para ambos escenarios son exactamente iguales, puesto que los resultados no varían en ninguna configuración, el almacenamiento en caché es igual compartiendo los mismos parámetros, lo único que varía es la forma como se obtiene la información.

4.2.4.1. Pruebas desde cliente

Para estas pruebas haremos uso del monitor de red de nuestro navegador, donde realizamos una petición hacia la página web “https://utn.edu.ec”

En primer lugar, nos dirigimos hacia la barra de navegación del servidor e ingresamos al enlace que nos dirigirá hacia la página de la Universidad Técnica del Norte:

Figura 45

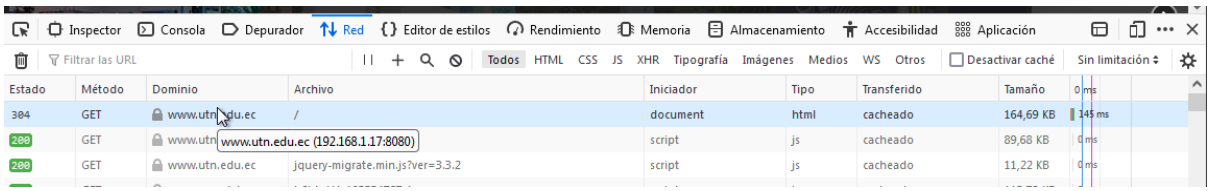
Ingreso a la página de la Universidad técnica del Norte.

Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Con la tecla F12 accedemos al monitor del navegador, el cual nos brindará toda la información acerca de la petición de la página, en el apartado de red podremos ver dicha información, empezando de los métodos y los archivos que se recuperan, sean estos directamente del servidor de origen como del servidor de caché, desmarcamos la opción de “Desactivar caché” para comprobar que la información que se obtiene no venga del caché del navegador, sino que venga directamente del servidor de caché.

Figura 46

Objeto HTML recuperado del servidor de caché 192.x.x.x:8080.



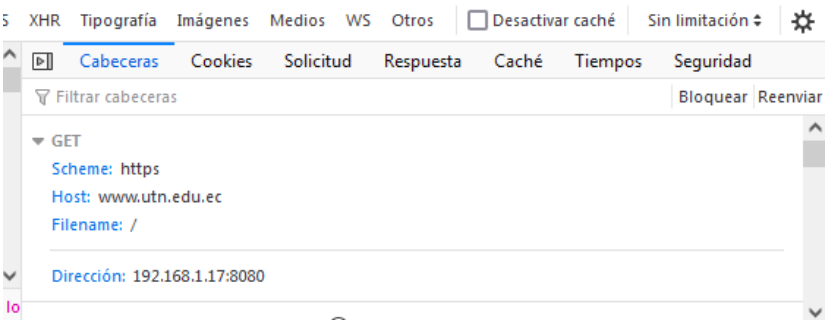
Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Esta imagen nos brinda muchísima información para comprobar el resultado de la implementación del servidor caché proxy web, como se puede observar, objetos como documentos HTML han sido cacheados, y si nos enfocamos en el apartado de dominio, vemos que viene anexada la información del servidor de caché, por lo que se asume que estos objetos han sido cacheados por nuestro servidor.

De esta forma con la información de la cabecera también se puede observar su origen, que ha sido recibida de la dirección IP del servidor de caché como lo muestra la figura 47:

Figura 47

Información de cabecera.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

4.2.4.2. Pruebas desde servidor

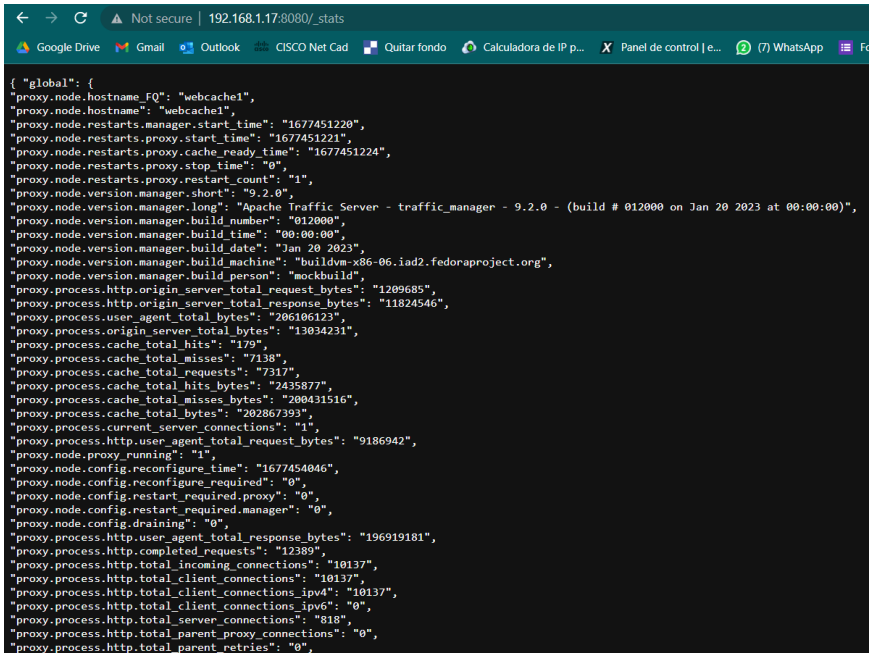
Desde el lado del servidor también es posible su monitoreo, puesto que el software incluye herramientas que nos permiten verificar el funcionamiento del servicio mediante algunas métricas de medición estadísticas.

Traffic Server incluye un muy completo complemento de estadísticas sobre HTTP que proporciona acceso HTTP a todas las estadísticas. El complemento devuelve un objeto JSON que contiene las estadísticas y sus valores actuales. No se puede devolver un subconjunto de estadísticas. El complemento debe estar activado para usarlo.

Para habilitar este complemento se debe agregar el comando "*stats_over_http.so*" en el archivo "*plugin.config*". y para acceder a esta función nos dirigiremos a un navegador local e ingresaremos lo siguiente "*http://192.x.x.x:8080/_stats*" donde ingresamos la dirección IP de nuestro servidor junto con el puerto de funcionamiento:

Figura 48

Acceso al archivo JSON de estadísticas de tráfico del servidor.



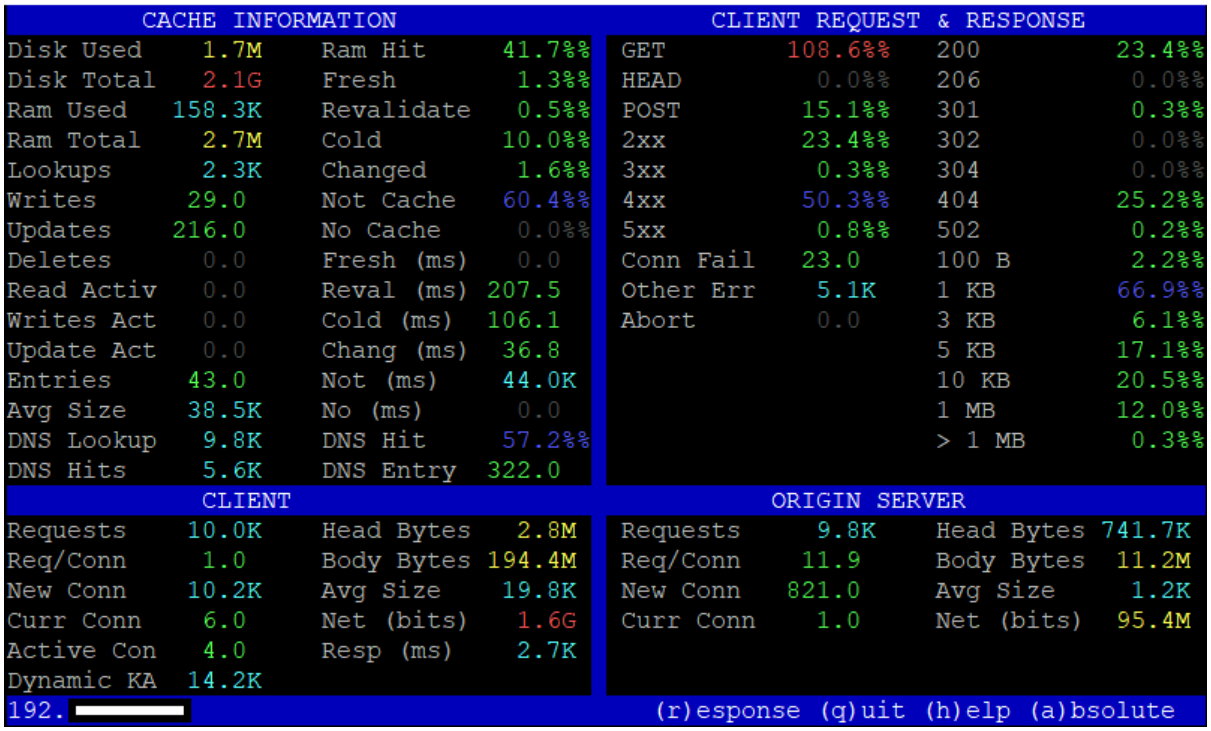
Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Pero esta información es complicada de analizar, por lo que la misma herramienta nos permite también el monitoreo de este archivo de forma más amigable a la vista, mediante una interfaz gráfica, para ello hay que activar la función, con el uso de "*traffic_top*".

Este programa solo requiere la modificación del archivo "*plugin.config*" añadiendo la siguiente línea "*stats_over_http.so statistics*" con esto solamente reiniciamos el servicio y mediante el comando "*traffic_top http://192.x.x.x:8080/statistics*" ingresamos a la interfaz:

Figura 49

Interfaz gráfica de traffic_top para el monitoreo.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

La información que se presenta es mucho más fácil de analizar, cada métrica identifica un parámetro en específico, aquí solo analizaremos las estadísticas más relevantes que competen a comprobar los resultados, en el apartado de memoria caché se puede rescatar:

- Disk Used: es el espacio que ocupa en disco actualmente por la memoria caché
- Disk Total: es la capacidad de disco que se asignó para memoria en caché
- Ram Used: es la cantidad de ram de caché usada por los objetos
- Lookups: es la cantidad de búsquedas de caché realizadas
- Writes: es la cantidad de escrituras de objetos hechas en la memoria de caché
- Upgrades: es el número de objetos de caché que existen y que se actualizaron con contenido nuevo del servidor de origen
- Deletes: es el número de objetos eliminados de traffic server.

También se puede apreciar en el apartado de cliente, la información que ha adquirido en el transcurso de su navegación, como el número de peticiones, con el contenido nuevo o el contenido concurrente.

En fin, esta herramienta nos permite un monitoreo en vivo de los procesos que se llevan a cabo, dentro de la memoria de caché, de parte de clientes y de los servidores de origen.

Para los fines de prueba se puede apreciar que los objetos en memoria de caché apenas superan el 1Mb de información, con la asignación de 2Gb es más que suficiente y la memoria RAM tampoco hace mucho uso, pero en ambientes con muchos clientes estos parámetros si deben cumplir un mínimo de capacidad mucho mayor al que está configurado actualmente.

4.3. Servidor DNS Caché

La caché de DNS es necesaria para responder más rápido a las consultas de DNS para los nombres de dominio que hemos visitado recientemente.

Tanto el dispositivo que usa el usuario como los múltiples solucionadores de DNS a los que llegan las solicitudes tienen un caché de DNS y pueden resolver dominios que todavía están en el caché. De lo contrario, una consulta DNS tendría que recorrer un largo camino hasta el servidor raíz, que iría al servidor TLD, luego al servidor de nombres autorizado para el nombre de dominio y finalmente recibiría una respuesta.

Es por ello que en base al diseño se procede con la implementación de un servidor DNS de caché en este caso los antecedentes ya vienen dados por la situación actual de la red donde se maneja un DNS con directorio activo en Windows server, pero el objetivo del presente proyecto es el de mejorar la eficiencia en la red, un servidor DNS con

almacenamiento en caché ayudaría enormemente a su cumplimiento, además de que sea aplicado mediante herramientas de software libre.

4.3.1. Software

En base a la investigación previa de diseño se tomó en cuenta varias herramientas que cumplen con los requisitos para el cumplimiento del objetivo de implementación de un servidor DNS caché, pero la herramienta a nivel corporativo con mejores prestaciones es Unbound.

Unbound se ejecuta en varias plataformas e incluye los repositorios necesarios para su correcto funcionamiento, es un sistema con validación y almacenamiento en caché de resolutores de DNS. Está diseñado para ser rápido y eficiente, combinado con características modernas basadas en estándares abiertos.

Figura 50

Logo identificativo de Unbound.

Nota. Adaptado de Ubound. (s.f.). Logo identificativo de Unbound. Recuperado el 15 de febrero, 2023, de https://unbound.docs.nlnetlabs.nl/en/latest/

Además, esta herramienta permite el control de contenidos, que bloquea el ingreso hacia dominios maliciosos, lo cual también se va a configurar como medidas de seguridad.

4.3.1.1. Instalación y primeros pasos

Para la Instalación del software se realizó una guía que se detalla en el Anexo 4. Aquí se especifica paso a paso la instalación y configuración básica del software.

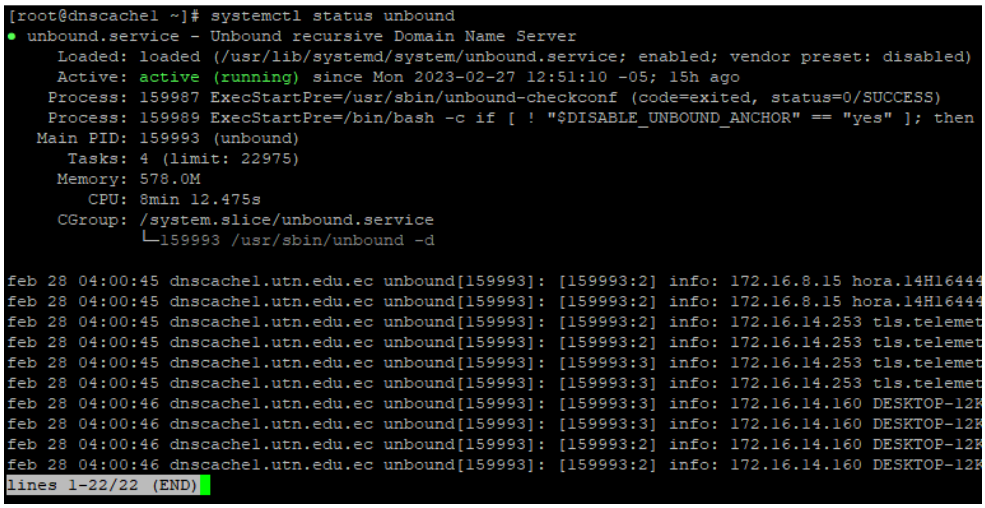
4.3.2. Pruebas y Resultados

Las pruebas se realizan tanto desde el servidor como del usuario, por ello este apartado se enfoca en los resultados obtenidos de ambas partes, y así verificar la convivencia de ambos en el entorno.

Para la realización de pruebas primero comprobamos el funcionamiento del servicio, en el arranque del mismo desde el servidor con el comando "*systemctl status unbound*"

Figura 51

Comprobación del arranque del servicio de Unbound.

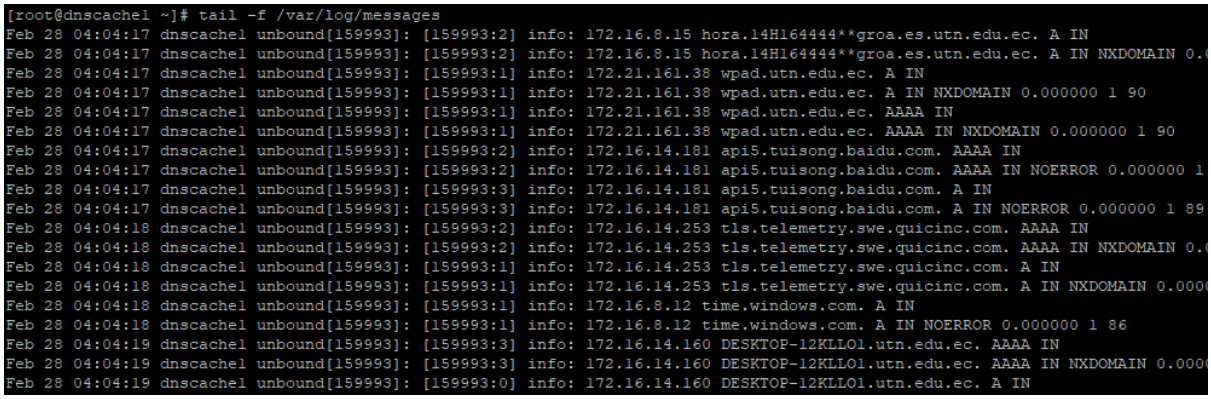


Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Una vez comprobado que el servicio está en funcionamiento, comprobamos el estado de "logs" de esta forma podremos identificar los procesos que el servidor lleva a cabo, esto de forma lineal:

Figura 52

Estado de Logs del servidor.

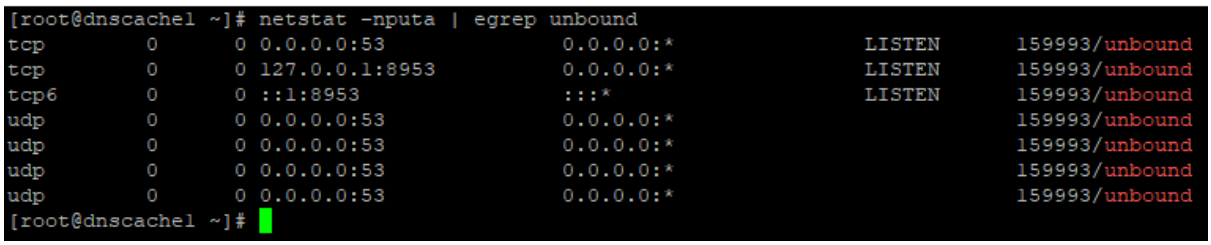


Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

También procedemos a verificar el servicio mediante el comando "*netstat -nputa | egrep unbound*" para verificar información importante:

Figura 53

Verificación del estado del servicio.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Como se puede observar, unbound hace uso de las interfaces que se han configurado, en este caso de todas las disponibles pues la configuración fue en la interfaz 0.0.0.0 así también, en el puerto 53 para TCP y UDP.

Desde el servidor podremos una consulta mediante la dirección IP del servicio así:

Figura 54

Consulta desde servidor.

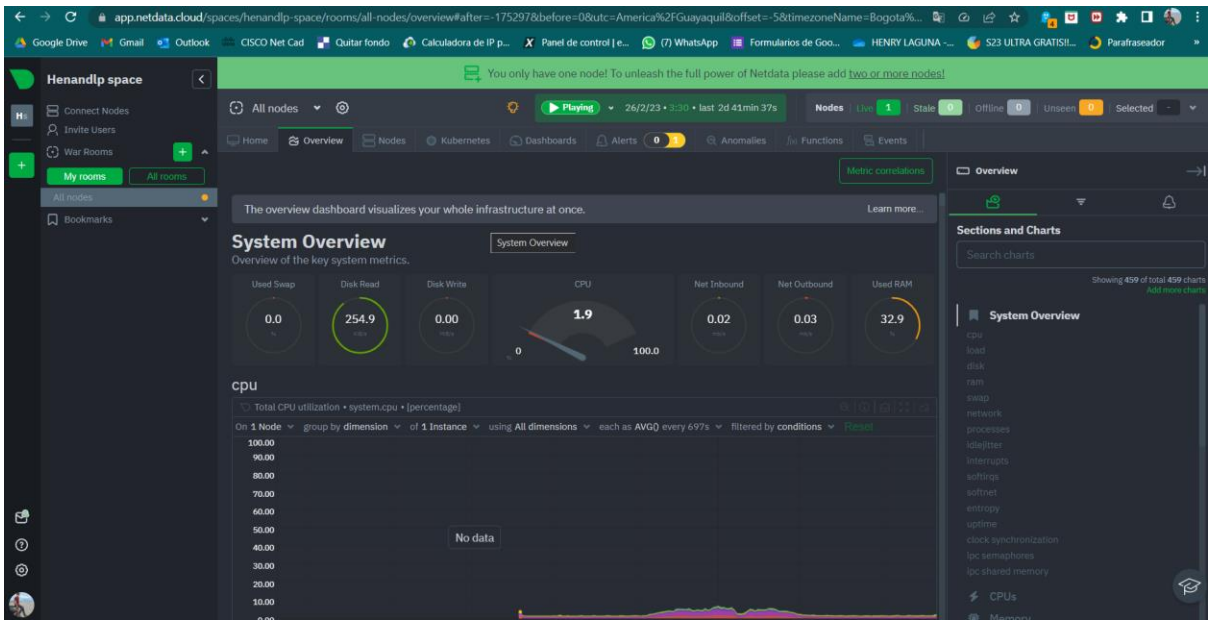
```
[root@dnscachel ~]# host -t a www.utn.edu.ec 1[redacted]
Using domain server:
Name: 1[redacted]
Address: 1[redacted] #53
Aliases:

www.utn.edu.ec has address 190.15.133.224
[root@dnscachel ~]#
```

Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Para el monitoreo del servicio se ancló el servidor hacia la herramienta Netdata, la cual nos permite analizar mediante gráficas una gran cantidad de métricas, aquí analizaremos las más relevantes para especificar el correcto funcionamiento de Unbound, entonces anclamos el nodo de la institución mediante la dirección IP del servidor:

Figura 55

Anclaje de Unbound hacia Netdata.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Las Pruebas se basaron específicamente en establecer datos estadísticos sobre el funcionamiento del servidor en un lapso de tiempo determinado, directamente en días laborables, lógicamente siguiendo la línea de razón en la que se puede mencionar que en este

tiempo es donde más carga útil se hace a los servicios de la institución, refiriéndonos de forma general, no obstante, los servicios de red entrarían a formar parte también de esta afirmación.

La herramienta de monitoreo, Netdata, ofrece una amplia gama de gráficas estadísticas que nos permiten analizar un sinfín de métricas de la máquina virtual a la que llamaremos nodo, sin embargo, únicamente haremos uso de los datos del servicio en sí, en este caso, de Unbound, el cual, también forma parte del conjunto de gráficas.

En este caso, vamos a analizar todas las estadísticas de tráfico que forman parte de Unbound, y de esta forma comprobar su correcto funcionamiento, por ejemplo observemos la figura 56:

Figura 56

Gráfica de peticiones recibidas de Unbound.

Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Como se puede analizar la gráfica se aprecia las fechas en las cuales se realizó la captura de esta información, en un lapso de 5 días, a partir del día 1, 2, 3, 4 y 5 de marzo, en base al calendario actual son los días miércoles, jueves, viernes, sábado y domingo, claramente se puede observar que los días con mayor actividad son los correspondientes a los laborables, mientras que el fin de semana es baja o casi nula, de la misma forma en el gráfico podremos identificar que la actividad se mide en base a las peticiones que se realizaron hacia el servidor con tope máximo de 600 peticiones por segundo, y promediando las 400, de hecho, incluso se puede apreciar que a la hora de almuerzo de los servidores de la institución y estudiantes la actividad del servicio promedia las 150 peticiones por segundo, es lógico analizar que a menor número de usuarios también es menor la actividad del servidor.

Ahora bien, se comprobó que los usuarios realizan peticiones que son procesadas por el servidor de Unbound, sin embargo, también hay la posibilidad de analizar los tipos de peticiones, hablamos de un servidor de DNS con almacenamiento en caché, por tanto, específicamente son los tipos de registros los que se analizan en la gráfica de la figura 57:

Figura 57

Peticiones de los usuarios por el tipo de registro.

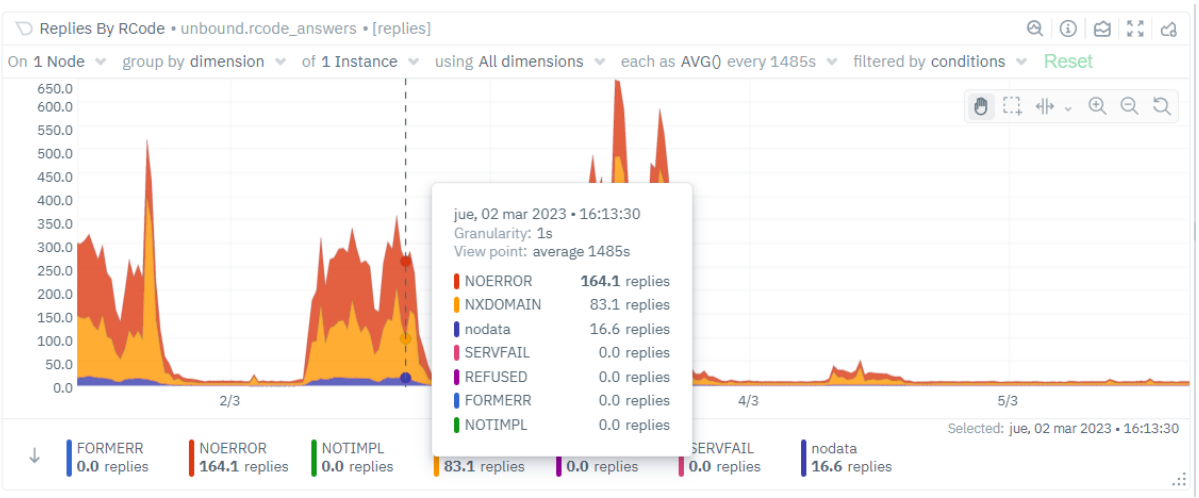
Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Se puede apreciar que la gráfica nos ofrece las estadísticas del número de peticiones por registro, es decir en este caso, en un punto de la línea del tiempo del día 2 de marzo, exactamente a las 16:38 horas ser realizaron 136 peticiones del registro de tipo A, de AAAA 113 peticiones, siendo estos los que más procesa el servidor, además se tiene un registro de peticiones HTTPS, que vienen siendo solo de consulta.

Ahora, conforme se realizaron peticiones también se realizan sus respectivas respuestas, estas se pueden observar en la figura 58:

Figura 58

Respuestas del servidor por tipo de código.



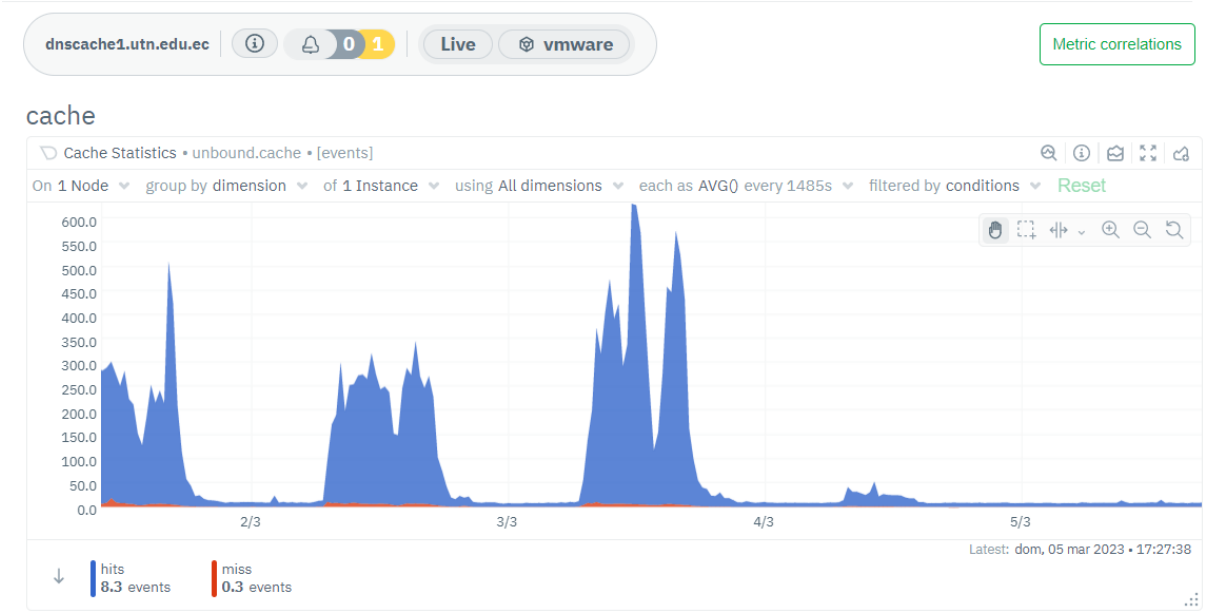
Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Las respuestas para cada petición vienen dadas por un código que identifica si la resolución del dominio se llevó a cabo o no, NXDOMAIN representa un dominio inexistente y mensajes de error de DNS recibidos por servidores DNS recursivos (clientes) cuando el dominio solicitado no se puede resolver en una dirección IP, pero como se observa los NOERROR son respuestas que superan la estadística al anterior código, esto se debe a que

Unbound realiza una consulta en su caché para resolver el dominio, si no obtiene la información que requiere simplemente se registra un código de NXDOMAIN, para resolverlo de forma externa, ahora bien, las gráficas nos permiten establecer las estadísticas del almacenamiento en caché, de esta forma comprobamos que Unbound si almacena información temporalmente mirando la siguiente figura:

Figura 59

Información de almacenamiento en la memoria caché de Unbound.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Primeramente, se puede observar que las gráficas se parecen a las de la figura 58, básicamente comparten similares números estadísticos en la misma línea de tiempo, esto se debe lógicamente porque las repuestas de las peticiones de cada usuario se establecen por el almacenamiento en caché del servicio, de lo contrario no sería un servidor DNS caché.

En base al cálculo de la siguiente ecuación 2, podremos verificar la proporción de aciertos en caché, esto nos permitirá medir cuántas solicitudes de contenido puede manejar el caché en comparación con la cantidad de solicitudes recibidas.

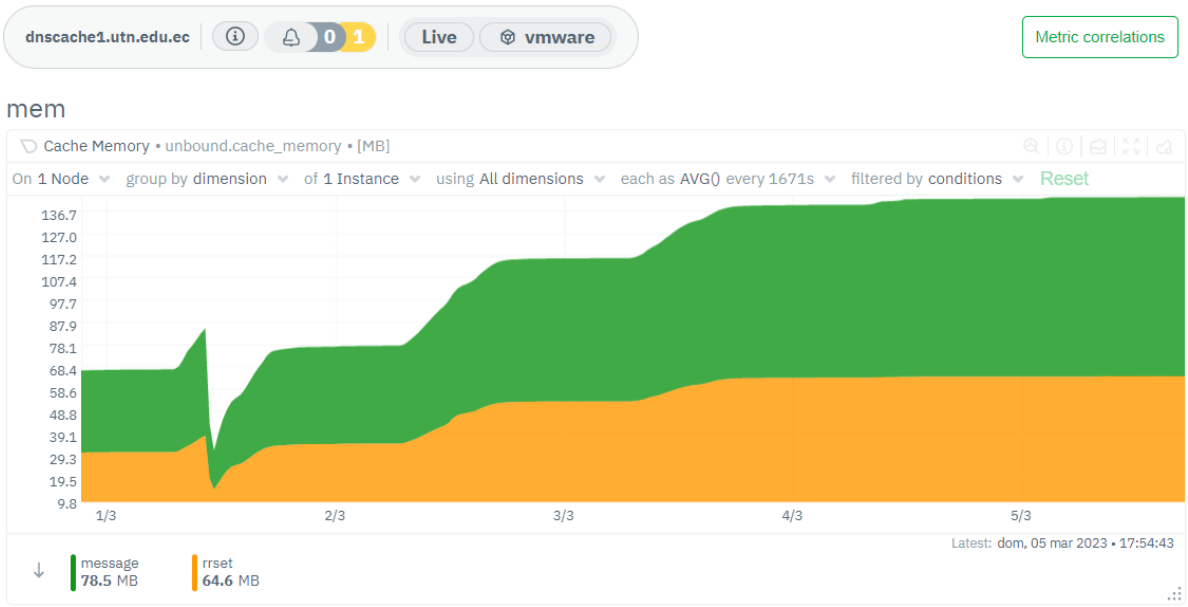
$$\frac{número\ de\ caché\ hits}{número\ de\ caché\ hits + número\ de\ caché\ miss} = proporión\ de\ aciertos\ de\ cache \quad [2]$$

Por ejemplo, tomamos en cuenta un dato promedial en base a las gráficas de 350 caché hits y 7 caché miss, aplicamos la fórmula, podremos calcular que existe 0.98 de proporción de aciertos de caché, en expresión porcentual, multiplicamos este valor por 100, dándonos un 98% de acierto, lo cual demuestra que el sistema funciona en óptimas condiciones.

Finalmente vamos a comprobar el estado de la memoria, entendiendo que un DNS como Unbound almacena información de caché directamente en la memoria RAM, debido a que son datos temporales, para ello analizamos la figura 60:

Figura 60

Estado de memoria de Unbound.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

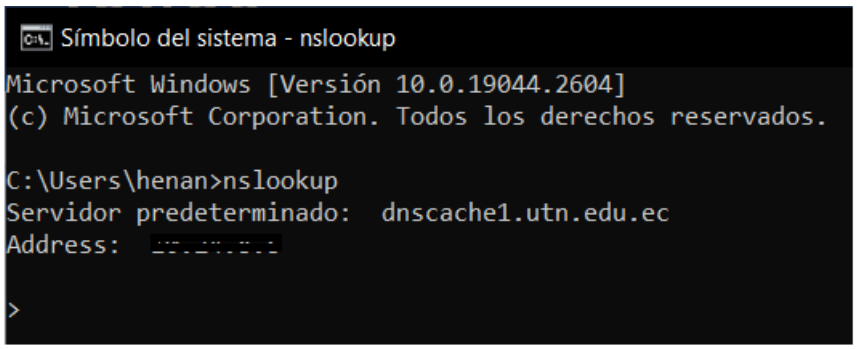
Esta gráfica nos proporciona mucha información, en primer lugar, podemos observar que los datos de registros se almacenan en forma de mensajes y de tipo rrset que no son más que conjuntos de registros del mismo tipo que forman parte del protocolo DNSSEC.

Se puede apreciar que el uso de la memoria se ve claramente afectado por la actividad del servidor, a partir del día viernes 4 de marzo, no se actualiza la información de memoria debido al bajo uso del servicio de los siguientes días por ser un fin de semana, por tanto, esa memoria permanece activa hasta que el usuario la requiera y el mismo sistema actualice la información de los registros almacenados en caché.

Del lado del cliente se realiza una prueba para comprobar el tiempo de respuesta de una consulta realizada mediante el servidor DNS caché, observemos la figura 61:

Figura 61

Estado de servidor DNS en un usuario.

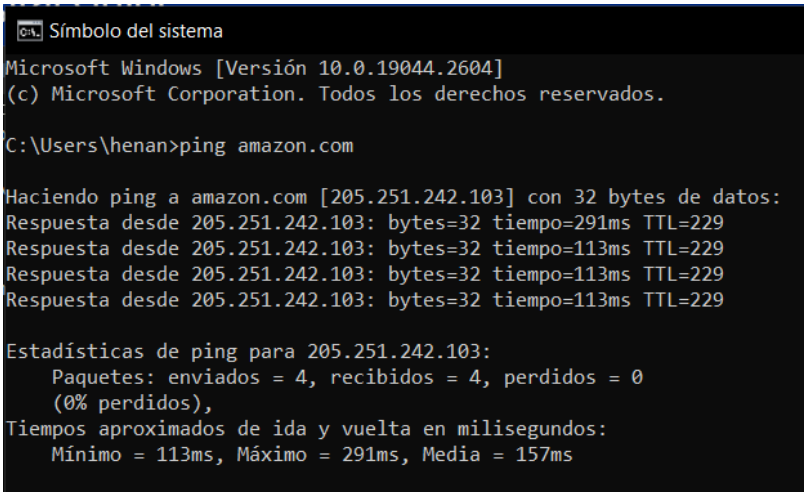


Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

En un usuario que se encuentra en uso de la red de la institución, se puede comprobar mediante un "*nslookup*" el estado de su servidor DNS, en este caso directamente se puede apreciar que se encuentra dentro del dominio que corresponde al servidor de DNS principal. Pudiendo resolver los dominios configurados en la zona, de la misma forma mediante un ping, fácilmente se puede apreciar la disminución del tiempo de estará de la resolución del dominio al que se quiere ingresar:

Figura 62

Ping hacia página web para comprobar la resolución de su dominio.



Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Claramente se aprecia, que en la primera respuesta se estima un tiempo de 291ms, mientras que en las siguientes ese número reduce a 113ms, aprovechando 178ms de tiempo de respuesta, porcentualmente existe un 61% de mejora.

CAPÍTULO V. ANÁLISIS COSTO-BENEFICIO

En este capítulo se realiza un análisis de beneficio en base los resultados obtenidos en la implementación de los servidores de DNS y Web caché, de esta forma se podría relacionar el estado de la red de la institución en una línea de tiempo, identificando un antes y un después frente a la implementación de los servicios antes mencionados, y así, analizar si existe o no beneficio alguno, sus repercusiones y finalmente la viabilidad del caso.

5.1. Viabilidad económica

Los costos de viabilidad económica se consideraron en base al proceso de diseño, implementación y configuración de los servidores para entrar en producción, mas no el de mano de obra, debido a que la institución ya posee el personal que se encarga de los procesos de mantenimiento y administración de la red, es decir, es un gasto que no se considera parte del análisis.

Debido a que la infraestructura de red de la institución ya consta del hardware necesario para la implementación de nuevos servicios, no fue necesario incluir esos gastos para el presente análisis, pues existió su disponibilidad al 100% mucho antes de iniciar con la investigación del caso.

Es por ello que, en el presente caso de estudio, los beneficios económicos no influyen directamente en los costos de la infraestructura de red, pues no se generaron gastos de implementación y configuración de los servidores.

Entendiendo esto, los únicos gastos que se podrían considerar parte del proyecto son de software, sin embargo, en las condiciones de diseño se especificó que las herramientas deben ser instaladas estrictamente en base a licencias de software libre, es por ello que, como se muestra en la tabla 5, los gastos en las herramientas para la creación de las máquinas virtuales, servidores y cuentas de monitoreo no generaron ningún gasto económico.

Tabla 5

Lista de costes de software.

Cantidad	Descripción	Costo
1	Licencia Instalador AlmaLinux	$0
1	Licencia “Apache Traffic Server”	$0
1	Licencia “Unbound”	$0
1	Cuenta “Netdata”	$0

Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Este ahorro resulta ser significativo, sin embargo, los costes de utilización de salida a internet por los proveedores externos de la institución siguen siendo los mismos, el beneficio se refleja en otros ámbitos.

5.2. Beneficios

Los beneficios del servidor de DNS caché que entra en producción se pueden clasificar en dos casos, aquellos que favorecen a la infraestructura de red y los que van conforme al usuario.

En el ámbito de red, el servicio actual de DNS con Unbound representaría mejoras significativas en comparación al anterior de Windows Server 2008, pues, en un contexto general este servicio con directorio activo tiene ciertas desventajas como:

- Active Directory es una solución únicamente para Windows, si se requiere administrar máquinas con Linux o Mac se deben implementar mediante clientes LDAP.
- Dependiendo de los sistemas que administre y su volumen, un Active Directory puede resultar en valores económicos costosos, algo que resulta no tan viable económicamente para una organización, pudiendo optar por servicios de DNS con costes nulos.

- El rendimiento puede verse afectado en redes grandes.
- Active Directory no permite la capacidad de monitoreo a los administradores de TI, algo que dificulta enormemente determinar el uso del sistema por parte de los clientes incluso el tráfico que se genera.
- El servicio actual de Active Directory se maneja mediante un Windows Server 2008, el cual, según la página web oficial de Microsoft para los siguientes productos, el soporte ya habría caducado desde el año 2020 como lo muestra la tabla 6.

Tabla 6

Datos de finalización de soporte para productos de Windows Server.

Producto Windows Server 2008	Fecha de fin del soporte
WS 2008 Datacenter	1/14/2020
WS 2008 Datacenter sin Hyper-V	1/14/2020
WS 2008 Enterprise	1/14/2020
WS 2008 Enterprise sin Hyper-V	1/14/2020
WS 2008 para sistemas basados en Itanium	1/14/2020
WS 2008 Foundation	1/14/2020
WS 2008 R2 Datacenter	1/14/2020
WS 2008 R2 Enterprise	1/14/2020
WS 2008 R2 para sistemas basados en Itanium	1/14/2020
WS 2008 R2 Standard	1/14/2020
WS 2008 Standard	1/14/2020
WS 2008 Standard sin Hyper-V	1/14/2020

Nota. Adaptado de Microsoft. (s.f.). Finalización de la compatibilidad con Windows Server 2008 y Windows Server 2008 R2. Recuperado el 3 de marzo del 2023 de https://learn.microsoft.com/es-es/troubleshoot/windows-server/windows-server-eos-faq/end-of-support-windows-server-2008-2008r2

A pesar de que este tipo de implementación resulta ser más segura por su confiabilidad y escalabilidad con una administración más centralizada, es justificado el cambio del sistema debido a las desventajas antes mencionadas.

Es entonces que con el sistema actual de DNS se presentan ciertas ventajas y beneficios como:

- Al ser un servidor de DNS con almacenamiento en caché, prácticamente desaparece la carga excesiva de la red, eliminando cuellos de botella, aglomeraciones y fallos de rendimiento.
- Por la misma razón los tiempos de respuesta para las consultas desde el usuario reducen drásticamente como se pudo evidenciar en el apartado de resultados para Unbound.
- Al mantener un licenciamiento de software libre, su uso, mantenimiento, configuración y administración es total, es decir, sin restricciones.
- No genera gastos económicos.
- Al ser un sistema implementado en Linux, su monitoreo es totalmente abierto y nos brinda información exacta de cualquier métrica en su uso en tiempo real, esto facilita al administrador en el control del sistema, corrección de errores, verificación de los usuarios y carga de tráfico.
- Permite la creación de zonas de DNS en el mismo sistema de caché.

Se analiza la figura 63 para comprobar la carga de tráfico del sistema como parte de las ventajas antes mencionadas.

Figura 63

Tráfico en las interfaces del servidor Unbound.

Nota. Adaptado de Dirección de Desarrollo Tecnológico e Informático UTN

Como se puede observar dentro de las estadísticas demuestran que el servidor consume un máximo de 1Mbps promediando a los 0,4Mbps por lo que en sí el servicio no representa una carga demasiado grande para la red, y otra ventaja es que las consultas de la zona de DNS se resuelven de forma interna, aparte de que se obtiene un 98% de aciertos de caché, es decir, el consumo de ancho de banda es mínimo, el beneficio se ve enfocado directamente al usuario en reducir sus tiempos de respuesta en la resolución de dominios como se pudo evidenciar en las pruebas del servidor Unbound.

En consecuencia, todas estas ventajas generan beneficios para el usuario, a pesar de que estos procesos se llevan a cabo de forma transparente para ellos, por ejemplo:

- Los tiempos de respuesta de una consulta se reducen en el proceso de resolución del dominio del sitio web al que quiere acceder, permitiendo al usuario mantener una navegación mucho más rápida y fluida, generando mayor conformidad en el uso de la red de la institución.

- El tener menos carga en el tráfico de la red, significa que el sistema se puede aprovechar para conectar más usuarios.
- Si el servicio mejora la eficiencia en un promedio del 60%, se puede aprovechar ese porcentaje de beneficio para permitir incluir incluso más servicios en la red que mejoren aún más la experiencia de usuario.

La herramienta de Unbound, al mantenerse en un licenciamiento de tipo libre, su soporte únicamente está limitado al sistema operativo al que se instala, sin embargo, cualquier versión Linux de la que el software es compatible, ofrece soporte para un largo tiempo al futuro.

Según los desarrolladores de la herramienta, el sistema se repotenció en el año 2019, además de que se incluyeron nuevos paquetes actualizables para nuevas configuraciones. El software está en constante evolución, además de que su mantenimiento se mantiene sin costes económicos, hablando en la parte de licenciamiento.

Varios son los beneficios que se pueden aprovechar en base a la aplicación de este tipo de sistemas como se pudo evidenciar en base a los datos obtenidos, incluso tomando en cuenta que el funcionamiento del servicio se proyecta para seguir activo en un contexto cronológico amplio.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Tras la investigación del caso, proceso de diseño, configuración, pruebas y resultados para la implementación de un servidor Web Caché y DNS Caché en la infraestructura de red de la Universidad Técnica del Norte, el desarrollo del proyecto se efectuó en base al diseño del modelo de dimensionamiento de recursos de hardware mediante una metodología por pruebas preliminares, la implementación se desplegó en base a software libre y finalmente se estableció la viabilidad de los servicios dentro del sistema de red, siendo mayormente satisfactorio el resultado para el servicio de DNS caché.
- Los servicios integrados al sistema hiperconvergente se diseñaron en base a los datos estadísticos obtenidos del modelo de dimensionamiento los cuales permitieron establecer las métricas necesarias para el correcto funcionamiento del servidor, de forma general, los requerimientos de hardware que se necesitaron fueron: un procesador que funcione a 2GHz en frecuencia de reloj y que posea al menos 2 núcleos de procesamiento, memoria RAM de 4GB y almacenamiento en disco duro de mínimo 15GB, dichos servidores convergen con aquellos que ya prestan servicios en el sistema, sin embargo el DNS caché entró en producción sustituyendo el antiguo servicio de DNS, y el Web caché únicamente como servidor de pruebas, por lo que no existió conflicto alguno en el proceso de implementación y funcionamiento con los otros servidores.
- Se concluye que, en base a los resultados obtenidos en las respuestas para la resolución de nombres de dominio, los tiempos para las consultas de los usuarios pueden reducir de forma promedio en un 61% a comparación de una consulta preliminar.

- El servidor de DNS caché mantiene un consumo promedio de 0,4Mbps en estaciones de tiempo de al menos 350 peticiones por segundo, además obtiene un promedio de 98% de aciertos de caché, por lo que se concluye que el sistema funciona satisfactoriamente.

RECOMENDACIONES

- Para la realización del diseño de los servicios que se implementaron se deben tomar en cuenta ciertas consideraciones, como la disponibilidad, confiabilidad, seguridad y operabilidad, todo en base a un modelo de dimensionamiento que permita establecer las métricas de consumo de recursos de hardware y de red.
- Se recomienda abiertamente el uso de este trabajo de titulación como base para futuras aplicaciones en la infraestructura de red de la institución, los datos estadísticos de cada caso pueden brindar información aplicable para nuevas configuraciones y rediseños del sistema actual.
- Se recomienda la implementación de dos servidores del tipo DNS caché, para configurar un sistema redundante, es decir, aplicar un servidor primario y un secundario, de tal forma que se mantenga una alta disponibilidad del servicio.
- El software implementado se mantiene en constantes actualizaciones ofreciendo complementos que permiten nuevas configuraciones y mejoras al sistema, es por ello que se recomienda su constante comprobación y monitoreo.
- Se recomienda el uso de herramientas administrativas como Netdata, Graylog o Unbound-DNS Monitoring para monitorear y administrar de manera óptima los servicios en funcionamiento, tanto los que se aplicaron como parte del presente proyecto como de los que ya existen dentro de la cartera de servicios.
- Se recomienda a la parte administrativa inspeccionar nuevas configuraciones para la implementación de iPv6 en los servicios de DNS así como también migrar el servidor con Windows Server 2008 en su totalidad, debido a que se pueden dar falencias en el sistema por la expiración de su soporte por parte de Microsoft.

REFERENCIAS BIBLIOGRÁFICAS

Abbate, J. (s.f.). *internet: su evolución y sus desafíos.* Obtenido de Open Mind BBVA: https://www.bbvaopenmind.com/wp-content/uploads/2009/02/BBVA-OpenMind-Internet-su-evolucion-y-sus-desafios-Janet-Abbate.pdf.pdf

AlmaLinux OS Foundation. (s.f. de s.f. de s.f.). *Logo AlmaLinux*. Obtenido de https://almalinux.org/

Asamblea Nacional. (2018). *LEY ORGANICA DE EDUCACION SUPERIOR, LOES.* Quito: Consejo de Educación Superior.

Awati, R. (21 de junio de 2022). Obtenido de Whatls.com: https://www.techtarget.com/whatis/definition/World-Wide-Web

Carranza, A. (11 de Noviembre de 2021). *¿Qué es un web service?* Obtenido de crehana: https://www.crehana.com/blog/desarrollo-web/que-es-web-service/

CISCO. (21 de abril de 2021). *Hoja de datos de interconexiones de estructura de la serie Cisco UCS 6400*. Obtenido de Cisco Systems: https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/datasheet-c78-741116.html

Deyimar, A. (25 de agosto de 2022). *¿Qué es la caché web?* Obtenido de HOSTINGUER: https://www.hostinger.es/tutoriales/cache-web

Donate, F. (2017). *Transmisión de imágenes de vídeo mediante Servicios Web XML sobre J2ME.* Sevilla: Universidad de Sevilla.

escope. (20 de Enero de 2020). *Qué es DNS: Tipos y Ventajas*. Obtenido de ESCOPE Informática SL: https://ecope.es/blog/dns-granada/

Fernández, S. (s.f.). *TEORÍA DE COLAS: MODELO M/M/1*. Madrid: Universidad Autónoma de Madrid. Obtenido de Portal Estadística Aplicada.

Gómez, C., Sepúlveda, L., & Candela, C. (2012). Servidor Proxy Caché: Comprensión y asimilación tecnológica. *INGE CUC*, 149-162.

Guijarro, Á. (2012). *Protocolo HTTP*. Obtenido de WordPress.com: https://alvaroprimoguijarro.wordpress.com/

IBM. (7 de Junio de 2022). *Proxy de almacenamiento en caché*. Obtenido de https://www.ibm.com/docs/en/was-nd/9.0.5?topic=overview-caching-proxy

Jeftovic, M. E. (2015). *Managing Mission-Critical Domains and DNS.* O'REILLY.

Liberatori, M. (2018). *Redes de Datos y sus Protocolos.* Mar de la Plata: EUDEM.

López, A. (s.f.). *Guía de Seguridad en Servicios.* Madrid: Instituto Nacional de Tecnologías de la Comunicación.

MDN contributors. (07 de noviembre de 2022). *Cache-Control*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Cache-Control

Pardo, S. (30 de agosto de 2021). *¿Cómo funciona la web?* Obtenido de CREHANA: https://www.crehana.com/blog/transformacion-digital/como-funciona-la-web/

Pramatarov, M. (10 de Junio de 2021). *Caché de DNS*. Obtenido de ClouDNS: https://www.cloudns.net/blog/dns-cache-explained/#:~:text=What%20is%20DNS%20cache%3F,for%20IPv6%2C%20etc.).

Pukocz, E. (03 de noviembre de 2019). *Partes de una URL*. Obtenido de EDYTAPUKOCZ: https://edytapukocz.com/url-partes-ejemplos-facil/

Río, N. d. (2015). *Diseño e implementación de una solución de administración de tráfico de red basada en DNS y chequeos de disponibilidad.* La Plata: Universidad Nacional de La Plata.

Sánchez, E., Figueroa, D., Gamarra, Á., & Mayorga, J. (s.f.). *Implementación de un Servidor DNS Seguro basado en Pi-Hole utilizando un entorno virtualizado.* Salta: Universidad Nacional de Salta.

Spera, C. (2013). Software Defined Network: el futuro de las arquitecturas de red. *Data Center*, 42-45.

TechTarget. (Agosto de 2019). *Servidores Caché*. Obtenido de Whatls: https://www.techtarget.com/whatis/definition/cache-server

Universidad Técnica del Norte. (s.f.). *CAMPUS UNIEVRSITARIOS*. Obtenido de UNIVERSIDAD TÉCNICA DEL NORTE: https://www.utn.edu.ec/campus-universitarios/

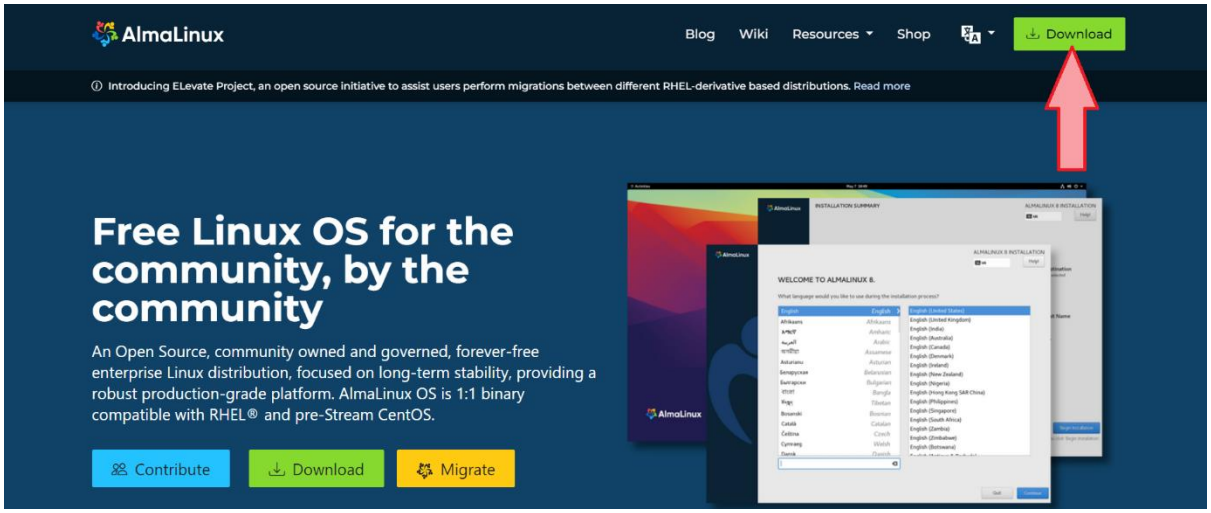
Vergara, A. (2017). *Servidores DNS: ventajas y desventajas*. Obtenido de Facilcloud: https://www.facilcloud.com/noticias/servidores-dns-ventajas-y-desventajas/

ANEXO 1

Guía de instalación y configuración de AlmaLinux (Minimal)

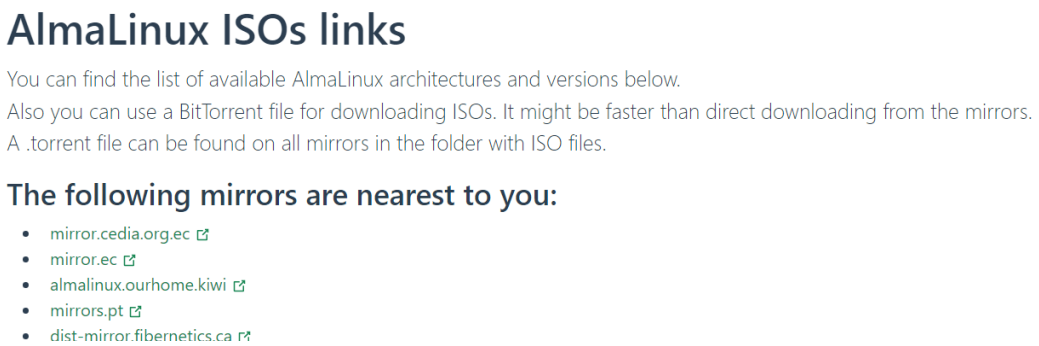
Instalación.

1. Acceder a la página Web oficial de AlmaLinux mediante el enlace y ubicar el apartado de “Download”

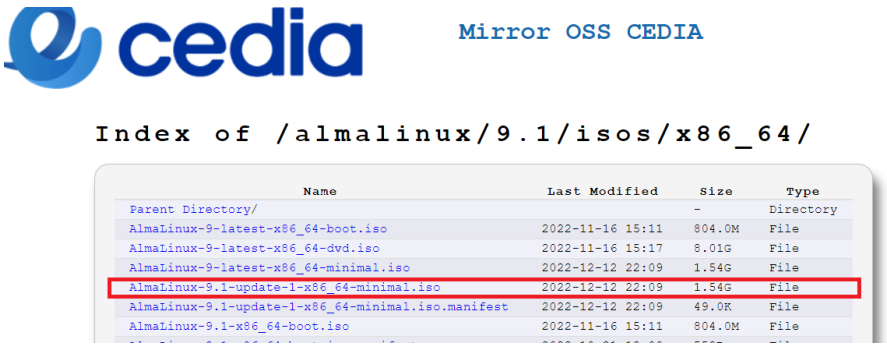


2. Elegir la arquitectura para el Sistema Operativo en la versión más estable, o más reciente, en este caso será una de 64 Bits, versión 9.1:

1. Ahora se debe buscar el “Mirror” o espejo para descargar el archivo de instalación, para ello buscaremos los servidores de Cedia, que son de los más seguros de acceder, en este caso directamente de Ecuador (primer enlace):



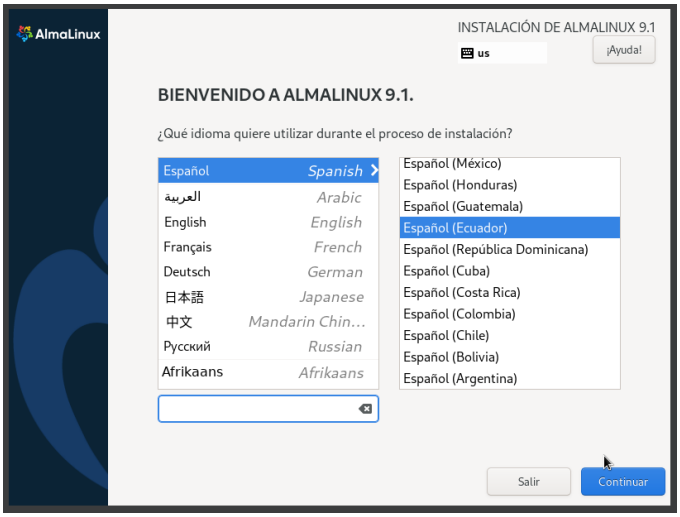
2. Se abrirá la página donde haremos clic en el archivo que se desea descargar, en este caso, será el archivo .iso de la versión Minimal:



3. Una vez descargado el archivo, es montado directamente en la máquina virtual asignada por el Administrador donde se procede a su instalación:

4. Una vez aquí empezamos con la instalación de AlmaLinux 9.1 con la primera opción que aparece en la interfaz, esperamos que el proceso se complete:

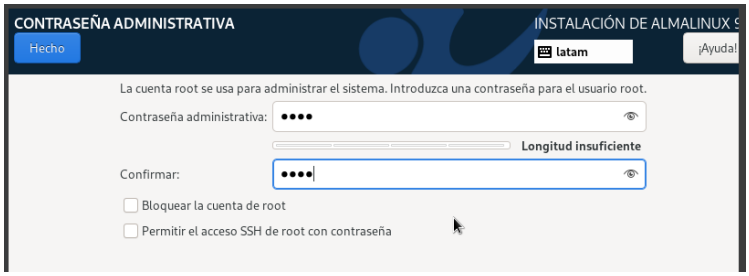
5. Se debe elegir el idioma y la región, esto para que el Sistema Operativo configure automáticamente la fecha, hora y teclado adecuados.



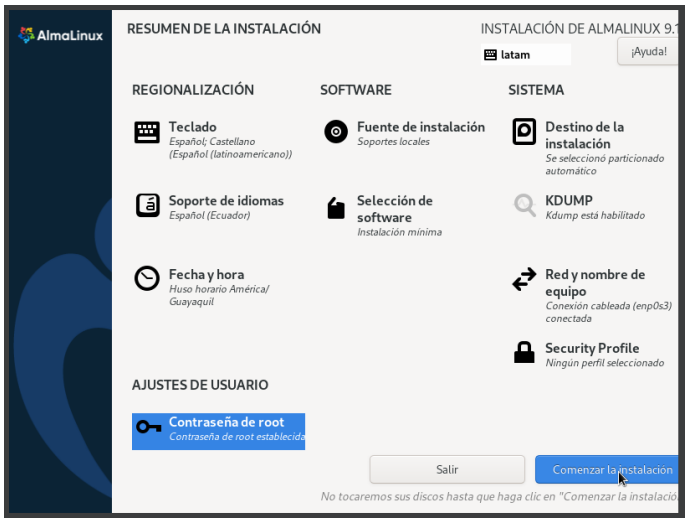
6. Hacemos clic en Comenzar y tendremos que modificar ciertas configuraciones, pues el proceso de instalación no dejará pasar a los siguientes pasos mientras no se corrijan los errores:

7. Se procede a la asignación del disco duro para que albergue el sistema con la partición asignada en la máquina virtual, la configuración de almacenamiento debe estar en “Automática”, finalmente se procede a dar clic en “Hecho”:

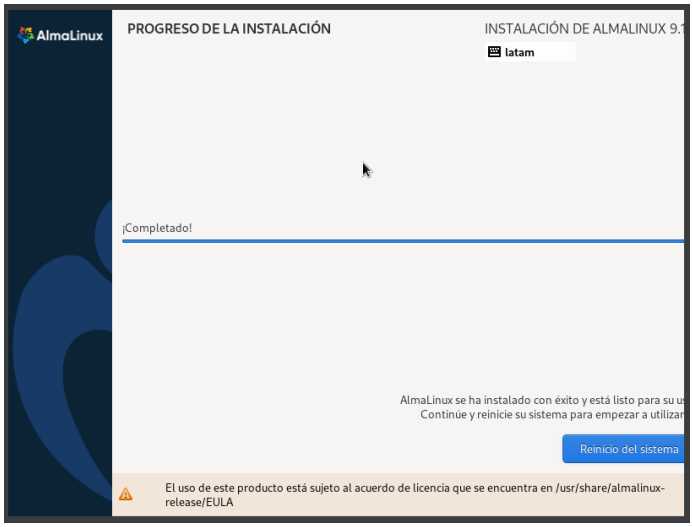
8. A continuación, se debe asignar un usuario Root, necesario para la configuración de cualquier arquitectura que se vaya a instalar, puesto que ciertos directorios y configuraciones necesitan un usuario de este tipo para poder tener acceso a ellos:



9. Finalmente, ya se habilitará el botón de “Continuar la instalación”:



10. Esperamos a que el proceso de instalación termine, y procedemos al reinicio del sistema:



Primeros Pasos con AlmaLinux

1. En primer lugar, hay que actualizar e instalar ciertos paquetes necesarios para una configuración general del sistema con los comandos:

```
[root@localhost ~]# dnf -y update_
```

```
[root@localhost ~]# dnf clean all
28 files removed
```

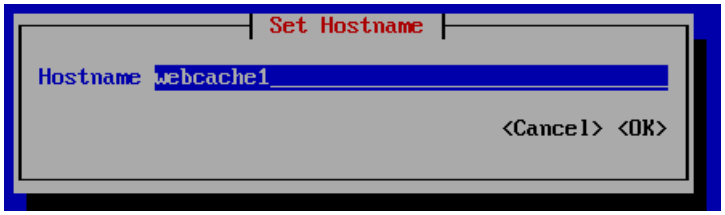
```
[root@localhost ~]# dnf -y install epel-release
```

```
[root@localhost ~]# dnf -y install net-tools neofetch wget screen tar_
```

2. Eliminamos el archivo creado por Anaconda que ocasiona que no se deniegue el acceso “ssh” por root:

```
[root@localhost ~]# rm -rf /etc/ssh/sshd_config.d/01-permitrootlogin.conf
```

3. Vamos a modificar el Hostname para identificar al servidor, en este caso será el servidor de Web Caché con el comando “*nmtui-hostname*”, aparecerá la siguiente ventana, donde asignaremos el nuevo Hostname:



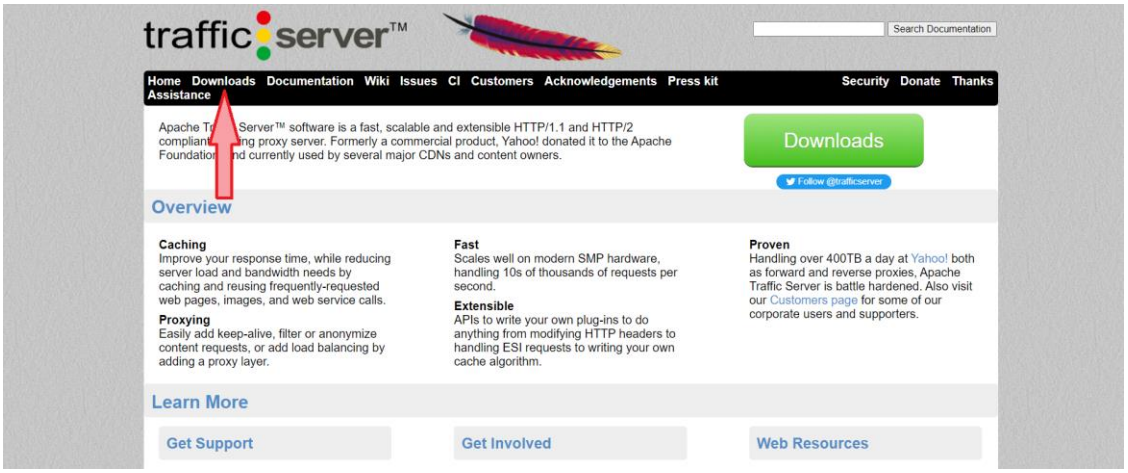
4. Finalmente revisamos la versión del sistema operativo, para comprobar que todo funciona correctamente con el comando "*cat /etc/almalinux-release*"

ANEXO 2

Guía de instalación y configuración de Apache Traffic Server

Instalación.

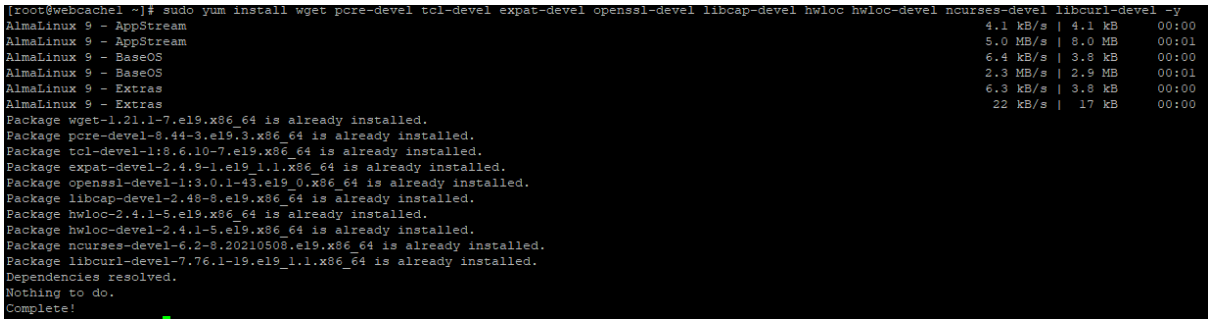
1. Nos dirigimos hacia la página web de apache Traffic Server mediante el enlace https://trafficserver.apache.org/ y nos vamos al apartado de “Downloads”



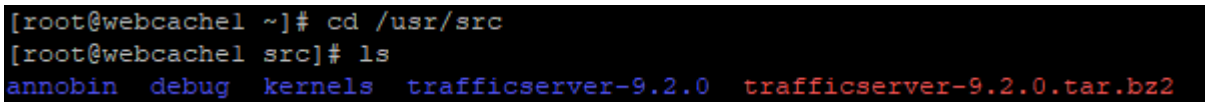
2. Nos dirigirá hacia el enlace de descarga donde haremos clic en Download Traffic Server en su versión 9.2.0:

3. Ubicamos el apartado HTTP y copiamos en enlace que se ubica a sus pies, dependerá de la versión que escoja para instalar:

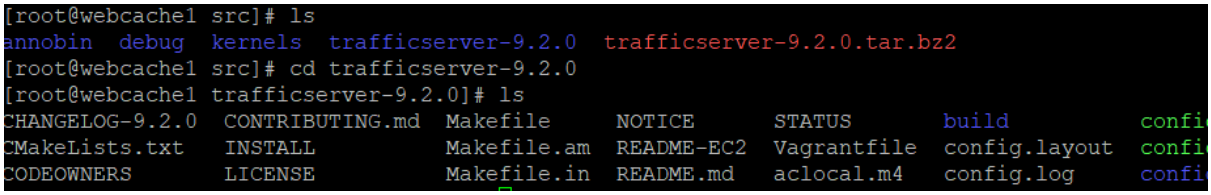
4. Iniciamos nuestra máquina virtual con AlmaLinux corriendo, e ingresamos en modo root, pero primero instalaremos librerías y complementos necesarios para la instalación de Traffic Server mediante el comando "*sudo yum install wget pcre-devel tcl-devel expat-devel openssl-devel libcap-devel hwloc hwloc-devel ncurses-devel libcurl-devel -y*"



5. Vamos a descargar trafficserver mediante el enlace que habíamos copiado antes con los comandos "*cd /usr/src*" y seguido el comando "*sudo wget https://dlcdn.apache.org/trafficserver/trafficserver-9.2.0.tar.bz2*"

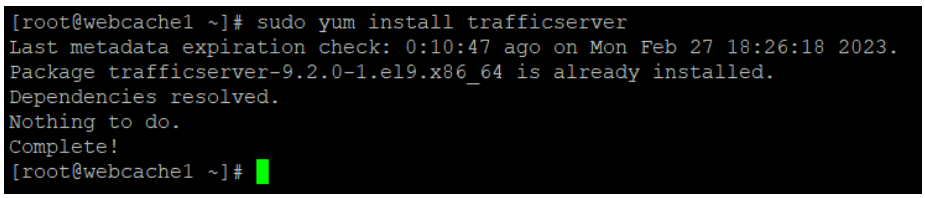


6. Dentro del fichero encontraremos los archivos descargados, los cuales descomprimiremos mediante el comando "*sudo tar xjvf trafficserver-9.2.0.tar.bz2*" finalmente entramos al directorio de trafficserver-9.2.0 mediante el comando "*cd trafficserver-9.2.0*"



7. Ahora compilamos trafficserver mediante el comando "*sudo ./configure --prefix=/opt/ts --with-group=trafficserver --with-user=trafficserver*" cuando termine

de compilar ejecutamos los comandos de comprobación "*sudo make*" y finalmente el comando de instalación "*sudo make install*"

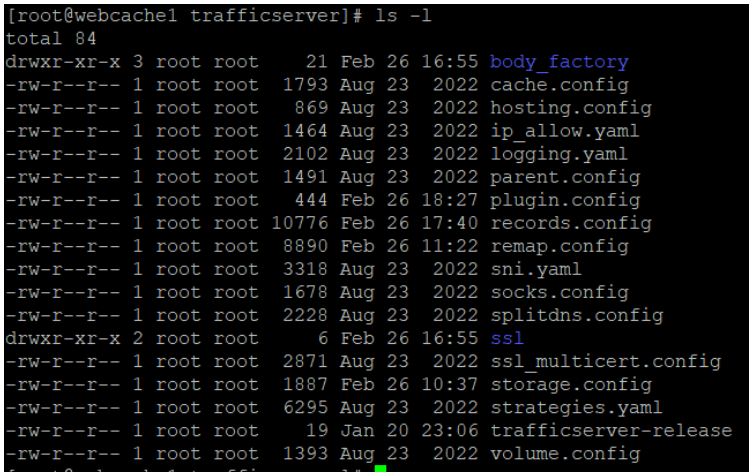


Tomar en cuenta que los desarrolladores recomiendan tener instalado las siguientes librerías para el correcto funcionamiento del software, empezando de su instalación:

- pkgconfig
- libtool
- C++ compiler (gcc >= 4.3 or clang > 3.0)
- GNU make
- OpenSSL or BoringSSL
- pcre
- libcap
- flex (for TPROXY)
- hwloc
- lua
- zlib
- curses (for traffic_top)
- curl (for traffic_top)

Configuración.

La configuración de Traffic Server se basa directamente en la modificación de los parámetros de los archivos que se encuentran en el directorio "*/etc/trafficserver*" se accede a él mediante el comando "*cd*" dentro se encuentran los siguientes archivos:



Para un acceso rápido de los archivos se recomienda el uso de FTP, utilizando cualquier software que permita esta conexión, esto con el fin de poder configurar los archivos de trafficserver de forma más eficiente, dicho esto, a continuación, se presentan el listado y que contiene cada uno:

- *cache.config*

Determina si Traffic Server almacena objetos en caché, cómo y durante cuánto tiempo según el destino, el cliente, los componentes de URL, etc.

- *hosting.config*

Permite a los administradores de servidores de tráfico asignar volúmenes almacenados en caché a servidores específicos o dominios de origen.

- *ip_allow.yaml*

Controla el acceso a las cachés de Traffic Server en función de la red y la dirección IP de origen, incluida la restricción de los métodos HTTP individuales.

- *logging.yaml*

Establece formatos de archivo de registro personalizados, filtros y opciones de procesamiento.

- *parent.config*

Configura proxies principales en diseños de almacenamiento en caché jerárquicos.

- *plugin.config*

Controla los complementos cargables en tiempo de ejecución disponibles para Traffic Server, así como sus configuraciones.

- *records.config*

Contiene muchas variables de configuración que afectan el funcionamiento de Traffic Server.

- *remap.config*

Define las reglas de mapeo utilizadas por Traffic Server para enrutar correctamente todas las solicitudes entrantes.

- *splitdns.config*

Configura servidores DNS para usar bajo condiciones específicas.

- *ssl_multicert.config*

Configura Traffic Server para usar diferentes certificados de servidor para la terminación de SSL cuando se escucha en varias direcciones o cuando los clientes emplean SNI.

- *sni.yaml*

Configura el enrutamiento de capa 4 basado en SNI.

- *storage.config*

Configura todos los dispositivos de almacenamiento y las rutas que se utilizarán para la memoria caché de Traffic Server.

- *strategies.yaml*

Configura las estrategias de NextHop utilizadas con remap.config

- *volume.config*

Define el uso del espacio de caché por protocolos individuales.

De estos archivos de configuración, no todos ellos los vamos a usar en nuestro caso, debido a que la herramienta ofrece múltiples configuraciones para cada uno de los tipos de servidor que se requiera, para configurar los dos escenarios planteados se configura únicamente los archivos:

- *records.config*
- *remap.config*
- *plugin.config*
- *storage.config*

Según sea el caso, cada uno de los archivos alberga los parámetros para que el servicio funcione correctamente.

El almacenamiento en caché de proxy HTTP nos permite guardar copias de objetos web a los que accedemos con frecuencia, como documentos, imágenes y artículos, y luego entregar esa información a los usuarios según sea necesario. Esto mejora el rendimiento y libera ancho de banda de Internet para otras tareas.

El servidor de tráfico recibe solicitudes de objetos web de los clientes. Usando la dirección del objeto, Traffic Server intenta encontrar el objeto solicitado en su base de datos de objetos (caché). Si el objeto está en la memoria caché, el software verifica si el objeto es lo suficientemente nuevo para "servir". Si es nuevo, lo envía al host cliente como un hit de caché.

Como este servidor está enfocado estrictamente al almacenamiento en caché, procedemos a la configuración de este parámetro, directamente en el archivo "*records.conf*"

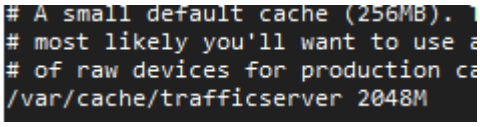
Para habilitar el almacenamiento de contenido de caché de objetos HTTP se establece el parámetro "*proxy.config.http.cache.http*" en INT 1, así:

```
##########################################################################
# Enable / disable HTTP caching. Useful for testing, but also as an
# overridable (per remap) config
##########################################################################
CONFIG proxy.config.http.cache.http INT 1
```

Y algo muy importante es también configurar el parámetro de puerto de escucha, por donde nuestro servidor va a recibir las peticiones de los clientes, para ello buscamos el parámetro "*proxy.config.http.server_ports*" donde configuramos el puerto que por defecto es el 8080 pero se puede personalizar:

```
##########################################################################
# Specify server addresses and ports to bind for HTTP and HTTPS. Docs:
#    https://docs.trafficserver.apache.org/records.config#proxy.config.http.server_ports
##########################################################################
CONFIG proxy.config.http.server_ports STRING 8080:ipv4
```

Otra configuración importante es cambiar el valor de memoria RAM que utilizará el servidor para almacenar hits de caché temporales, esto se lo realiza en el archivo "*storage.config*" al final del archivo cambiamos el valor por defecto al que se considere necesario, en este caso le hemos aumentado a 2 Gb, cambiando el valor numérico a "2048M"

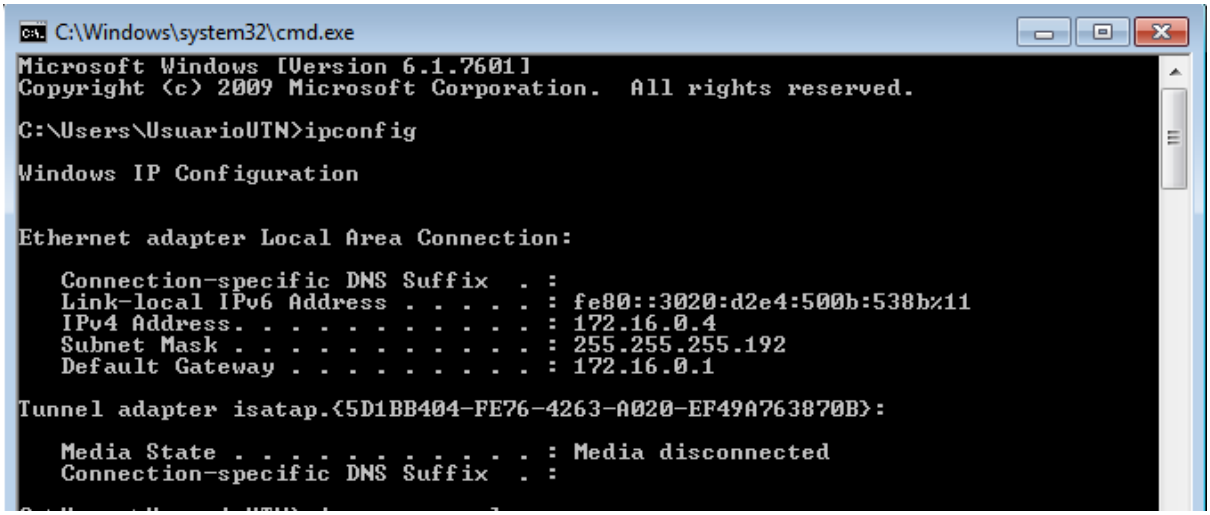


ANEXO 3

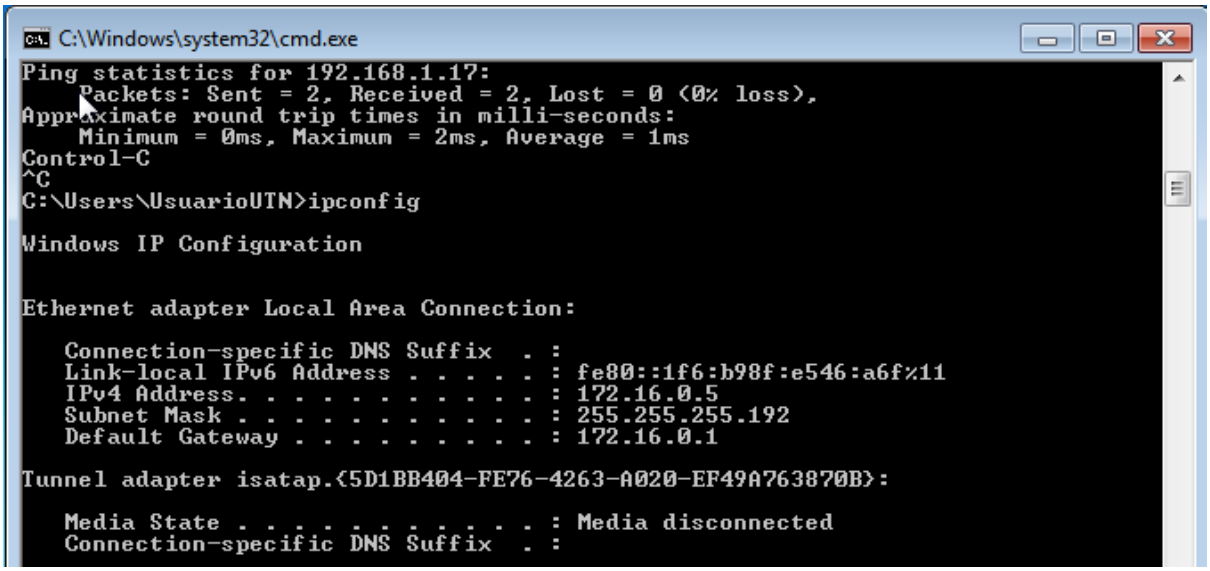
Guía de configuración de Traffic Server como Servidor Caché Proxy de Reenvío.

Para empezar, se estima que el direccionamiento lógico de los clientes está establecido dentro del rango de la clase a la que se le ha asignado, para ello comprobamos esa configuración en cada cliente:

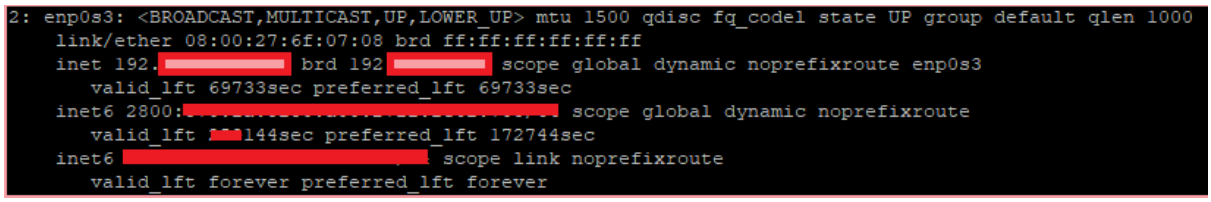
Cliente 1:



Cliente 2:

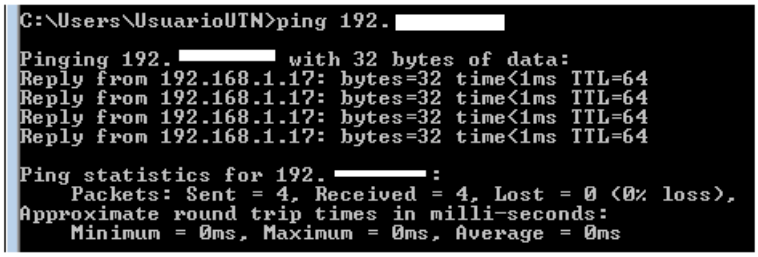


Continuamos con la configuración, en este caso nos vamos a la máquina virtual, donde ya está instalado nuestro Apache Traffic Server, donde se configura en primer lugar el direccionamiento en su interfaz de entrada también con el direccionamiento asignado por la parte administrativa así, mediante el comando "*nmtui*" se puede configurar su interfaz:



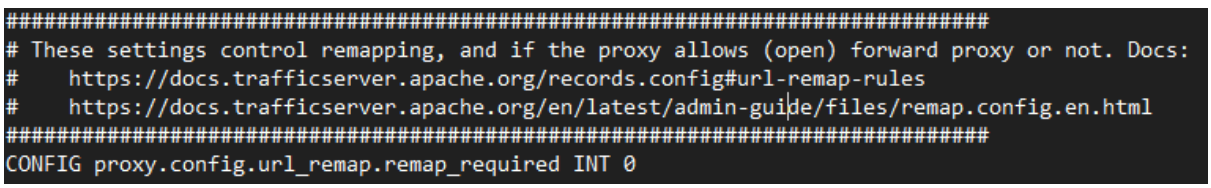
Cabe mencionar que a esta interfaz se le asignó una dirección IP de clase C.

Podremos comprobar si los clientes tienen conexión hacia el servidor mediante un ping de cliente – servidor:



Ahora procedemos a la configuración de los archivos de trafficserver. El archivo que se va a configurar es el "*records.config*" con las siguientes directrices:

Primero se le concede el permiso a Traffic Server que procese las solicitudes de los clientes que no están configurados explícitamente en las reglas de reasignación, entonces dentro del archivo buscamos el siguiente parámetro:
"*proxy.config.url_remap.remap_required*" y lo establecemos en 0 (cero):



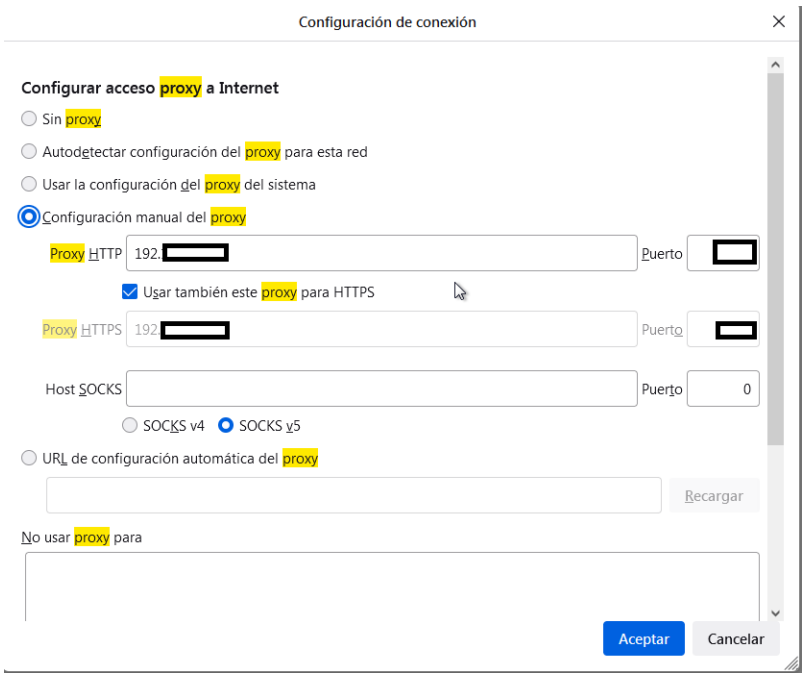
Como el servidor estará operando en forma de reenvío también hay que deshabilitar el soporte para proxy inverso mediante el parámetro "*proxy.config.reverse_proxy.enabled*" también estableciendo el INT 0:

```
    # https://docs.trafficserver.apache.org/records.config#reverse-proxy
CONFIG proxy.config.reverse_proxy.enabled INT 0
```

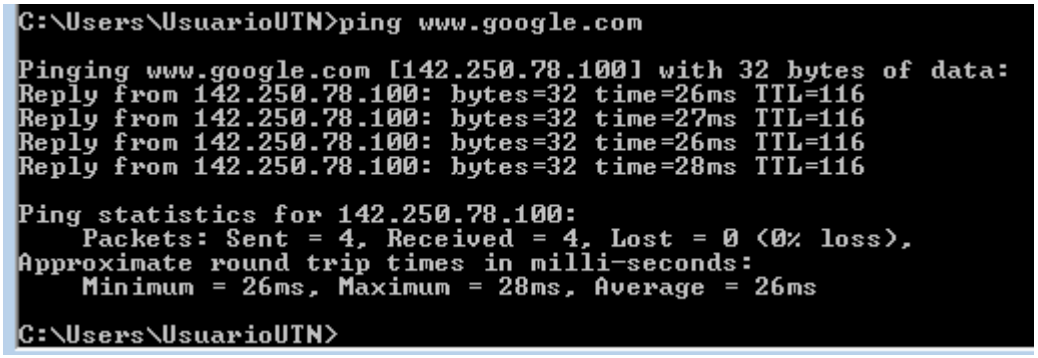
Continuamos reiniciando el servicio trafficserver y comprobamos el estado:

```
[root@webcachel ~]# systemctl status trafficserver
● trafficserver.service - Apache Traffic Server is a fast, scalable and extensible caching proxy server.
     Loaded: loaded (/usr/lib/systemd/system/trafficserver.service; enabled; vendor preset: disabled)
     Active: active (running) since Sun 2023-02-26 17:40:20 -05; 1 day 4h ago
       Docs: man:traffic_server(8)
   Main PID: 58364 (traffic_manager)
      Tasks: 21 (limit: 11077)
     Memory: 82.8M
        CPU: 3min 3.588s
     CGroup: /system.slice/trafficserver.service
             ├─58364 /usr/bin/traffic_manager
             └─58369 /usr/bin/traffic_server -M --httpport 8080:fd=8

Feb 26 17:40:20 webcachel systemd[1]: Started Apache Traffic Server is a fast, scalable and extensible c
Feb 26 17:40:20 webcachel traffic_manager[58364]: [E. Mgmt] log ==> [TrafficManager] using root director
Feb 26 17:40:20 webcachel traffic_manager[58364]: NOTE: --- Manager Starting ---
Feb 26 17:40:20 webcachel traffic_manager[58364]: NOTE: Manager Version: Apache Traffic Server - traffic_
Feb 26 17:40:23 webcachel traffic_server[58369]: NOTE: --- traffic_server Starting ---
Feb 26 17:40:23 webcachel traffic_server[58369]: NOTE: traffic_server Version: Apache Traffic Server - t
[root@webcachel ~]#
```

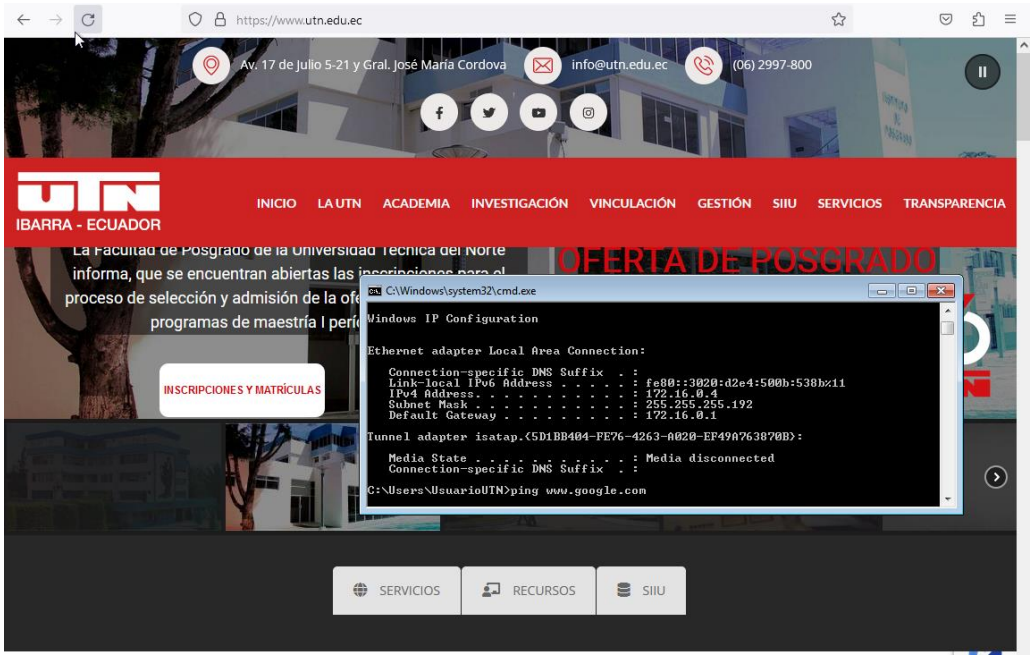
Ahora bien, del lado del cliente se configura la dirección IP del servidor con su puerto, así:



Comprobamos si tenemos conectividad hacia Internet para verificar que el servicio si está activo y en funcionamiento:



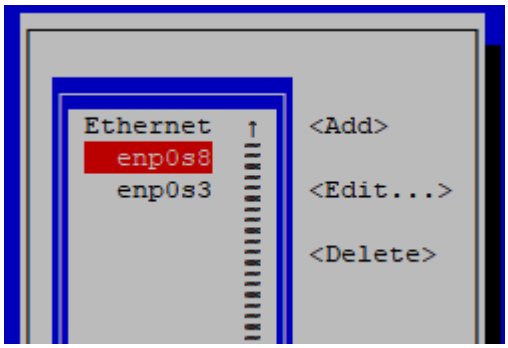
Y también hacemos lo mismo con el navegador ingresando a una página web:



Guía de configuración de Traffic Server como Servidor Caché Proxy Transparente.

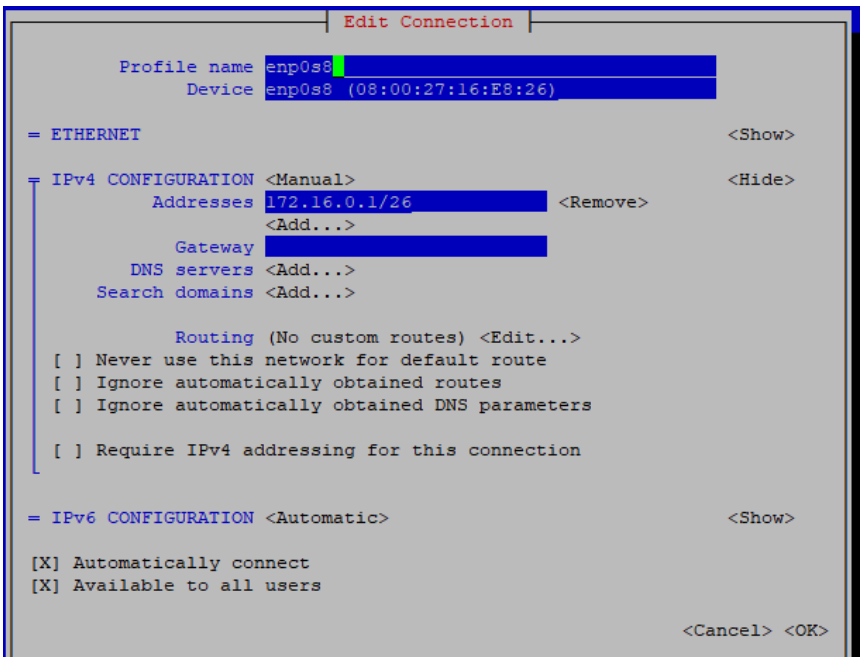
La configuración de los clientes es la misma a excepción de que aquí no se especifica directamente la dirección IP y puerto del servidor proxy, pues los paquetes son enrutados directamente a la interfaz de entrada del servidor sin que el cliente tuviese conocimiento alguno, así en cada navegador, la configuración de proxy es nula:

El primer paso es configurar las interfaces del servidor, para este caso se activan dos interfaces, que como se mencionó una será de entrada donde interceptará los paquetes de datos y la otra será de salida, que enrutará esos paquetes hacia Internet, con "*nmtui*" configuramos estas interfaces:

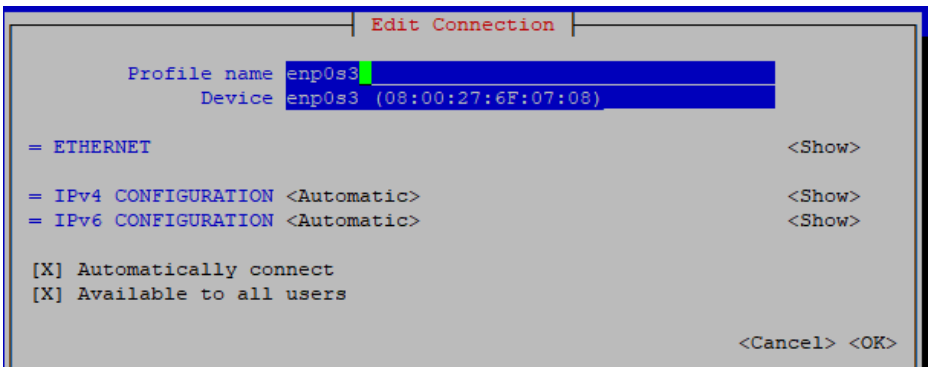


Para nuestro escenario la interfaz enp0s8 está conectada a la red LAN que se configuró en base a la topología y tendrá un direccionamiento lógico clase B (172.x.x.x/26) y

la interfaz enp0s3 será la que tenga salida a Internet, en este caso es un puerto cualquiera que se nos asignó para fin de la prueba por lo que tomará la dirección IP asignada por el DHCP de la institución:



La interfaz enp0s8 solamente se le asigna una dirección IP que servirá como puerta de enlace de la red LAN de prueba donde se ubican los clientes es por ello que no se configura ni Gateway ni DNS.

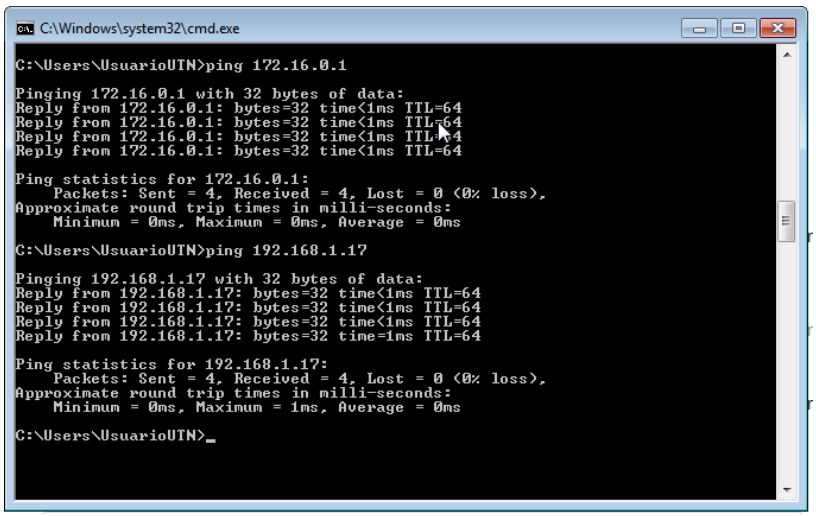


Y como se había mencionado el direccionamiento de la interfaz enp0s3 se recibe automáticamente puesto que esta será nuestra salida a Internet.

Así comprobamos el direccionamiento de cada puerto mediante el comando "*ip address*":

```
[root@webcache1 ~]# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:6f:07:08 brd ff:ff:ff:ff:ff:ff
    inet 192.[redacted] 26 brd 192.[redacted] scope global dynamic noprefixroute enp0s3
       valid_lft 64233sec preferred_lft 64233sec
    inet6 2800:[redacted]'64 scope global dynamic noprefixroute
       valid_lft 258920sec preferred_lft 172520sec
    inet6 [redacted]/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:16:e8:26 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.1/26 brd 172.16.0.63 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::7771:b92e:887f:3a1e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: br0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
[root@webcache1 ~]#
```

Una vez configuradas las interfaces comprobamos la conectividad entre cliente-servidor, se debería, en teoría, poder hacer un ping hacia ambas interfaces:



Ahora, el siguiente paso es configurar "*iptables*" esto dependerá del modo de configuración de las reglas, y como las ejecute, en lo personal, la utilización de scripts siempre es recomendable, verificamos si el servicio está disponible y activo:

```
[root@webcachel ~]# systemctl status iptables
● iptables.service - IPv4 firewall with iptables
     Loaded: loaded (/usr/lib/systemd/system/iptables.service; enabled; vendor preset: disabled)
     Active: active (exited) since Sat 2023-02-25 15:02:21 -05; 2 days ago
   Main PID: 675 (code=exited, status=0/SUCCESS)
        CPU: 27ms

Notice: journal has been rotated since unit was started, output may be incomplete.
[root@webcachel ~]#
```

Ahora establecemos un script para enrutar los paquetes de la LAN hacia Internet, y al mismo tiempo se realiza una regla que redirija los paquetes, en este caso del puerto 80 hacia el puerto donde trabaja Traffic Server, para ello abrimos un script en el directorio "*/etc/*" con un nombre cualquiera con extensión "*.sh*" en este caso creamos un script llamado "*firewall.sh*":

```
[root@webcachel ~]# vi /etc/firewall.sh
```

Tenemos vista hacia un nuevo script donde configuraremos todas las reglas que nos permitan realizar el proceso mencionado, en primer lugar se eliminan reglas de iptables previas que hubiera y cadenas definidas por el usuario con las siguientes líneas:

```
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
```

Ahora se permite el acceso al firewall desde la Red LAN con la siguiente regla:

```
iptables -A INPUT -s 172.16.0.0/26 -i enp0s8 -j ACCEPT
```

Y se procede a compartir Internet desde nuestro router Linux hacia los clientes:

```
iptables -t nat -A POSTROUTING -s 172.16.0.0/26 -o enp0s3 -j MASQUERADE
```

Ahora redirigimos los paquetes hacia el puerto de escucha del servidor para que nos ayude en la filtración y almacenamiento del caché mediante la regla:

```
iptables -t mangle -A PREROUTING -i eth1 -p tcp -m tcp --dport 80 -j TPROXY \
   --on-ip 0.0.0.0 --on-port 8080 --tproxy-mark 1/1
```

Finalmente se guardan los cambios mediante el comando:

```
service iptables save
```

Para guardar el script, presionamos "*:wq*" de esta forma ya solo tendremos que ejecutar este archivo mediante el comando "*sh*":

```
[root@webcache1 ~]# sh /etc/firewall.sh
iptables: Saving firewall rules to /etc/sysconfig/iptables: [  OK  ]
```

Así entonces si al final de la línea nos aparece un OK quiere decir que las reglas se han establecido correctamente, podremos verificar las reglas que están establecidas mediante el comando "*iptables -t filter -L*":

```
[root@webcache1 ~]# iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  172.16.0.0/26        anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@webcache1 ~]#
```

La configuración de Traffic Server únicamente se realiza dentro del archivo "*records.conf*" con tres parámetros:

En "*proxy.config.http.server_ports*" se establece el puerto que se configuró dentro de la regla de iptables donde se redirigen las peticiones, en este caso el puerto por defecto:

```
##############################################################################
# Specify server addresses and ports to bind for HTTP and HTTPS. Docs:
#    https://docs.trafficserver.apache.org/records.config#proxy.config.http.server_ports
##############################################################################
CONFIG proxy.config.http.server_ports STRING 8080:ipv4
```

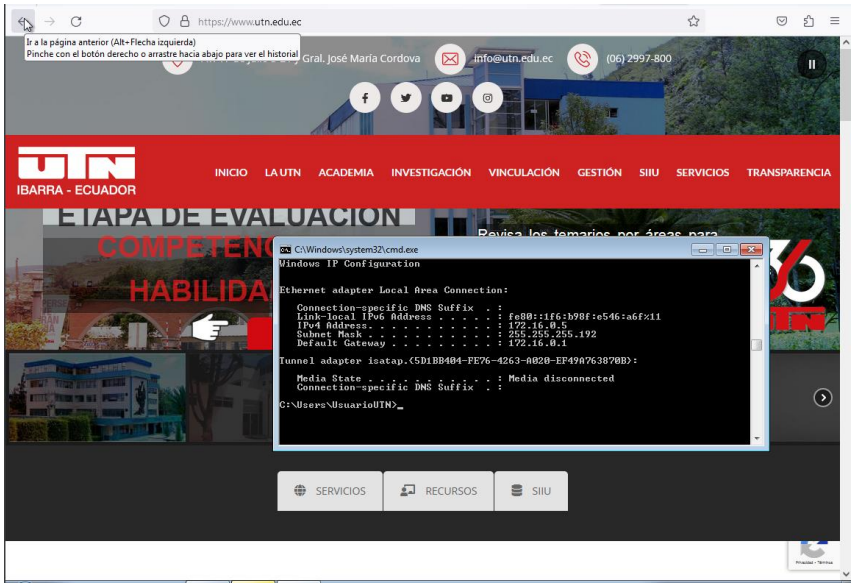
En "*proxy.config.reverse_proxy.enabled*" por defecto el valor de INT 1:

```
    # https://docs.trafficserver.apache.org/records.config#reverse-proxy
CONFIG proxy.config.reverse_proxy.enabled INT 1
```

En "*proxy.config.url_remap.remap_required*" por defecto el valor de INT 0:

```
###############################################################################
# These settings control remapping, and if the proxy allows (open) forward proxy or not. Docs:
#    https://docs.trafficserver.apache.org/records.config#url-remap-rules
#    https://docs.trafficserver.apache.org/en/latest/admin-guide/files/remap.config.en.html
###############################################################################
CONFIG proxy.config.url_remap.remap_required INT 1
```

Con estas configuraciones, el servidor de forma transparente ya estaría en funcionamiento, desde el cliente podemos comprobar el acceso a Internet:



ANEXO 4

Guía de instalación y configuración de Unbound

Instalación

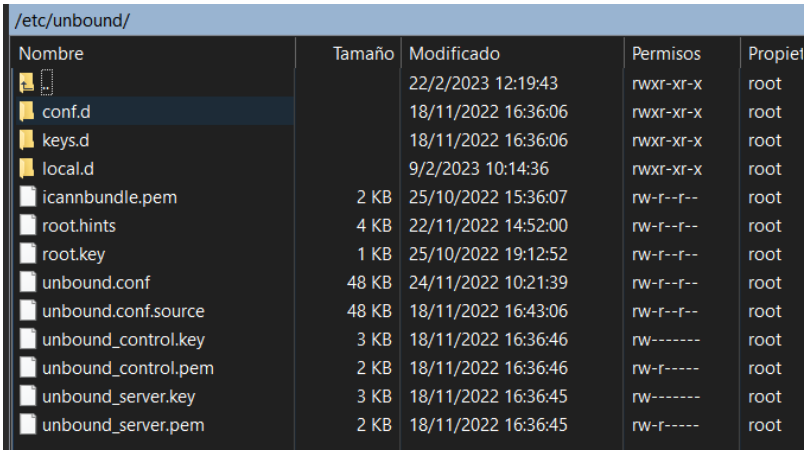
En primer lugar, procedemos a la instalación mediante el comando "*yum install unbound bind-utils net-tools*":

```
[root@dnscachel ~]# yum install unbound bind-utils net-tools
Última comprobación de caducidad de metadatos hecha hace 2:32:41, el lun 27 feb 2023 23:22:47.
El paquete unbound-1.16.2-2.el9.x86_64 ya está instalado.
El paquete bind-utils-32:9.16.23-5.el9_1.x86_64 ya está instalado.
El paquete net-tools-2.0-0.62.20160912git.el9.x86_64 ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
[root@dnscachel ~]#
```

Se realiza un respaldo del archivo original de configuración mediante el comando "*cp /etc/unbound/unbound.conf /etc/unbound/unbound.conf.orig*" aquí pondremos como nombre de archivo cualquiera que pueda recordar.

```
[root@dnscachel ~]# cp /etc/unbound/unbound.conf /etc/unbound/unbound.conf.orig
```

Y lo podemos verificar ingresando al directorio especificado, mediante los comandos "*cd*" y "*ls*" o bien como nuestro caso usando un administrador de archivos con FTP:



Se procede a hacer el cambio de dueño del directorio con el comando "*chown unbound:unbound /etc/unbound*" esto para realizar las configuraciones sin tener el problema de acceso.

```
[root@dnscachel ~]# chown unbound:unbound /etc/unbound
```

A partir de aquí no hay más configuraciones que realizar más que comprobar el funcionamiento del servicio:

```
[root@dnscachel ~]# systemctl status unbound
● unbound.service - Unbound recursive Domain Name Server
     Loaded: loaded (/usr/lib/systemd/system/unbound.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2023-02-27 12:51:10 -05; 13h ago
    Process: 159987 ExecStartPre=/usr/sbin/unbound-checkconf (code=exited, status=0/SUCCESS)
    Process: 159989 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_UNBOUND_ANCHOR" == "yes" ]; then /usr
   Main PID: 159993 (unbound)
      Tasks: 4 (limit: 22975)
     Memory: 578.0M
        CPU: 8min 1.869s
     CGroup: /system.slice/unbound.service
             └─159993 /usr/sbin/unbound -d

feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:3] info: 172.17.57.189 ctldl.windowsup
feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:3] info: 172.17.57.189 ctldl.windowsup
feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:2] info: 172.17.57.189 ctldl.windowsup
feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:2] info: 172.17.57.189 ctldl.windowsup
feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:2] info: 172.16.8.20 time.windows.com.
feb 28 02:03:06 dnscachel.utn.edu.ec unbound[159993]: [159993:2] info: 172.16.8.20 time.windows.com.
feb 28 02:03:07 dnscachel.utn.edu.ec unbound[159993]: [159993:0] info: 172.16.14.253 tls.telemetry.s
feb 28 02:03:07 dnscachel.utn.edu.ec unbound[159993]: [159993:0] info: 172.16.14.253 tls.telemetry.s
feb 28 02:03:07 dnscachel.utn.edu.ec unbound[159993]: [159993:3] info: 172.16.14.253 tls.telemetry.s
feb 28 02:03:07 dnscachel.utn.edu.ec unbound[159993]: [159993:3] info: 172.16.14.253 tls.telemetry.s
lines 1-22/22 (END)
```

Esto quiere decir que el software funciona correctamente.

Configuración.

Para nuestro caso de implementación se utilizará una configuración extendida de la herramienta, y el archivo que se debe modificar es "*unbound.conf*" que se ubica en el directorio "/etc/unbound/".

Directamente como servidor se deben establecer los siguientes parámetros con sus respetivas métricas, aquí se detalla cada uno y qué función cumple:

El parámetro verbosity se configura en 5 niveles y cada uno de ellos nos proporciona diferentes tipos de información, 1 proporciona información operativa, el 2 nos da información operativa detallada, incluida información corta por consulta, el 3 brinda información de nivel de consulta, salida por consulta. el 4 da información de nivel de algoritmo y el 5 Registra la identificación del cliente para fallas de caché. El nivel 1 es el que viene por defecto:

```
# verbosity number, 0 is least verbose. 1 is default.
verbosity: 1
```

Interface, es un parámetro en el cual se debe especificar las interfaces por donde responden las consultas desde una dirección IP, cuando este parámetro se establece en 0.0.0.0 significa que une todas las interfaces disponibles:

```
# specify the interfaces to answer queries from by ip-address.
# The default is to listen to localhost (127.0.0.1 and ::1).
# specify 0.0.0.0 and ::0 to bind to all available interfaces.
# specify every interface[@port] on a new 'interface:' labelled line.
# The listen interfaces are not changed on reload, only on restart.
interface: 0.0.0.0
```

En port se establece el puerto por donde se van a responder las consultas:

```
# port to answer queries from
port: 53
```

Para los parámetros do-ip4, do-udp y do-tcp se establece yes o no, especificando permitir dicho protocolo, iPv4, TCP y UDP respectivamente:

```
# Enable IPv4, "yes" or "no".
do-ip4: yes
```

```
# Enable TCP, "yes" or "no".
do-tcp: yes
```

```
# Enable UDP, "yes" or "no".
# NOTE: if setting up an Unbound on tl
# disable UDP to avoid being used in D
do-udp: yes
```

En el parámetro access-control se especifican las redes y subredes las cuales tendrán acceso al servidor para realizar las consultas, en nuestro caso identificamos con un allow al final de cada subred para habilitar las consultas recursivas de cada segmento de vlan configuradas en la red de la institución:

```
# control which clients are allowed to make (recursive) queries
# to this server. Specify classless netblocks with /size and action.
# By default everything is refused, except for localhost.
# Choose deny (drop message), refuse (polite error reply),
# allow (recursive ok), allow_setrd (recursive ok, rd bit is forced on),
# allow_snoop (recursive and nonrecursive ok)
# deny_non_local (drop queries unless can be answered from local-data)
# refuse_non_local (like deny_non_local but polite error reply).
# access-control: 0.0.0.0/0 refuse
# access-control: 127.0.0.0/8 allow
# access-control: ::0/0 refuse
# access-control: ::1 allow
# access-control: ::ffff:127.0.0.1 allow
access-control: 12[redacted] allow
access-control: 10[redacted] allow
access-control: 17[redacted] allow
access-control: 19[redacted] allow
access-control: 17[redacted] allow
access-control: 17[redacted] allow
access-control: 17[redacted] allow
access-control: 17[redacted] allow
access-control: 17[redacted] allow
access-control: 17[redacted] allow
access-control: 11[redacted] allow
```

En el parámetro root-hints se establece un directorio donde se almacenará la información sobre los servidores tipo root necesarios para inicializar el caché de los servidores de nombres de dominio de Internet:

```
# file to read root hints from.
# get one from https://www.internic.net/domain/named.cache
# root-hints: ""
root-hints: "/etc/unbound/root.hints"
```

Se habilita hide-identity para no recibir consultas de id.server o hostname.bind para no revelar el nombre del host remoto. Así mismo se habilita el hide-versión para evitar las consultas de las versiones del servidor.

```
# enable to not answer id.server and hostname.bind queries.
hide-identity: yes

# enable to not answer version.server and version.bind queries.
hide-version: yes
```

harden-dnssec-stripped es un parámetro que se activa para el anclaje a zonas de confianza.

```
# Harden against receiving dnssec-stripped data. If you turn it
# off, failing to validate dnskey data for a trustanchor will
# trigger insecure mode for that zone (like without a trustanchor).
# Default on, which insists on dnssec data for trust-anchored zones.
harden-dnssec-stripped: yes
```

cache-min-ttl y cache-max-ttl son los parámetros que establecen el tiempo de vida mínimo y máximo de un registro guardado en la memoria caché, este valor se establece a conveniencia:

```
# the time to live (TTL) value cap for RRsets and messages in the
# cache. Items are not cached for longer. In seconds.
cache-max-ttl: 43200

# the time to live (TTL) value cap for negative responses in the cache
# cache-max-negative-ttl: 3600
```

Prefetch realiza la captación previa de entradas de caché de mensajes casi caducados:

```
# if yes, perform prefetching of almost expired message cache entries.
prefetch: yes
```

Num_threats establece el número de intentos de consulta:

```
# number of threads to create. 1 disables threading.
num-threads: 4
```

El parámetro msg-cache-slabs son el número de losas en el caché de mensajes. Las losas reducen la contención del bloqueo por hilos. Debe establecerse en una potencia de 2. El msg-cache-size establece el tamaño de memoria que usa un mensaje de caché.

```
# the amount of memory to use for the message cache.
# plain value in bytes or you can append k, m or G. default is "4Mb".
msg-cache-size: 128m

# the number of slabs to use for the message cache.
# the number of slabs must be a power of 2.
# more slabs reduce lock contention, but fragment memory usage.
msg-cache-slabs: 8
```

En private-address se configuran los rangos de red privados, el parámetro hace cumplir la privacidad de las redes, en este caso se ponen todas las redes privadas, de clase A, B y C:

```
# Enforce privacy of these addresses. Strips them away from answers.
# It may cause DNSSEC validation to additionally mark it as bogus.
# Protects against 'DNS Rebinding' (uses browser as network proxy).
# Only 'private-domain' and 'local-data' names are allowed to have
# these private addresses. No default.
private-address: 10.0.0.0/8
private-address: 172.16.0.0/12
private-address: 192.168.0.0/16
```

Una vez que se han configurado estos parámetros, se procede a establecer a Unbound como un servidor DNS autoritativo. Cuando una resolución recursiva recibe una respuesta de un servidor de nombres de nivel superior, esa respuesta dirige la resolución al servidor de nombres autoritativo.

Un servidor de nombres autoritativo contiene información específica de los nombres de dominio que sirve (por ejemplo, google.com) y puede devolver una resolución recursiva con la dirección IP del servidor que se encuentra en el registro DNS A, o devolver una resolución recursiva con el dominio de alias si el dominio tiene un registro CNAME (alias). el resolutor debe realizar recursivamente una nueva búsqueda de DNS para obtener un registro (generalmente un registro A que contiene la dirección IP) del servidor de nombres autorizado.

```
[root@dnscachel ~]# wget https://www.internic.net/domain/named.cache -O /etc/unbound/root.hints
```

Ahora se configura el archivo "*unbound.conf*"

```
[root@dnscachel ~]# root-hints: "/etc/unbound/root.hints"
```

Ahora verificamos la configuración del archivo unbound.conf

```
[root@dnscachel ~]# unbound-checkconf /etc/unbound/unbound.conf
unbound-checkconf: no errors in /etc/unbound/unbound.conf
```

Con el visto bueno, sin errores arrancamos el servicio, de esta forma continuamos con la configuración de la zona local de DNS, esto se realiza mediante un archivo "*zona-utn.conf*" que se alberga en el directorio *"/etc/unbound/local.d/"*.

Para esta configuración se asigna una dirección IP para cada una de las zonas que se mencionan a continuación, por cuestiones de seguridad no se hace pública la dirección IP:

- dnscache1.utn.edu.ec
- dnscache2.utn.edu.ec
- ntp.utn.edu.ec
- cctv.utn.edu.ec
- srv_eduroam.utn.edu.ec
- radius.utn.edu.ec
- dhcp.utn.edu.ec
- vcenter.utn.edu.ec
- servidorwsus.utn.edu.ec
- aplicaciones.utn.edu.ec
- repofica.utn.edu.ec
- telefonia.utn.edu.ec
- geoportal.utn.edu.ec
- appfica.utn.edu.ec
- appsms.utn.edu.ec
- coderepo.utn.edu.ec
- h1.utn.edu.ec
- eduvirtual.utn.edu.ec
- eduvirtual1.utn.edu.ec
- eduvirtual2.utn.edu.ec

- eduvirtual3.utn.edu.ec
- eduvirtual4.utn.edu.ec
- eduvirtual5.utn.edu.ec
- eduvirtual6.utn.edu.ec
- eduvirtual7.utn.edu.ec
- eduvirtual8.utn.edu.ec
- eduvirtual8.utn.edu.ec
- h2.utn.edu.ec
- uemprende.utn.edu.ec
- eduvirtual9.utn.edu.ec
- eduvirtual10.utn.edu.ec
- eduvirtual11.utn.edu.ec
- eduvirtual12.utn.edu.ec
- monitoreosrv.utn.edu.ec
- monitoreo.utn.edu.ec
- quipux.utn.edu.ec
- fulltime.utn.edu.ec

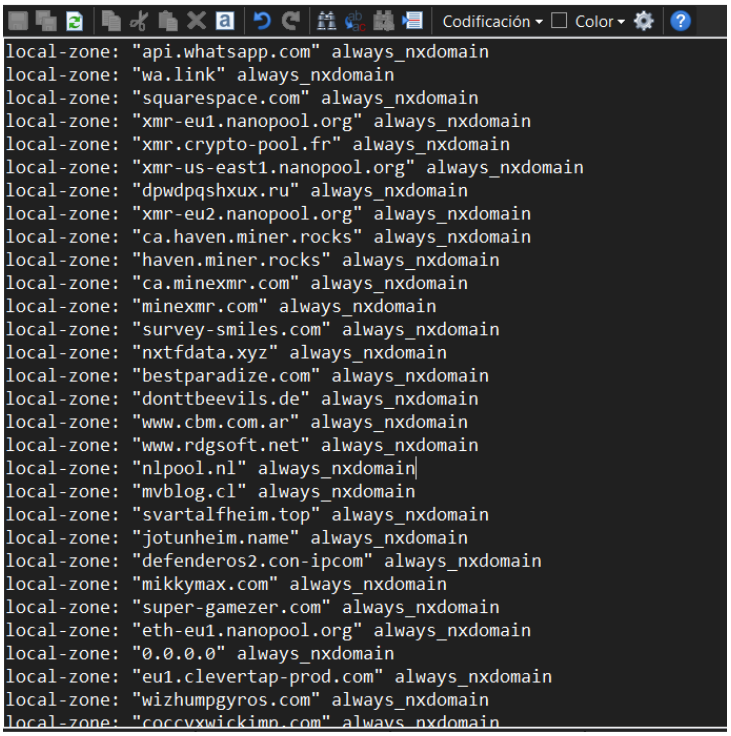
La mayoría de estos dominios son con fines de prueba, cada uno de los cuales se enlazan directamente con su dirección IP correspondiente.

Para finalizar la configuración, se procede a agregar una lista de bloqueo para evitar contagios por enlaces maliciosos, en este caso se descarga la lista de bloqueo de StevenBlack:

```
[root@dnscachel ~]# curl -o hosts https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts ls -lh
```

Comprobamos su existencia en el directorio asignado:

/etc/unbound/local.d/blocklist.conf - henry@10.24.8.8 - Editor - WinSCP



Estas listas por lo general están en constante actualización, por lo que es un paso hacia la seguridad en la red muy bien acentuado.