UTPL
La Universidad Católica de Loja

Vicerrectorado de Modalidad Abierta y a Distancia

Arquitectura de Redes

Guía didáctica

UTPL

Facultad Ingenierías y Arquitectura

Arquitectura de Redes

Guía didáctica

Carrera	PAO Nivel
Tecnologías de la Información	VII

Autor:

Byron Gustavo Jaramillo Campoverde

DRBD_4018

Universidad Técnica Particular de Loja

Arquitectura de redes

Guía didáctica

Byron Gustavo Jaramillo Campoverde

Diagramación y diseño digital

Ediloja Cía. Ltda.
Marcelino Champagnat s/n y París
edilojacialtda@ediloja.com.ec
www.ediloja.com.ec

ISBN digital -978-9942-47-248-9

Año de edición: abril, 2025

Edición: primera edición

El autor de esta obra ha utilizado la inteligencia artificial como una herramienta complementaria. La creatividad, el criterio y la visión del autor se han mantenido intactos a lo largo de todo el proceso.

Loja-Ecuador

Índice

1. Datos de información

1.1 Presentación de la asignatura

1.2 Competencias genéricas de la UTPL

Orientación a la investigación e innovación

1.3 Competencias del perfil profesional

Administrar los servicios de tecnologías de información de la organización utilizando buenas prácticas de la industria asegurando la continuidad operacional del negocio.

1.4 Problemática que aborda la asignatura

La asignatura aborda aspectos fundamentales sobre la gestión, seguridad y automatización de redes de comunicaciones de datos, con énfasis en los protocolos de la capa de aplicación. Se centra en la configuración, análisis y optimización de servicios de red críticos como HTTP, DNS, correo electrónico y protocolos multimedia. Además, se exploran soluciones modernas para la gestión eficiente de infraestructuras tecnológicas, incluyendo redes en la nube

y seguridad en entornos empresariales. A través de metodologías prácticas, los estudiantes aprenderán a seleccionar y aplicar esquemas de redes idóneos según las necesidades de diferentes escenarios tecnológicos.






2. Metodología de aprendizaje

La asignatura se dicta en modalidad en línea y emplea una metodología de aprendizaje activa basada en el análisis de casos prácticos, simulaciones y el uso de herramientas y aplicaciones en la nube. Los estudiantes desarrollarán competencias a través de la resolución de problemas reales y simulados, utilizando herramientas de simulación de acceso libre y plataformas en la nube como AWS, Azure y Google Cloud.

El enfoque metodológico fomenta la aplicabilidad práctica de los contenidos mediante actividades como la configuración de servicios, la simulación de redes empresariales y el análisis de protocolos en entornos virtualizados. Asimismo, se incentiva la participación en foros, tutorías y actividades de discusión asincrónica y sincrónica, donde los estudiantes podrán compartir sus ideas y resolver dudas.

Para reforzar el aprendizaje, se realizarán evaluaciones formativas y proyectos prácticos, promoviendo la reflexión crítica y el desarrollo de soluciones innovadoras en redes de comunicaciones modernas. Esta metodología asegura una formación integral y alineada con las necesidades del entorno profesional actual.

3. Orientaciones didácticas por resultados de aprendizaje

Primer bimestre

Resultado de aprendizaje 1 y 2:

- Discute las arquitecturas típicas de gestión de la red.
- Diseña aplicaciones de red orientada a datos.

Con el fin de alcanzar los resultados de aprendizaje establecidos, las Unidades 1, 2 y 3 abordan los fundamentos esenciales de la arquitectura de redes, con un enfoque especial en la capa de aplicación. En la Unidad 1, se analiza la importancia de esta capa dentro de los modelos de referencia OSI y TCP/IP, además de explorar casos de uso que evidencian cómo las arquitecturas de red facilitan la comunicación eficiente entre dispositivos y aplicaciones.

La Unidad 2 profundiza en los protocolos básicos que operan en esta capa, tales como HTTP, HTTPS, DNS, y DHCP. Se examinan las versiones más recientes de HTTP, así como el rol de los certificados de seguridad en la protección de datos. También se aborda el funcionamiento de los protocolos de correo electrónico (SMTP, IMAP, POP3), proporcionando una base sólida sobre los servicios de red esenciales para la gestión y escalabilidad de infraestructuras tecnológicas.

En la Unidad 3, el enfoque se centra en el desarrollo, monitoreo y análisis de aplicaciones de red. Se revisan conceptos avanzados como la programación de sockets tanto en TCP como en UDP, el diseño de aplicaciones basadas en REST y GraphQL, y el análisis de tráfico con herramientas modernas como Wireshark. Los estudiantes realizarán actividades prácticas que les permitirán capturar, interpretar y monitorear el tráfico de red, fortaleciendo sus habilidades técnicas en la gestión de aplicaciones orientadas a datos.

Cada unidad culminará con autoevaluaciones y actividades prácticas utilizando herramientas de simulación. Estas experiencias serán complementadas con recursos visuales, videos explicativos y encuentros sincrónicos, con el propósito de consolidar los conocimientos adquiridos y resolver dudas.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.

Semana 1

La arquitectura de red constituye la base sobre la cual se diseñan, operan y optimizan los sistemas de comunicación modernos. La capa de aplicación es fundamental para la interacción entre usuarios y servicios de red, permitiendo la transmisión eficiente, segura y escalable de datos. Protocolos esenciales como HTTP, HTTPS, DNS, y DHCP desempeñan un rol clave en la conectividad global, facilitando la comunicación entre servidores, clientes y otros componentes de red. Además, los protocolos de correo electrónico (SMTP, IMAP y POP3) complementan esta infraestructura, garantizando la gestión efectiva de servicios.

El análisis detallado de estas arquitecturas y protocolos es crucial para comprender la organización de los sistemas de red, así como para mejorar la disponibilidad, seguridad y rendimiento. Los modelos de referencia OSI y TCP/IP ofrecen un marco conceptual que permite identificar las funciones específicas de cada capa, lo que es indispensable para diseñar e implementar redes que cumplan con los requisitos actuales de conectividad y gestión de datos.

La Unidad 3 amplía este enfoque mediante el desarrollo y análisis de aplicaciones de red, introduciendo conceptos como la programación de sockets en TCP y UDP, el diseño de APIs con REST y GraphQL, y el monitoreo de tráfico con herramientas especializadas como Wireshark. Estas actividades

no solo reforzarán los conocimientos adquiridos en las Unidades 1 y 2, sino que también le permitirán al estudiante implementar soluciones avanzadas en escenarios prácticos de redes.

Preste especial atención a los contenidos desarrollados en estas tres unidades, ya que proporcionan la base teórica y técnica necesaria para diseñar, analizar y gestionar aplicaciones de red orientadas a datos.

Unidad 1. Introducción a la Capa de Aplicación

Durante esta semana, exploraremos la importancia de la capa de aplicación en las redes de comunicación y su papel en la interacción entre los usuarios y los servicios digitales. Nos adentraremos en su evolución y en cómo se ha convertido en un elemento clave para la conectividad en entornos empresariales y tecnológicos. Descubra cómo esta capa permite el funcionamiento de aplicaciones y servicios esenciales en la red, y conozca los modelos de referencia que han definido su estructura. ¡Iniciemos!

1.1 Importancia de la capa de aplicación

La capa de aplicación es el nivel superior en la arquitectura de redes y es donde se ejecutan las diversas aplicaciones que permiten la comunicación entre los usuarios y los servicios de red. Esta capa define los protocolos y servicios que facilitan la interacción entre dispositivos y software, proporcionando funcionalidades esenciales para el uso eficiente de la red. Dentro de esta capa residen protocolos fundamentales como HTTP, SMTP, FTP, DHCP, DNS, entre otros, los cuales hacen posible la transferencia de información en la Internet, el envío de correos electrónicos, la gestión de archivos, asignación de recursos de direccionamiento y la resolución de nombres de dominio, entre otras funcionalidades.

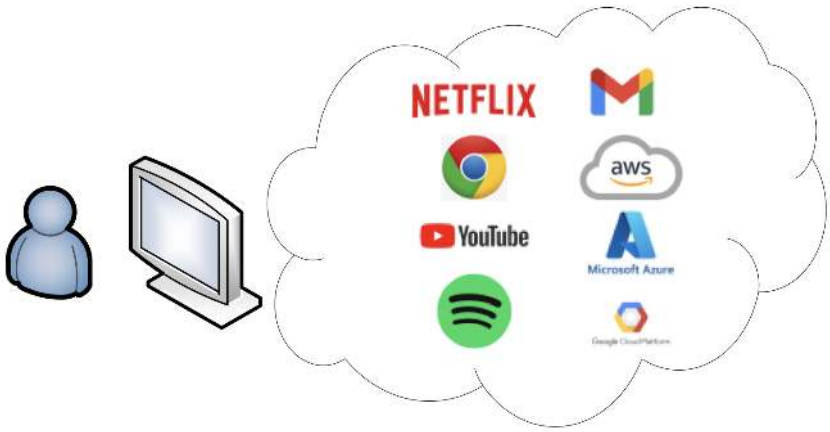
La capa de aplicación es fundamental en la conectividad de redes, ya que, sin ella, la comunicación entre dispositivos carecería de utilidad. Su propósito principal es permitir el intercambio de información a través de diversas

aplicaciones. En la actualidad, interactuamos constantemente con esta capa mediante plataformas que nos permiten ver películas en línea, leer noticias, reproducir videos, escuchar música y, en general, acceder a servicios de comunicación, entretenimiento e información. Gracias a la capa de aplicación, es posible establecer conexiones eficientes y seguras entre usuarios, dispositivos y servicios digitales.

La importancia de la capa de aplicación radica en su capacidad para permitir la interoperabilidad entre diversas plataformas y sistemas, garantizando que las aplicaciones puedan comunicarse de manera estandarizada, independientemente de la infraestructura subyacente. Además, facilita la seguridad en la transmisión de datos mediante protocolos como HTTPS y la autenticación en redes empresariales. Esta capa es fundamental para el funcionamiento de Internet, ya que permite a los usuarios acceder a los recursos y servicios que necesitan tanto en su vida diaria como en el entorno empresarial, optimizando procesos y mejorando la accesibilidad a la información (Kurose & Ross, 2017); la Figura 1 ilustra la cercanía e interacción que existe entre el usuario y algunas aplicaciones populares en la actualidad.

Figura 1

Interacción entre el usuario y la capa de aplicación



Nota. Jaramillo, B., 2025.

Además de las aplicaciones mencionadas en la Figura 1, la capa de aplicación desempeña un papel crucial en la conectividad de redes, ya que en ella operan protocolos esenciales para el intercambio de información y la gestión de recursos. Entre ellos se encuentra el *Dynamic Host Configuration Protocol* (DHCP), el cual permite la asignación dinámica de direcciones IP, facilitando la conectividad automatizada en redes locales. También destaca el *Domain Name System* (DNS), encargado de la resolución de nombres de dominio, permitiendo que los usuarios accedan a sitios web mediante nombres amigables en lugar de direcciones IP numéricas. Asimismo, *Hypertext Transfer Protocol* y *Hypertext Transfer Protocol Secure* (HTTP/HTTPS) son protocolos fundamentales para la comunicación y transferencia segura de datos en la web, garantizando el acceso eficiente a páginas y servicios en línea. A continuación, en la Tabla 1, se presentan algunos de los principales protocolos de la capa de aplicación que son esenciales para el funcionamiento de las redes.

Tabla 1
Protocolos principales de la capa de aplicación

Aplicación	Protocolo de la capa de aplicación	Protocolo de capa de transporte
Correo electrónico	SMTP (RFC 5321)	TCP
Acceso remoto a terminal	SSH (RFC 4253)	TCP
Web	HTTPS (RFC 2818)	TCP
Resolución de nombres de dominio	DNS (RFC 1035)	UDP

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Tabla 1 presenta los principales protocolos que operan en la capa de aplicación, junto con el protocolo correspondiente de la capa de transporte que utilizan para su funcionamiento. Estos protocolos son fundamentales para la comunicación en redes modernas, ya que cada uno cumple una función específica en el intercambio de datos.

¡Excelente!Ahora que ha comprendido la importancia de la capa de aplicación en la comunicación de datos y el papel fundamental que desempeñan sus protocolos, es momento de profundizar en su relación con los modelos de referencia más utilizados en el diseño de redes: OSI y TCP/IP. Estos modelos le permitirán entender de manera estructurada cómo se organizan las distintas funciones de una red, facilitando el análisis y la implementación de soluciones tecnológicas.

1.2 Relación con modelos de referencia (OSI, TCP/IP)

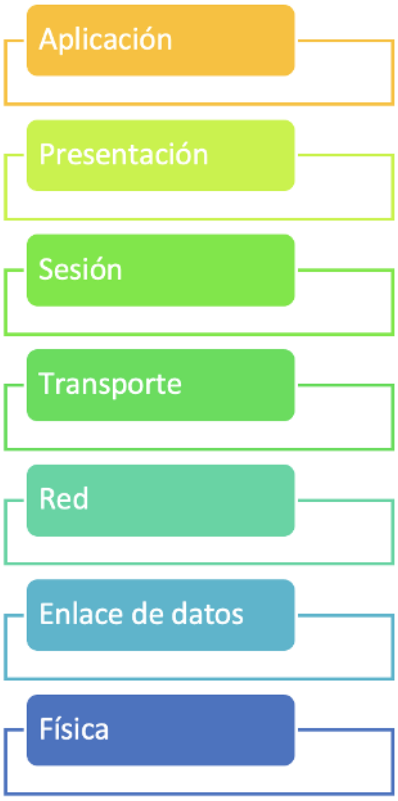
Este apartado se centrará en la forma en que la capa de aplicación se integra en los dos principales modelos de referencia: el modelo OSI *(Open Systems Interconnection)* y el modelo TCP/IP. Estos modelos son esenciales para la comprensión de las arquitecturas de red, ya que segmentan las funciones de comunicación en diferentes capas, permitiendo una mayor organización y control sobre los procesos de intercambio de datos.

Modelo OSI:

El modelo OSI, desarrollado por la Organización Internacional de Normalización (ISO), consta de siete capas. La capa de aplicación es la más alta y se encarga de proporcionar servicios directamente a los usuarios o aplicaciones. En este modelo, la capa de aplicación trabaja junto con las capas de presentación y sesión para gestionar la comunicación, aunque estas dos últimas funciones suelen estar integradas en muchos protocolos modernos. La Figura 2 representa las siete capas del modelo OSI, se puede apreciar a la capa de aplicación como la capa superior.

Figura 2

Capas del modelo OSI



Nota. Jaramillo, B., 2025.

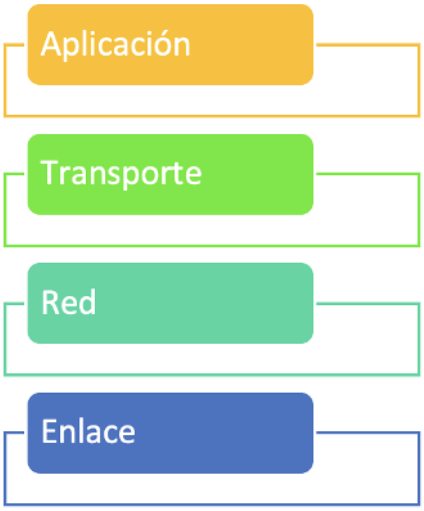
La Figura 2 indica las siete capas del modelo OSI, la importancia de la capa de aplicación radica en que define los protocolos que soportan servicios como el acceso a diferentes servicios (bases de datos, archivos multimedia, etc.), la transferencia de archivos, el correo electrónico, garantizando la interoperabilidad entre diferentes sistemas.

Modelo TCP/IP:

Por otro lado, el modelo TCP/IP, utilizado ampliamente en Internet, organiza las funciones de red en cuatro capas; la Figura 3 representa las capas de este modelo.

Figura 3

Capas del modelo TCP/IP



Nota. Jaramillo, B., 2025.

Como se aprecia en la Figura 3, a diferencia del modelo OSI, TCP/IP combina las responsabilidades de las capas de aplicación, presentación y sesión en una única capa de aplicación. Aquí, los protocolos como HTTP, SMTP, DNS y FTP operan sobre el protocolo de transporte TCP o UDP, asegurando una comunicación eficiente y confiable.

El modelo TCP/IP se caracteriza por su enfoque práctico, lo que ha contribuido a su adopción generalizada en redes comerciales. La capa de aplicación es esencial para conectar aplicaciones distribuidas, permitiendo la transferencia de datos entre servidores y clientes de manera estandarizada (Kurose & Ross, 2017).

1.3 Casos de uso

En este apartado, se presentan ejemplos prácticos que ilustran la importancia de los protocolos de la capa de aplicación en el funcionamiento de servicios digitales. Estos casos muestran situaciones en las que protocolos como DHCP, SMTP, DNS y HTTPS son esenciales para garantizar la conectividad, la comunicación y la seguridad en las redes, iniciemos con la revisión.

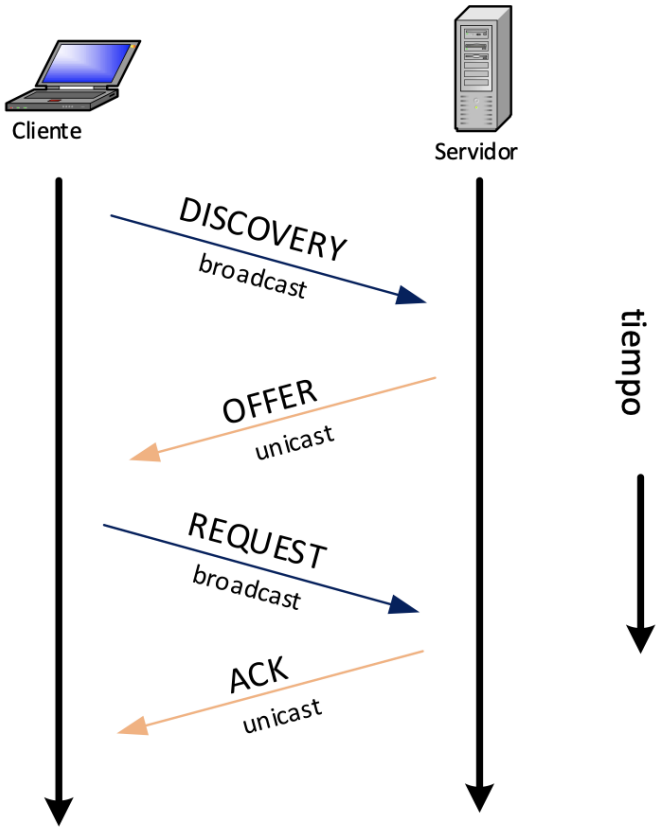
DHCP: Asignación dinámica de direcciones IP

En redes locales, el *Dynamic Host Configuration Protocol (DHCP)* es clave para la asignación automática de direcciones IP a los dispositivos que se conectan. Este mecanismo evita la configuración manual de cada dispositivo y mejora la gestión de los recursos de red.

- **Caso de uso:** Cuando un dispositivo se conecta a una red Wi-Fi, el servidor DHCP le asigna automáticamente una dirección IP. La Figura 4 representa la interacción entre un cliente y un servidor DHCP con el objetivo de obtener los recursos de red necesarios para la conectividad.

Figura 4

Interacción en el servicio DHCP



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

SMTP: Envío de correos electrónicos

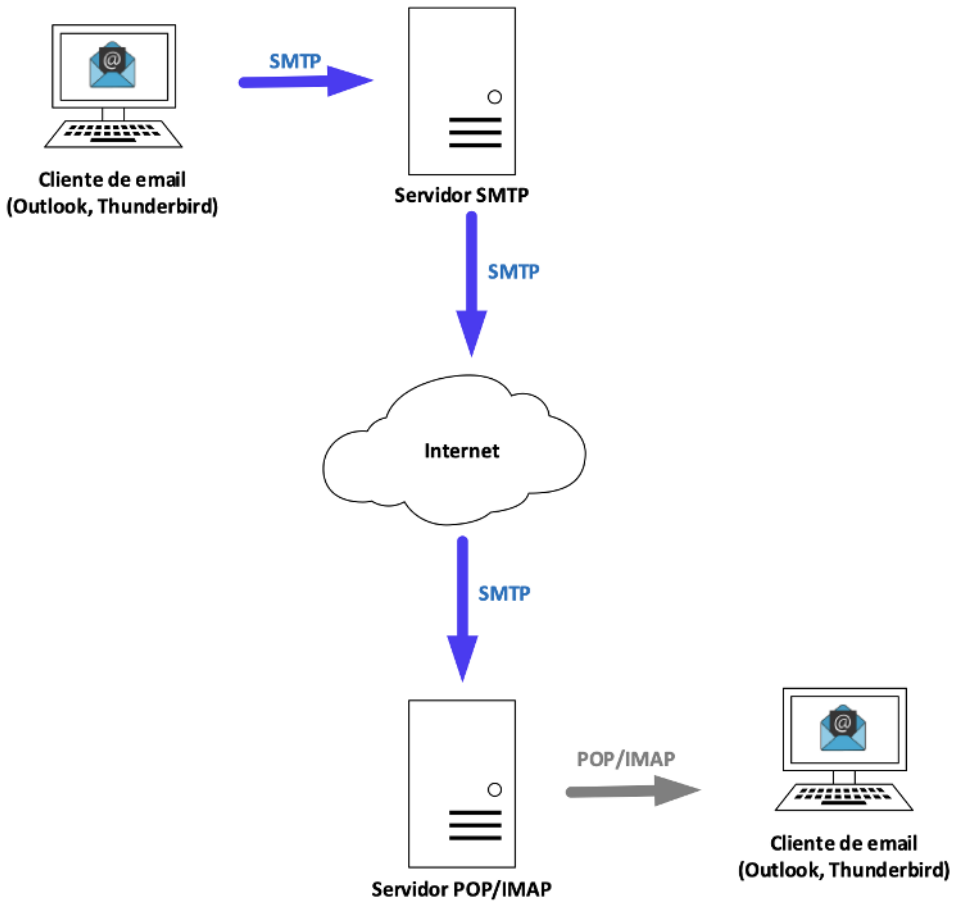
El Simple Mail Transfer Protocol (SMTP) permite la transmisión de correos electrónicos entre servidores. Este protocolo asegura que los mensajes lleguen a su destino de manera confiable, siendo una herramienta indispensable para la comunicación en entornos organizacionales.

- **Caso de uso:** Un servidor de correo envía mensajes electrónicos entre empleados de una empresa, garantizando la entrega a sus respectivos

buzones. La Figura 5 representa el proceso de envío de mensajes utilizando SMTP.

Figura 5

Proceso de envío de emails con SMTP



Nota. Adaptado de *¿Qué es SMTP y para qué sirve?* [Ilustración], por López, 2021, Geeknetic, CC BY 4.0.

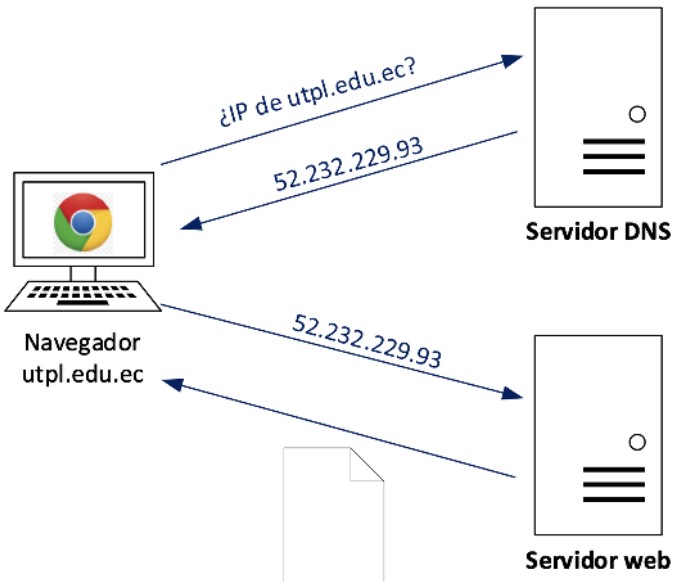
DNS: Resolución de nombres de dominio

El Domain Name System (DNS) convierte nombres de dominio, como utpl.edu.ec, en direcciones IP que los dispositivos utilizan para conectarse a servidores. Este proceso simplifica el acceso a los servicios de red al eliminar la necesidad de recordar direcciones numéricas complejas.

- **Caso de uso:** Un usuario ingresa un nombre de dominio en el navegador, y el sistema DNS resuelve ese nombre en una dirección IP. La Figura 6 presenta el esquema de consulta y respuesta en un servidor DNS.

Figura 6

Proceso de consulta de un dominio mediante DNS



Nota. Jaramillo, B., 2025.

HTTPS: Navegación web segura

El Hypertext Transfer Protocol Secure (HTTPS) protege la comunicación entre clientes y servidores mediante cifrado y autenticación. Esto es crucial para garantizar la privacidad y seguridad de los datos transmitidos en la web, especialmente en transacciones sensibles.

- **Caso de uso:** Al realizar una compra en línea, el protocolo HTTPS asegura que los datos del cliente estén cifrados durante toda la transacción. La Figura 7 muestra el diagrama de interacción de datos seguro utilizando HTTPS.

Figura 7

Navegación segura con HTTPS

Nota. Tomado de *Explain the Working of HTTPS* [Ilustración], por Geeks for Geeks 2024, Geeks for Geeks, CC BY 4.0.

Con estos casos de uso, se resalta la relevancia de los protocolos de la capa de aplicación en el desarrollo y operación de redes modernas. En apartados posteriores, se profundizará en el funcionamiento detallado de cada uno de estos protocolos

Luego de la lectura realizada, es fundamental que identifique y relacione las funciones de la capa de aplicación con los principales protocolos que operan en ella. Le invito a participar en el siguiente juego de arrastrar y soltar, donde podrá asignar cada función de la capa de aplicación (como transferencia de archivos, resolución de nombres de dominio o envío de correo electrónico) a

su protocolo correspondiente y sus acrónimos (FTP, DNS, SMTP, entre otros). Además, le ayudará a afianzar sus conocimientos de manera dinámica y práctica.

Relacionando conceptos de la capa de aplicación

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Examine cómo protocolos como HTTP, DNS, SMTP y DHCP contribuyen al funcionamiento eficiente de aplicaciones y servicios en la red. Analice su uso en sectores como el comercio electrónico, las comunicaciones y la gestión de infraestructuras empresariales. Le sugiero revisar los Protocolos de la Capa de Aplicación para complementar su investigación. Esta actividad le permitirá identificar la relevancia de cada protocolo en la conectividad y seguridad de los servicios de red.
2. Participe activamente en la charla de tutoría semanal, comparta sus ideas, plantee preguntas y discuta los conceptos relacionados con la capa de aplicación y sus protocolos. Reflexione sobre ejemplos prácticos abordados en el estudio y debata posibles aplicaciones en distintos entornos tecnológicos. Esta interacción le ayudará a consolidar sus conocimientos y resolver posibles dudas sobre los temas tratados durante la semana.

¡Felicitaciones por completar la Unidad 1: Introducción a la Capa de Aplicación! Ha dado un paso importante en su comprensión de los conceptos fundamentales de la arquitectura de redes, incluyendo la función de la capa de aplicación, su relevancia en la comunicación de datos y los protocolos esenciales que operan en ella. Ahora

que ha alcanzado este hito, le animo a continuar avanzando en su proceso de aprendizaje, explorando más a fondo los elementos que conforman una infraestructura de red moderna. ¡Siga adelante!

3. Antes de continuar, le recomiendo completar la autoevaluación 1. Esta actividad le permitirá evaluar su comprensión de los conceptos abordados en esta unidad y reforzar sus conocimientos. Dedique tiempo y esfuerzo, ya que le ayudará a consolidar lo aprendido.

   Las preguntas están relacionadas con la Unidad 1: Introducción a la Capa de Aplicación y abordan conceptos clave como su importancia, su relación con los modelos de referencia OSI y TCP/IP, y los principales casos de uso de protocolos como DNS, DHCP, SMTP y HTTPS.

¡Mucho éxito en su camino de aprendizaje!

Autoevaluación 1

Seleccione la opción de respuesta que considere correcta para reforzar y evaluar su comprensión de los contenidos estudiados.

1. ¿Cuál es el objetivo principal de la capa de aplicación en una red?

   A. Realizar la conexión física entre dispositivos.
   B. Facilitar la comunicación entre aplicaciones distribuidas.
   C. Asignar direcciones IP a los dispositivos.

2. ¿Qué modelo de referencia define siete capas funcionales, incluida la capa de aplicación?

   A. Modelo TCP/IP.
   B. Modelo OSI.

C. Modelo de comunicación digital.

3. ¿Cuál de los siguientes protocolos pertenece a la capa de aplicación?

   A. ARP.
   B. HTTPS.
   C. ICMP.

4. ¿Qué función realiza el protocolo DNS en una red?

   A. Cifrar las comunicaciones web.
   B. Traducir nombres de dominio a direcciones IP.
   C. Asignar direcciones IP de manera dinámica.

5. ¿Cuál es la principal ventaja de utilizar el protocolo HTTPS sobre HTTP?

   A. Mejora la velocidad de transmisión.
   B. Proporciona cifrado para proteger los datos transmitidos.
   C. Permite la transmisión de datos multimedia en tiempo real.

6. ¿Cuál es la función del protocolo DHCP en una red?

   A. Administrar la resolución de nombres.
   B. Asignar automáticamente direcciones IP a dispositivos.
   C. Controlar el tráfico de datos multimedia.

7. ¿Cuál de las siguientes opciones representa un servicio común de la capa de aplicación?

   A. Acceso a bases de datos mediante SQL.
   B. Gestión de enrutamiento con OSPF.
   C. Navegación web mediante HTTPS.

8. ¿Qué protocolo es comúnmente utilizado para el envío de correos electrónicos?

   A. DNS.
   B. SMTP.
   C. FTP.

9. ¿Qué proceso describe mejor la función del DNS?

   A. Otorgar acceso remoto a un servidor.
   B. Traducir nombres de dominio en direcciones IP.
   C. Transferir archivos de gran tamaño.

10. ¿Cómo organiza el modelo TCP/IP las funciones de red en comparación con el modelo OSI?

    A. Fusiona las capas de aplicación, presentación y sesión en una sola.
    B. Incluye ocho capas en lugar de siete.
    C. Se enfoca únicamente en el transporte de datos físicos.

Ir al solucionario

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño; caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje. Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Semana 2

Unidad 2. Protocolos básicos de la capa de aplicación

Bienvenido a una nueva etapa de aprendizaje. En esta unidad, se profundizará en los protocolos básicos de la capa de aplicación, esenciales para garantizar la conectividad y el funcionamiento de los servicios en red. Aunque ya se ha mencionado la importancia de estos protocolos, ahora es momento de explorar en detalle cómo operan HTTP/HTTPS, DNS, y DHCP, entre otros. Descubrirá cómo estos protocolos gestionan la transferencia de datos, la seguridad y la resolución de nombres, así como su papel fundamental en la arquitectura de las aplicaciones distribuidas. ¡Adelante, con entusiasmo, para afianzar y ampliar sus conocimientos sobre los pilares de la comunicación en red!

2.1 HTTP descripción y evolución

Este apartado se enfocará en comprender la evolución de HTTP (Hypertext Transfer Protocol), desde su versión inicial HTTP/1.1 hasta las versiones más avanzadas, HTTP/2 y HTTP/3, así como el protocolo HTTPS que garantiza la seguridad en las conexiones web. Se abordarán sus diferencias, mejoras, usos principales y la relevancia que tienen en el rendimiento y la seguridad de los servicios en línea

Descripción de HTTP

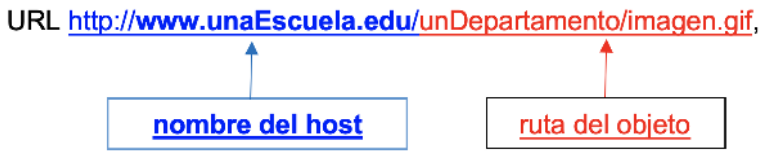
El Protocolo de Transferencia de Hipertexto (HTTP) es el pilar fundamental de la Web y pertenece a la capa de aplicación. Este protocolo, definido en los documentos RFC 1945 y RFC 2616, se implementa mediante dos programas: un cliente y un servidor, los cuales se comunican intercambiando mensajes HTTP. HTTP define tanto la estructura de estos mensajes como el proceso para su intercambio, permitiendo que las aplicaciones web funcionen correctamente (Kurose & Ross, 2017).

Una página web, también conocida como documento web, está compuesta por múltiples objetos. Un objeto es cualquier archivo accesible a través de un URL, como un documento HTML, una imagen JPEG, un applet Java o un video. Generalmente, una página web consta de un archivo base HTML que hace referencia a otros objetos mediante sus respectivas URL. Por ejemplo, si una página incluye texto HTML y cinco imágenes, estará formada por seis objetos: el archivo HTML base y las cinco imágenes.

Cada URL se compone de dos partes: el nombre del host, que identifica al servidor donde se aloja el objeto, y la ruta al archivo; por ejemplo:

Figura 8

Componentes de una URL



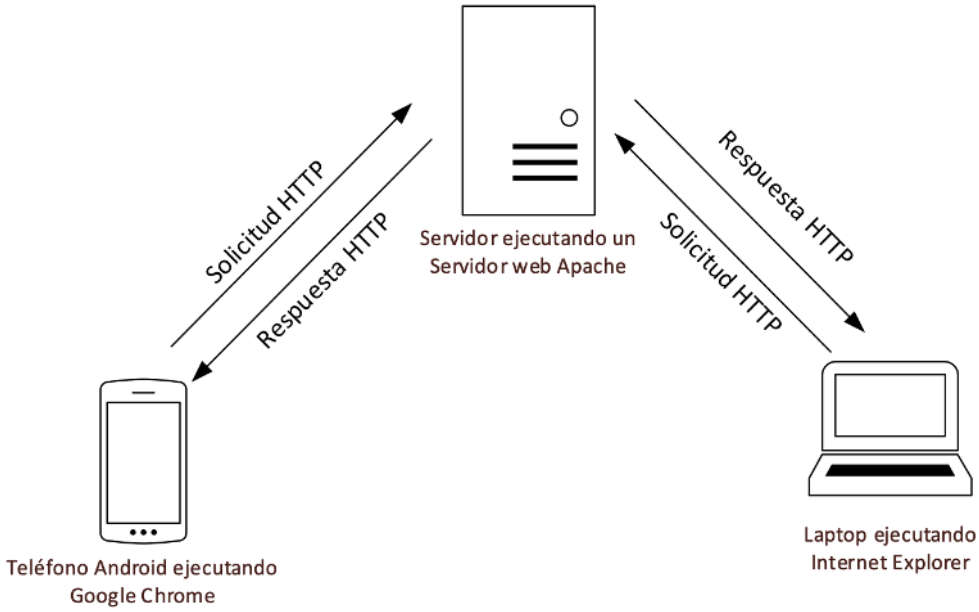
Nota: Adaptado de *Redes de computadoras: Un enfoque descendente* (Kurose, J. F., & Ross, K. W., 2017), Pearson.

Los navegadores web, como Chrome o Firefox, implementan el cliente HTTP, por lo que los términos "navegador" y "cliente" suelen usarse indistintamente. Por otro lado, los servidores web, como Apache o Microsoft Internet Information Server, implementan el servidor HTTP y almacenan los objetos direccionables mediante URL.

Cuando un usuario solicita una página web (por ejemplo, al hacer clic en un enlace), el navegador envía mensajes de solicitud HTTP al servidor, pidiendo los objetos que conforman la página. El servidor responde con mensajes HTTP que contienen los objetos solicitados. Este proceso de interacción entre cliente y servidor es la clave para la transferencia eficiente de páginas web en Internet. La Figura 9 representa la interacción entre dos clientes que ejecutan un navegador y su conexión con un servidor web.

Figura 9

Conexión mediante HTTP



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Figura 9 muestra el proceso de comunicación mediante el protocolo HTTP entre clientes y un servidor web. En el diagrama se ilustran dos dispositivos clientes interactuando con un servidor web que ejecuta Apache mediante dos procesos clave:

- **Solicitud HTTP:** Cada cliente envía una solicitud HTTP al servidor para acceder a un recurso web, como una página o archivo.
- **Respuesta HTTP:** El servidor procesa la solicitud y devuelve una respuesta HTTP que contiene el recurso solicitado, como un documento HTML o una imagen.

Este proceso refleja la arquitectura cliente-servidor, donde los navegadores actúan como clientes y los servidores web gestionan las solicitudes, enviando los objetos necesarios para la visualización de una página web.

Formato de mensaje de HTTP

Las especificaciones de HTTP (documentadas en los RFC 1945, RFC 2616 y RFC 7540) definen el formato de los mensajes HTTP, que se dividen en dos tipos principales: mensajes de solicitud y mensajes de respuesta.

- Mensaje de solicitud HTTP**;** un mensaje de solicitud HTTP típico puede tener el siguiente formato.

  GET /unadireccion/pagina.html HTTP/1.1

  Host: www.unaEscuela.edu

  Connection: close

  User-agent: Mozilla/5.0

  Accept-language: fr

  Este mensaje tiene varias características importantes. En primer lugar, está escrito en texto ASCII, lo que facilita su lectura e interpretación por personas con conocimientos técnicos. En segundo lugar, el mensaje se compone de múltiples líneas, cada una seguida de un retorno de carro (*carriage return*) y un salto de línea. La estructura puede variar dependiendo del contenido de la solicitud, con mensajes que pueden contener solo una línea o muchas más.

  A continuación, la Tabla 2 indica el propósito y significado de cada línea del mensaje de solicitud HTTP.

Tabla 2

Mensaje de solicitud mediante HTTP

Línea	Descripción	Ejemplo
Línea de solicitud	Contiene el método HTTP, la URL solicitada y la versión del protocolo HTTP (Los métodos más comunes son **GET**, **POST**, **HEAD**, **PUT** y **DELETE)**	GET /unadireccion/ pagina.html HTTP/1.1
Host	Especifica el nombre del servidor donde reside el recurso solicitado. Es necesario para que los proxies web identifiquen correctamente el servidor de destino.	Host: www.unaEscuela.edu
Connection	Define si el cliente desea mantener o cerrar la conexión después de la respuesta	Connection: close
User-agent	Identifica el navegador o cliente que realiza la solicitud. Permite al servidor personalizar el contenido según el cliente.	User-agent: Mozilla/5.0
Accept-language	Indica el idioma preferido por el cliente para el contenido solicitado. Si el servidor dispone del objeto en ese idioma, lo enviará.	Accept-language: fr

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

- **Mensaje de respuesta HTTP**

Un mensaje de respuesta HTTP es enviado por el servidor al cliente en respuesta a una solicitud. Este mensaje contiene información sobre el resultado de la solicitud, el estado de la respuesta y el contenido solicitado (si está disponible). Similar al mensaje de solicitud, el mensaje de respuesta se compone de múltiples líneas estructuradas en diferentes secciones.

```
HTTP/1.1 200 OK
```

Date: Mon, 01 Feb 2025 10:00:00 GMT

Server: Apache/2.4.41 (Ubuntu)

Content-Type: text/html; charset=UTF-8

Content-Length: 438

<html>

<head><title>Página de ejemplo</title></head>

<body><h1>Bienvenido</h1></body>

</html>

A continuación, la Tabla 3 indica el propósito y significado de cada línea del mensaje de respuesta HTTP.

Tabla 3

Mensaje de respuesta mediante HTTP

Línea	Descripción	Ejemplo
Línea de estado	Indica la versión del protocolo HTTP, el código de estado de la respuesta y una breve descripción del estado.	HTTP/1.1 200 OK
Date	Especifica la fecha y hora en que el servidor generó la respuesta.	Date: Mon, 01 Feb 2025 10:00:00 GMT
Server	Identifica el software del servidor que procesó la solicitud.	Server: Apache/2.4.41 (Ubuntu)
Content-Type	Define el tipo de contenido que se envía en el cuerpo de la respuesta (HTML, texto, imagen, etc.).	Content-Type: text/html; charset=UTF-8
Content-Length	Indica el tamaño del contenido en bytes.	Content-Length: 438
Cuerpo del mensaje	Contiene el recurso solicitado, como un documento HTML, imagen u otro tipo de archivo.	<html><head><title>Página de ejemplo... </title></head></html>

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Tabla 3 resume la estructura del mensaje de respuesta HTTP, explicando cómo cada línea contribuye a la entrega del recurso solicitado por el cliente.

Cuando un cliente envía una solicitud HTTP (por ejemplo, para acceder a una página web), el servidor genera un mensaje de respuesta como el mostrado anteriormente. Si la solicitud es exitosa, el servidor devuelve el

contenido solicitado junto con el código de estado 200 OK. Si hay un error, el servidor puede devolver un mensaje de error acompañado de un código de estado como 404 Not Found o 500 Internal Server Error.

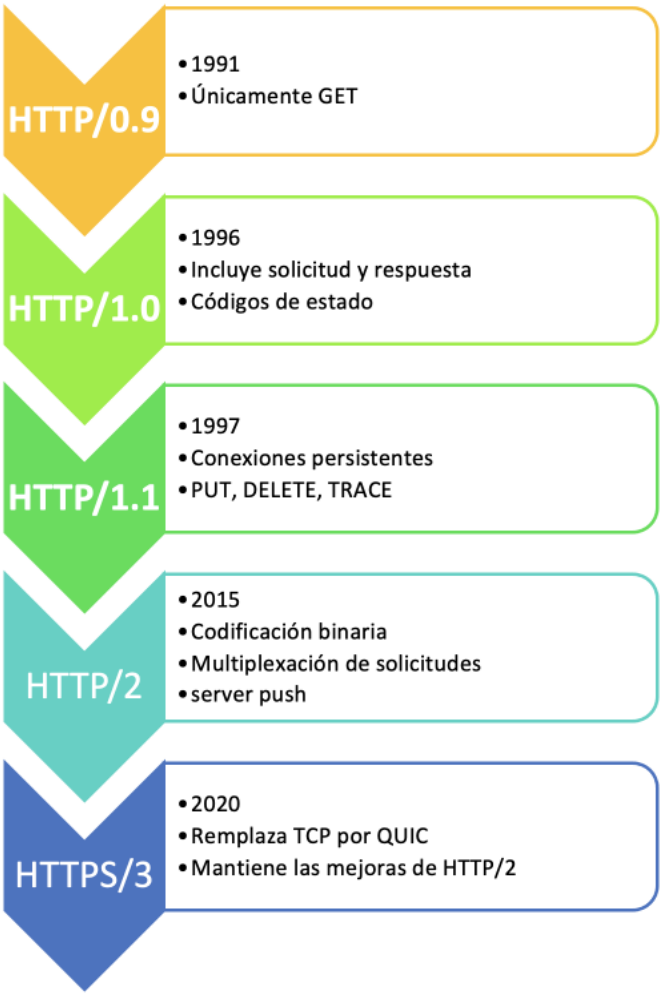
Esta estructura permite a los navegadores interpretar correctamente la respuesta del servidor, renderizar el contenido y mostrarlo al usuario.

Evolución de HTTP

El Protocolo de Transferencia de Hipertexto (HTTP) es la base de la comunicación en la World Wide Web, permitiendo la transferencia de información entre clientes y servidores. Desde su creación, HTTP ha experimentado varias actualizaciones significativas para mejorar su eficiencia, seguridad y rendimiento. A continuación, la Figura 10 representa los hitos clave en la evolución del protocolo HTTP.

Figura 10

Evolución de HTTP



Nota. Jaramillo, B., 2025.

HTTP/0.9: La primera versión

Creada en 1991, esta versión de HTTP era extremadamente básica. Solo soportaba el método GET, lo que permitía transferir documentos HTML sin cabeceras ni metadatos. La simplicidad de HTTP/0.9 limitaba sus funcionalidades, pero fue suficiente para el inicio de la Web (Kurose & Ross, 2017).

HTTP/1.0: Fundamentos del protocolo

Lanzado en 1996, HTTP/1.0 introdujo métodos adicionales como POST y HEAD, así como las cabeceras que permitían la transmisión de metadatos. Sin embargo, esta versión requería una conexión nueva para cada solicitud, lo que generaba sobrecarga y tiempos de respuesta lentos. La falta de conexiones persistentes era un obstáculo importante para la eficiencia en el manejo de múltiples recursos web (*RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1*, s. f.).

HTTP/1.1: Mejoras en rendimiento

En 1997, HTTP/1.1 trajo conexiones persistentes, lo que permitió enviar varias solicitudes a través de una misma conexión TCP. Además, se introdujo la canalización, donde los clientes podrían enviar múltiples solicitudes sin esperar la respuesta de cada una. Esto optimizó el rendimiento, mejorando la experiencia del usuario al cargar páginas web con varios recursos (Kurose & Ross, 2017).

HTTP/1.1 también añadió el encabezado Host, lo que posibilitó que varios sitios web compartieran una misma dirección IP. A pesar de estas mejoras, la necesidad de procesar solicitudes en serie genera cuellos de botella en entornos con alta demanda (*RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2)*, s. f.)

HTTP/2: Innovación en la comunicación web

Publicado en 2015, HTTP/2 implementó multiplexación, compresión de cabeceras y el empuje de servidor (server push). Estas características permitieron enviar múltiples solicitudes y respuestas de manera simultánea a través de una única conexión TCP, lo que redujo la latencia significativamente. Además, el servidor podía enviar recursos críticos al cliente sin que fueran solicitados previamente, mejorando los tiempos de carga de páginas (Belshe et al., 2015).

La adopción de HTTP/2 fue rápida debido a sus beneficios de rendimiento, con soporte en navegadores y servidores líderes en la industria.

HTTP/3: Un nuevo enfoque basado en QUIC

Lanzado en 2020, HTTP/3 se basa en el protocolo QUIC, que utiliza UDP en lugar de TCP. Esto permite un establecimiento de conexiones más rápido, ideal para redes inestables. HTTP/3 incluye cifrado por defecto, mejorando la seguridad y protección de datos. Con estas características, HTTP/3 reduce el impacto de la pérdida de paquetes y mejora la experiencia en aplicaciones que requieren alta interactividad, como el streaming y las videollamadas (Ranasinghe, 2023)

Impacto en la experiencia del usuario

La evolución del protocolo HTTP ha mejorado notablemente la eficiencia de la comunicación web. Mientras HTTP/1.0 presentaba problemas de rendimiento al manejar múltiples solicitudes, HTTP/2 y HTTP/3 optimizaron la transferencia de datos. Estas mejoras se traducen en tiempos de carga más cortos, menor latencia y una experiencia de navegación más ágil, especialmente en dispositivos móviles y redes de baja calidad.

2.2 HTTP y HTTPS: Seguridad en la Comunicación Web

HTTP transmite datos sin cifrar, lo que hace que la información enviada entre cliente y servidor sea vulnerable a interceptaciones por parte de terceros. Para solucionar este problema, se creó HTTPS, que agrega una capa de seguridad al combinar las solicitudes y respuestas HTTP con los protocolos de cifrado SSL (Secure Sockets Layer) y TLS (Transport Layer Security).

Un sitio web que utiliza HTTPS debe obtener un certificado SSL/TLS de una autoridad certificadora (CA). Este certificado verifica la autenticidad del sitio y contiene información criptográfica que permite establecer una conexión segura entre el navegador y el servidor. El proceso de comunicación segura inicia cuando el navegador solicita el certificado al servidor. El servidor envía el certificado, que incluye una clave pública. Una vez verificada la autenticidad del sitio, el navegador utiliza esta clave pública para cifrar una clave de sesión secreta, la cual es descifrada por el servidor mediante su clave privada. Después de este intercambio, ambos utilizan la clave de sesión para continuar la comunicación de forma segura (*HTTP y HTTPS: Diferencia Entre los Protocolos de Transferencia. AWS*, s. f.).

Certificado SSL/TLS

Un certificado SSL/TLS es un documento digital que permite a los sistemas establecer conexiones seguras mediante protocolos criptográficos. Estos certificados, emitidos por autoridades certificadoras (CA), funcionan como identificaciones digitales que aseguran la comunicación en Internet y redes privadas. El proceso de validación del certificado establece confianza entre el servidor y el navegador, permitiendo el intercambio seguro de datos mediante claves públicas y privadas.

Los certificados SSL/TLS ofrecen varios beneficios, entre ellos:

- Protección de datos privados
- Cumplimiento normativo

- Mejora en la confianza de los clientes (visualizada con el ícono de candado en la barra de direcciones) y,
- Mejor posicionamiento en *Search Engine Optimization* (SEO). Los navegadores verifican estos certificados para asegurar que las conexiones sean confiables y seguras.

Los principios clave de esta tecnología incluyen el cifrado, que garantiza que solo el destinatario pueda descifrar los datos, y la autenticación, que permite a los navegadores verificar la identidad de los servidores.

Los certificados SSL/TLS tienen un periodo de validez máximo de 13 meses para minimizar los riesgos de seguridad asociados con certificados vencidos. Cuando caduca un certificado, los navegadores advierten a los usuarios sobre la falta de seguridad del sitio web. Los certificados modernos utilizan el protocolo TLS como estándar, aunque el término SSL/TLS sigue siendo común en el ámbito de la seguridad digital.

Relación entre el certificado SSL/TLS y firmas digitales

Los certificados SSL/TLS contienen una firma digital generada por la autoridad certificadora. Esta firma garantiza que el certificado es legítimo y no ha sido alterado. Durante el proceso de comunicación segura, el navegador verifica esta firma para autenticar el certificado SSL/TLS y establecer una conexión confiable.

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Elabore un esquema comparativo que analice las diferencias entre los protocolos HTTP y HTTPS. En este esquema, examine aspectos clave como el cifrado, la seguridad, los certificados SSL/TLS y los métodos de verificación, evaluando su impacto en la protección de datos en la web. Esta actividad

le ayudará a reforzar los conceptos esenciales sobre la comunicación segura en redes.

Nota: Por favor complete la actividad en un cuaderno o documento Word

2. Realice el siguiente juego de arrastrar y soltar, donde podrá validar su comprensión sobre el propósito, funcionamiento y relación entre estos elementos de seguridad. Este juego le permitirá consolidar su conocimiento acerca de cómo se autentican las conexiones seguras en la web.
Diferencias entre firmas digitales y certificados SSL/TLS

Estas actividades le permitirán analizar, sintetizar y aplicar los conceptos tratados durante la semana, asegurando una comprensión más profunda del tema.

Ha finalizado una nueva semana de aprendizaje, es gratificante ver cómo se ha enfrentado a nuevos conceptos y ha aprovechado al máximo esta oportunidad de crecimiento. Estoy seguro de que, ha adquirido valiosos conocimientos que le serán útiles en su camino académico o profesional.

En esta etapa de su aprendizaje, sugiero que se tome un momento para reflexionar sobre el progreso que ha hecho, debe estar orgulloso de sus avances y su determinación para continuar; ¡ánimo, siga adelante!

Semana 3

Unidad 2. Protocolos básicos de la capa de aplicación

¡Es hora de comenzar una nueva semana de estudio, llena de oportunidades y descubrimientos! En estos días, explorará conceptos fundamentales que juegan un papel crucial en la conectividad de redes: DNS, DHCP y los usos comunes de estos protocolos en la arquitectura de red. Le animo a enfocarse en los desafíos y conocimientos que le esperan. Aproveche al máximo los recursos a su disposición, organice su tiempo de manera efectiva y mantenga una mentalidad positiva.

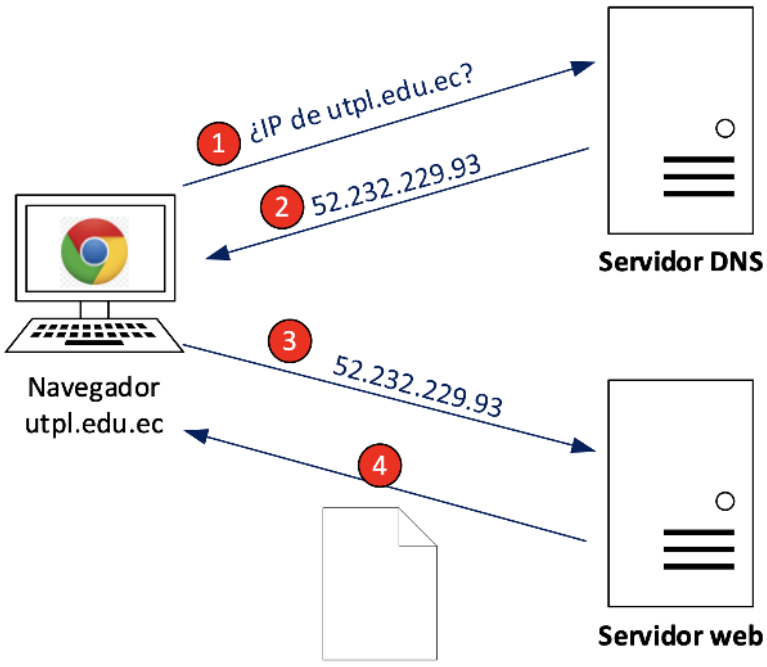
2.3 Sistema de nombres de dominio (DNS)

Para que los dispositivos en Internet se identifiquen y se comuniquen entre sí, es necesario utilizar identificadores específicos. Aunque los nombres de host son fáciles de recordar para las personas, no son prácticos para los routers y sistemas de red. Por ello, se emplean las direcciones IP, que ofrecen información técnica sobre la ubicación de un dispositivo en la red.

El Sistema de Nombres de Dominio (DNS) se encarga de traducir los nombres de host en direcciones IP, permitiendo una comunicación eficiente en Internet. Este sistema es esencial para que los usuarios puedan acceder a sitios web y otros recursos sin necesidad de memorizar complejas direcciones numéricas (Kurose & Ross, 2017). La Figura 11 representa el funcionamiento de una consulta de un nombre de dominio utilizando DNS en un nivel general.

Figura 11

Consulta de un nombre de dominio con DNS – alto nivel



Nota. Jaramillo, B., 2025.

La Figura 11 ilustra el funcionamiento básico del Sistema de Nombres de Dominio (DNS) durante una solicitud web. A continuación, se detalla cada paso del proceso:

- **Solicitud al servidor DNS:** El navegador web, en este caso utilizando la dirección *utpl.edu.ec*, envía una solicitud al servidor DNS para obtener la dirección IP correspondiente a ese nombre de dominio.
- **Respuesta del servidor DNS:** El servidor DNS responde proporcionando la dirección IP asociada al nombre de dominio solicitado, en este caso *52.232.229.93*.
- **Solicitud al servidor web:** Con la dirección IP ya obtenida, el navegador envía una solicitud al servidor web ubicado en *52.232.229.93*. Esta solicitud contiene la información necesaria para acceder al contenido del sitio web.

- **Respuesta del servidor web:** El servidor web procesa la solicitud y envía el contenido de la página solicitada de vuelta al navegador, completando así el ciclo de comunicación.

Este proceso permite a los usuarios acceder a recursos en la web sin necesidad de recordar complejas direcciones IP, garantizando una experiencia fluida y eficiente.

Jerarquía de servidores DNS

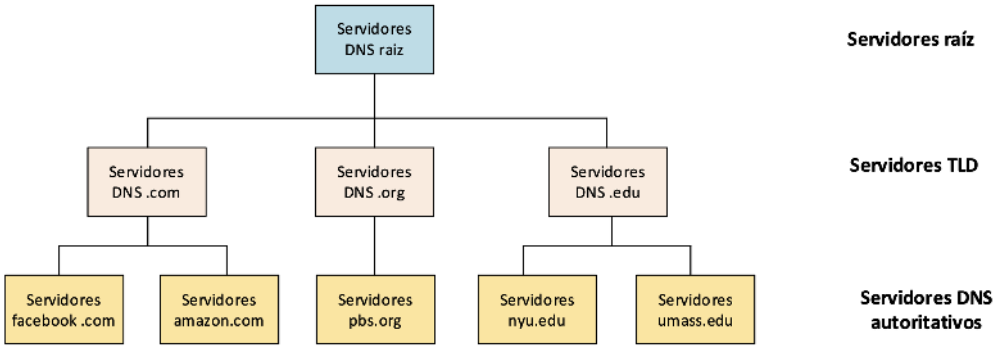
El Sistema de Nombres de Dominio (DNS) se basa en una estructura jerárquica y distribuida de servidores para gestionar eficientemente las consultas relacionadas con la resolución de nombres. Para manejar el gran volumen de solicitudes a nivel global, se utilizan tres tipos principales de servidores DNS: servidores raíz, servidores de dominio de nivel superior (TLD) y servidores autoritativos. A continuación, se describe cómo interactúan entre sí para resolver una consulta típica:

- Los servidores DNS raíz, son los puntos más altos de la jerarquía. Actualmente, existen alrededor de 400 servidores raíz distribuidos a nivel mundial, administrados por diversas organizaciones. Estos servidores actúan como el primer contacto en una consulta DNS, proporcionando la dirección de los servidores TLD correspondientes al dominio solicitado. Por ejemplo, al buscar *www.amazon.com*, el servidor raíz dirige la consulta hacia los servidores TLD responsables del dominio *.com*.
- Servidores TLD (Top-Level Domain), estos servidores gestionan los dominios de nivel superior, como *com*, *org*, *edu*, o aquellos específicos de países (*uk*, *fr*, *ec*). Una vez recibida una consulta, los servidores TLD devuelven la dirección IP del servidor DNS autoritativo encargado del dominio específico. Por ejemplo, un servidor TLD para el dominio *com* indica cuál es el servidor autoritativo para *amazon.com*.
- Servidores DNS autoritativos, las organizaciones con servicios accesibles en Internet mantienen registros DNS en estos servidores. Un servidor autoritativo contiene las direcciones IP asociadas a los nombres de host de una organización. Puede ser gestionado directamente por la organización o

por un proveedor externo. Por ejemplo, un servidor autoritativo de Amazon almacenaría la dirección IP de *www.amazon.com*.

La Figura 12 representa la jerarquía de los servidores DNS.

Figura 12
Jerarquía del servicio DNS



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

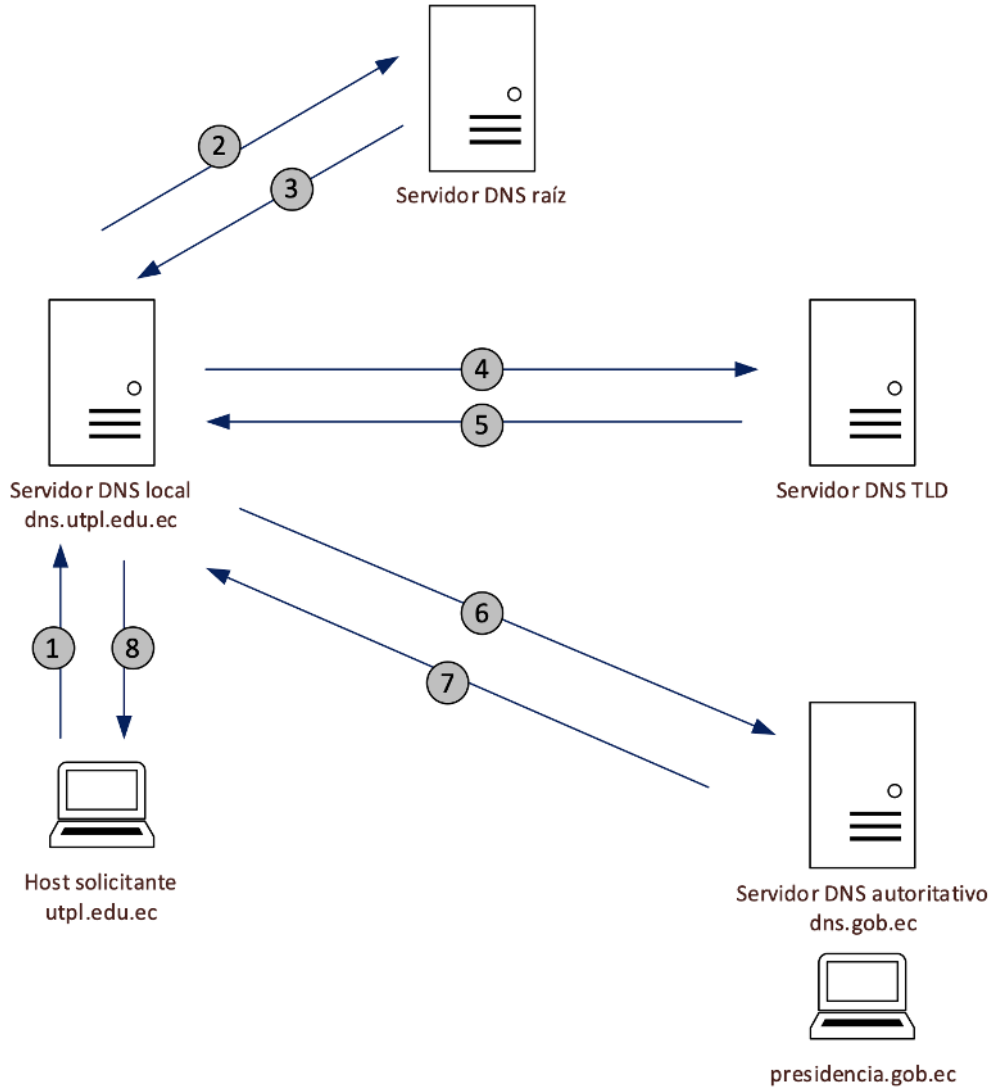
Además de estos tres tipos de servidores, también existe un servidor DNS local, que no forma parte de la jerarquía principal, pero juega un papel fundamental en la arquitectura DNS. Este servidor local es proporcionado por el proveedor de servicios de Internet (ISP) o por una red institucional. Actúa como intermediario entre el cliente y los servidores DNS jerárquicos, enviando consultas y almacenando temporalmente las respuestas en caché para mejorar el rendimiento en consultas futuras.

Ejemplo de consulta DNS detallada

Para ilustrar el proceso, supongamos que el host cse.nyu.edu desea resolver el nombre de host gaia.cs.umass.edu. El flujo de consulta se presenta en la Figura 13.

Figura 13

Consulta detallada de un dominio utilizando DNS



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Figura 13 representa una consulta DNS detallada, que, paso a paso describe:

- **Solicitud del host al servidor DNS local**, el host solicitante (*utpl.edu.ec*) inicia una consulta DNS, enviando un mensaje al servidor DNS local (*dns.utpl.edu.ec*), solicitando la dirección IP del nombre de host *presidencia.gob.ec*.
- **Consulta del servidor DNS local al servidor DNS raíz**, el servidor DNS local no tiene la información solicitada en su caché, por lo que reenvía la consulta a un servidor DNS raíz. Este servidor raíz es responsable de identificar la ubicación de los servidores DNS TLD que gestionan los dominios de nivel superior (como *.ec*).
- **Respuesta del servidor raíz,** el servidor DNS raíz responde al servidor local proporcionando una lista de direcciones IP de los servidores TLD que administran el dominio *.ec*.
- **Consulta al servidor TLD,** con la información proporcionada por el servidor raíz, el servidor DNS local envía una nueva consulta a uno de los servidores TLD del dominio *.ec*. Esta consulta solicita la dirección IP del servidor DNS autoritativo correspondiente al dominio específico (*gob.ec*).
- **Respuesta del servidor TLD**, el servidor TLD responde al servidor DNS local, enviando la dirección IP del servidor DNS autoritativo que tiene los registros de *presidencia.gob.ec*.
- **Consulta al servidor autoritativo,** el servidor DNS local se comunica con el servidor autoritativo (*dns.gob.ec*), enviando una consulta para obtener la dirección IP del host solicitado.
- **Respuesta del servidor autoritativo**, el servidor autoritativo responde al servidor DNS local, proporcionando la dirección IP correspondiente a *presidencia.gob.ec*.
- **Respuesta al host**, finalmente, el servidor DNS local reenvía la respuesta al host solicitante. El host ahora puede utilizar la dirección IP obtenida para establecer una conexión con *presidencia.gob.ec*.

Este proceso permite una resolución jerárquica y eficiente de nombres de dominio, facilitando la conectividad entre diferentes sistemas en Internet mediante la colaboración de servidores distribuidos globalmente.

2.4 Protocolo de Configuración Dinámica de Host (DHCP)

El Protocolo de Configuración Dinámica de Host (DHCP, por sus siglas en inglés) es un mecanismo crucial en la gestión de redes, diseñado para asignar automáticamente direcciones IP y otros parámetros de configuración a los dispositivos conectados. Sin DHCP, cada dispositivo requeriría una configuración manual de red, lo que resultaría en un proceso tedioso, especialmente en redes de gran tamaño.

Con DHCP, cuando un dispositivo se conecta a la red, el servidor DHCP se encarga de asignar de manera dinámica una dirección IP disponible, junto con otros datos necesarios como la puerta de enlace predeterminada, la máscara de subred y los servidores DNS. Esto facilita el mantenimiento de la red y garantiza que no haya conflictos de direcciones IP, mejorando así la eficiencia y escalabilidad del entorno de red. En este apartado, exploraremos el funcionamiento de DHCP, sus mensajes clave y el flujo de trabajo que permite una configuración automática eficaz.

El Protocolo de Configuración Dinámica de Host (DHCP) es un protocolo cliente-servidor que facilita la asignación automática de direcciones IP y otros parámetros de configuración de red a dispositivos en una red. Opera sobre el Protocolo de Datagrama de Usuario (UDP), utilizando el puerto 67 para el servidor y el puerto 68 para el cliente. Esta automatización simplifica la gestión de redes, especialmente en entornos con numerosos dispositivos o con hosts móviles que se conectan y desconectan con frecuencia.

DHCP ofrece tres mecanismos principales de asignación de direcciones:

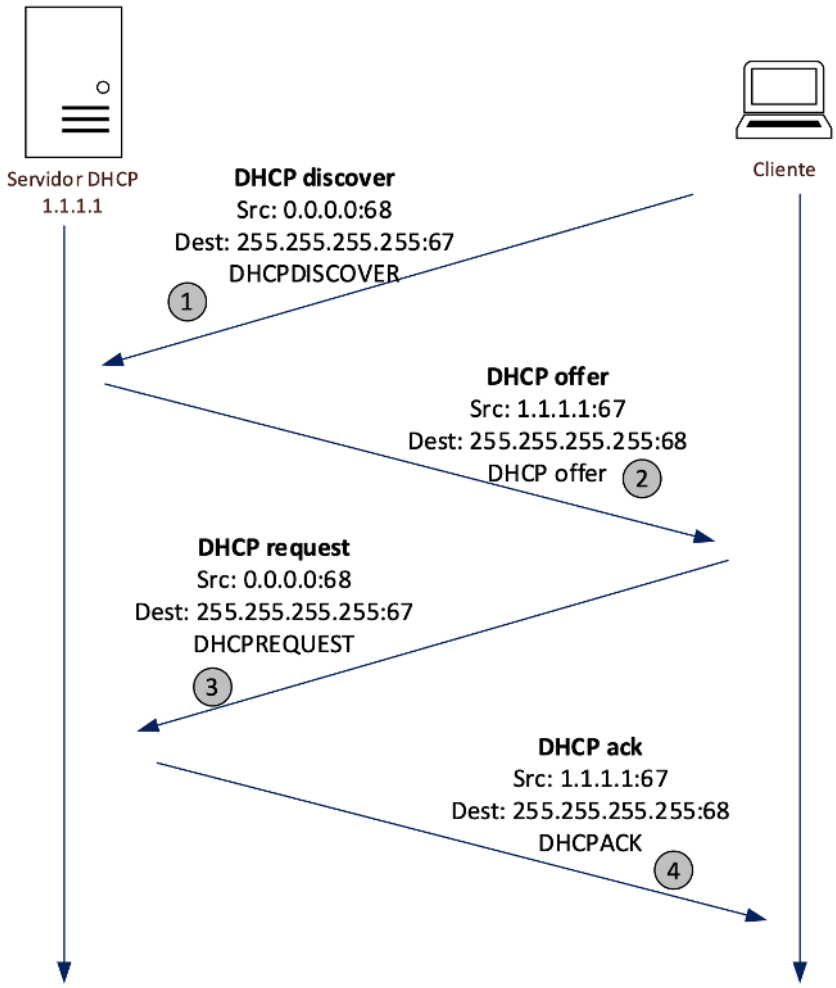
1. **Asignación Manual:** El administrador asigna una dirección IP específica a un cliente, y DHCP se encarga de comunicar dicha dirección al dispositivo.

2. **Asignación Automática:** DHCP asigna permanentemente una dirección IP estática a un dispositivo, seleccionándola de un conjunto de direcciones disponibles.
3. **Asignación Dinámica:** DHCP asigna dinámicamente una dirección IP de un pool de direcciones por un período de tiempo limitado o hasta que el cliente ya no necesite la dirección.

El proceso de concesión de una dirección IP mediante DHCP implica una serie de pasos entre el cliente y el servidor, la Figura 14 describe el flujo.

Figura 14

Proceso de asignación de direcciones IP con DHCP



Nota. Adaptado de *Dynamic Host Configuration Protocol (DHCP)* [Ilustración], por w3.ual, s. f., w3.ual, CC BY 4.0.

En la Figura 14 se puede apreciar el proceso de asignación de direcciones IP mediante DHCP, el cual se describe a continuación.

1. **Descubrimiento (DHCP Discover):** El cliente envía un mensaje de broadcast para localizar servidores DHCP disponibles.

2. **Oferta (DHCP Offer):** Los servidores DHCP responden con una oferta que incluye una dirección IP y otros parámetros de configuración.
3. **Solicitud (DHCP Request):** El cliente selecciona una oferta y responde con una solicitud para confirmar la aceptación de la dirección IP ofrecida.
4. **Reconocimiento (DHCP Ack):** El servidor confirma la concesión de la dirección IP y proporciona los parámetros de configuración adicionales necesarios.

Este proceso permite que los dispositivos se configuren automáticamente al unirse a una red, reduciendo la necesidad de configuraciones manuales y minimizando errores. Además, DHCP utiliza un mecanismo de arrendamiento que asegura que las direcciones IP no utilizadas se devuelvan al pool para su reasignación, optimizando así el uso de direcciones en la red.

Es importante destacar que DHCP es una evolución del Protocolo de Arranque (BOOTP). Aunque ambos son protocolos cliente-servidor que utilizan los mismos puertos UDP, DHCP introduce mejoras significativas, como la asignación dinámica de direcciones y la capacidad de recuperar y reasignar direcciones IP mediante un mecanismo de arrendamiento ((*Dynamic Host Configuration Protocol (DHCP)*, s. f.).

¡Felicidades! Ha finalizado exitosamente el estudio de la semana 3. Ahora es momento de avanzar a la semana 4, donde explorará en profundidad el proceso de envío y recepción de correos electrónicos. Además, pondrá en práctica los conocimientos adquiridos mediante simulaciones para afianzar su aprendizaje.

Pero antes de continuar, le invito a participar en algunas actividades diseñadas para reforzar los conceptos clave.

Actividades de aprendizaje recomendadas

1. Investigue y elabore un cuadro comparativo que incluya las funciones, características, beneficios y posibles limitaciones

del Protocolo DNS y el Protocolo DHCP en entornos empresariales. Incluya al menos un ejemplo práctico del uso de cada protocolo en una organización.
2. Analice un caso donde una empresa deba implementar ambos protocolos en una red corporativa. Proponga una solución que incluya el diseño de una arquitectura de red con servidores DNS y DHCP, justificando su importancia en el manejo de recursos y conectividad.

Nota: Por favor complete las actividades en un cuaderno o documento Word

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 4

Unidad 2. Protocolos básicos de la capa de aplicación

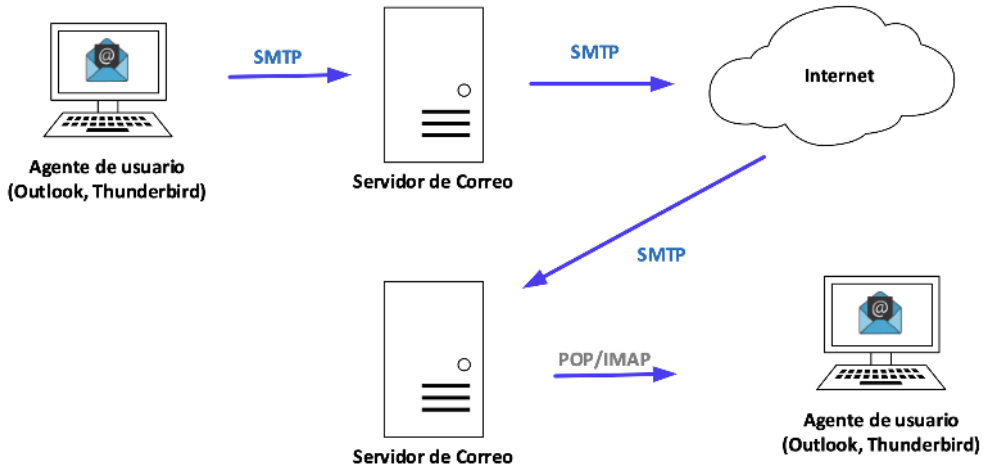
2.5 Protocolo de correo electrónico - SMTP, IMAP y POP3.

El correo electrónico, una de las aplicaciones más antiguas y fundamentales de Internet, ha evolucionado significativamente desde sus inicios, convirtiéndose en un medio esencial para la comunicación. Al igual que el correo postal tradicional, el correo electrónico es asíncrono, permitiendo a las personas enviar y leer mensajes cuando lo deseen, sin necesidad de una coordinación temporal. Sin embargo, a diferencia del correo postal, es más rápido, económico y fácil de distribuir, con capacidades como listas de distribución, envío masivo (incluido el spam), adjuntos y mensajes enriquecidos con HTML o imágenes (Kurose & Ross, 2017).

El sistema de correo electrónico en Internet se basa en tres componentes esenciales; la Figura 15 representa estos componentes.

Figura 15

Componentes principales del correo electrónico en Internet



Nota. Jaramillo, B., 2025.

La Figura 15 muestra los componentes principales para el servicio de correo electrónico, estos son:

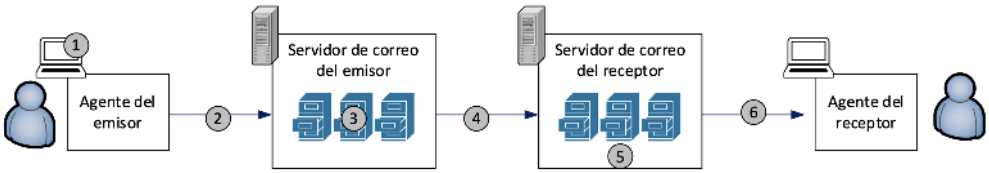
1. Un agente de usuario es el software que permite al remitente redactar, enviar y administrar sus mensajes; ejemplos de estos son Microsoft Outlook y Apple Mail. Cuando un usuario (emisor) compone un mensaje, este es enviado desde su agente de usuario al servidor de correo, donde se coloca en una cola de mensajes salientes.
2. Los servidores de correo forman la infraestructura central del correo electrónico. Cada usuario, tiene un buzón de correo en uno de estos servidores. El mensaje del emisor viajará desde el servidor de correo del emisor hasta el servidor de correo del destinatario, donde se almacenará en su buzón. Para acceder a sus mensajes, el destinatario debe autenticarse mediante su nombre de usuario y contraseña.
3. Esta arquitectura y funcionamiento se sustentan en el protocolo SMTP, el cual se encarga de la transferencia de mensajes entre servidores. En las siguientes secciones profundizaremos en los detalles técnicos de este protocolo y cómo interactúa con otros, como IMAP y POP3, para la recepción de correos.

SMTP

El protocolo SMTP, definido en el RFC 5321, es el pilar fundamental del correo electrónico en Internet. Este protocolo se encarga de transferir los mensajes entre los servidores de correo del remitente y del destinatario. A diferencia de HTTP, SMTP es una tecnología más antigua, cuyo primer estándar fue publicado en 1982. Aunque ha demostrado ser confiable y ampliamente utilizado, presenta ciertas limitaciones heredadas. Una de estas es la restricción a mensajes en formato ASCII de 7 bits, lo cual era adecuado en una época con recursos de transmisión limitados. Sin embargo, en la actualidad, esta restricción genera inconvenientes, ya que los datos multimedia deben ser codificados a ASCII antes de su envío y decodificados nuevamente al llegar a su destino. Esto contrasta con HTTP, que permite la transferencia directa de datos multimedia (Kurose & Ross, 2017).

El funcionamiento básico de SMTP puede explicarse con un ejemplo práctico, la Figura 16 representa la interacción de un remitente que quiere enviar un mensaje de texto a un destinatario.

Figura 16
Envío de email usando SMTP



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

1. El remitente utiliza su agente de usuario de correo electrónico, introduce la dirección de correo del destinatario (por ejemplo, destinatario@unaescuela.edu), redacta su mensaje y selecciona la opción de enviar.
2. El mensaje se envía al servidor de correo del remitente, donde es colocado en una cola de mensajes.

3. El cliente SMTP, que opera en el servidor de correo del remitente, detecta el mensaje en la cola y abre una conexión TCP con el servidor SMTP del destinatario.
4. Tras la fase inicial de negociación entre los servidores, el cliente SMTP envía el mensaje a través de la conexión TCP.
5. El servidor de correo del destinatario recibe el mensaje mediante su servidor SMTP, colocándolo en el buzón correspondiente.
6. El destinatario accede a su agente de usuario para leer el mensaje cuando lo desee.

Este proceso ilustra cómo SMTP facilita la comunicación asíncrona, garantizando que los mensajes se transfieran de forma eficiente entre servidores.

Protocolos de acceso para correo electrónico

Actualmente, el acceso al correo electrónico sigue una arquitectura cliente-servidor. El mensaje enviado por el emisor se almacena primero en su servidor de correo antes de ser transferido al servidor de correo del destinatario mediante el protocolo SMTP. Esta transferencia en dos etapas es necesaria para garantizar la entrega del mensaje, incluso si el servidor de destino está temporalmente inactivo. Una vez que el mensaje llega al servidor de correo del destinatario, este permanece allí hasta que el usuario lo recupera. Para esta operación, no se utiliza SMTP, ya que es un protocolo de inserción (*push*). En su lugar, se emplean protocolos de extracción (*pull*), como POP3, IMAP o HTTP, los cuales permiten transferir los mensajes del servidor de correo al dispositivo local del usuario. Estos protocolos garantizan que los usuarios puedan acceder a sus mensajes de forma eficiente y desde cualquier terminal conectado a la red

Protocolo de oficina de correos (POP3)

POP3 es un protocolo de acceso a correo electrónico sencillo, definido en el RFC 1939. Funciona en tres fases: autorización, transacción y actualización. En la fase de autorización, el cliente establece una conexión TCP en el puerto 110 y se autentica enviando un nombre de usuario y una contraseña en texto plano. Luego, en la fase de transacción, el cliente recupera los mensajes, pudiendo también marcarlos para eliminación, desmarcarlos o consultar estadísticas. Por último, en la fase de actualización, cuando el cliente ejecuta el comando quit, el servidor elimina los mensajes marcados.

Durante la transacción, el cliente ejecuta comandos y el servidor responde con "+OK" si el comando fue exitoso o "-ERR" si hubo un error. Entre los comandos básicos se encuentran user y pass para autenticación, y comandos como list, retr y dele para gestionar los mensajes. POP3 puede operar en dos modos: "descargar y borrar", donde los mensajes se eliminan después de la descarga, o "descargar y guardar", donde permanecen en el servidor. Esta simplicidad hace que el protocolo sea limitado pero efectivo para la gestión básica del correo electrónico (Kurose & Ross, 2017).

Internet Message Access Protocol (IMAP)

IMAP (Internet Message Access Protocol) es un protocolo avanzado que permite una gestión más flexible y eficiente del correo electrónico en comparación con POP3. A diferencia de este último, IMAP mantiene los mensajes almacenados en el servidor y sincroniza automáticamente los cambios entre todos los dispositivos del usuario. Esto significa que cualquier acción, como leer, mover o eliminar un mensaje, se reflejará en tiempo real en todos los dispositivos conectados.

Una de sus características más destacadas es la posibilidad de organizar los mensajes en múltiples carpetas en el servidor, permitiendo una estructura más ordenada. Además, IMAP ofrece la opción de acceder de forma parcial a los mensajes, descargando solo partes específicas, como los encabezados, lo cual es útil en conexiones de baja velocidad (Kurose & Ross, 2017).

El protocolo soporta una amplia gama de comandos avanzados, como la selección de carpetas (SELECT), la recuperación parcial de mensajes (FETCH) y la búsqueda con criterios específicos (SEARCH). También permite mantener sesiones persistentes, lo que mejora la eficiencia al gestionar múltiples operaciones durante una misma conexión TCP.

Otra ventaja importante es la capacidad de recibir notificaciones en tiempo real, lo que permite al usuario estar siempre actualizado sobre la llegada de nuevos correos. En cuanto a la seguridad, IMAP se integra con métodos de autenticación segura, como OAuth, y soporta cifrado mediante TLS/SSL para proteger la conexión y la privacidad de los datos.

En términos de uso, los usuarios pueden optar por leer los mensajes directamente desde el servidor sin necesidad de descargarlos, o mantener copias sincronizadas en sus dispositivos. Esto lo convierte en la solución ideal para quienes necesitan acceder a sus correos desde diferentes lugares y dispositivos, a diferencia de POP3, que es más adecuado para usuarios con acceso limitado a Internet.

2.6 Simuladores, escenarios de prueba y requerimientos técnicos.

Los simuladores de red son herramientas esenciales para el aprendizaje y la validación de configuraciones de red, ya que permiten recrear entornos reales de infraestructura de TI sin necesidad de equipos físicos. Estas aplicaciones emulan el comportamiento de dispositivos de red, como routers, switches, servidores y clientes, posibilitando la configuración, prueba y análisis del tráfico de datos.

En el mercado existen múltiples opciones, tanto de software libre como de pago. Algunos ejemplos populares incluyen:

- **GNS3 (Graphical Network Simulator 3):** Es un simulador avanzado que permite la integración con imágenes de dispositivos reales, ideal para

entornos complejos y pruebas detalladas (*Getting Started With GNS3 | GNS3 Documentation*, s. f.).

- **Cisco Packet Tracer:** Ofrecido por Cisco, es una herramienta educativa que permite crear redes virtuales con un enfoque didáctico, facilitando la comprensión de conceptos básicos y avanzados (Cisco Packet Tracer, s. f.).
- **EVE-NG (Emulated Virtual Environment Next Generation):** Una plataforma colaborativa para simular grandes infraestructuras con múltiples tecnologías (EVE-NG Ltd, 2025)
- **NS3 (Network Simulator 3):** Utilizado en investigaciones y simulaciones detalladas de redes, orientado a académicos y desarrolladores (Nsnam, s. f.).

Estas herramientas permiten implementar y probar configuraciones de red sin riesgos, además de ofrecer opciones para monitorear y analizar el tráfico, replicando de manera precisa la comunicación entre dispositivos y aplicaciones.

Escenario de simulación y requerimientos técnicos

Para consolidar los conocimientos adquiridos en las unidades anteriores, se desarrollará un escenario práctico basado en la implementación de una red de datos que integre los siguientes servicios fundamentales:

- **Servicio DNS:** Permite la resolución de nombres de dominio a direcciones IP.
- **Servicio DHCP:** Encargado de la asignación dinámica de direcciones IP a los dispositivos de la red.
- **Servicio HTTP:** Proporciona acceso a recursos web mediante una arquitectura cliente-servidor.

Este escenario contempla una red con al menos un servidor centralizado, un conjunto de dispositivos clientes, y la interconexión mediante dispositivos de red simulados. Los requerimientos incluyen la configuración de los servicios mencionados, así como la verificación del correcto funcionamiento de la conectividad y la resolución de nombres en el entorno simulado.

Esquema de conexión

En el diseño de redes, la arquitectura de los servicios juega un papel crucial para la eficiencia, escalabilidad y seguridad. Existen dos esquemas principales que se implementarán como opciones en el entorno de simulación:

Opción 1: Servidor centralizado para todos los servicios

En esta arquitectura, un único servidor administra múltiples servicios de red, tales como DNS, DHCP y HTTP.

Características:

- **Gestión centralizada:** Todas las configuraciones y servicios se concentran en un solo servidor, lo que simplifica la administración.
- **Menores costos iniciales:** Requiere menos hardware o recursos virtuales, reduciendo los costos asociados a la infraestructura.

Desventajas:

- La disponibilidad se ve comprometida, ya que, si el servidor falla, todos los servicios se interrumpen.
- Puede haber una mayor carga de procesamiento en el servidor único.

Opción 2: Servicios distribuidos en servidores independientes

En esta alternativa, cada servicio se implementa en hardware o máquinas virtuales separadas. Por ejemplo, un servidor para DNS, otro para DHCP y un tercero para **HTTP.**

Características:

- **Mayor redundancia:** Si un servidor falla, los demás servicios permanecen operativos.

- **Escalabilidad:** Es más fácil distribuir la carga, dado que cada servidor puede ser optimizado y dimensionado según las necesidades específicas del servicio.
- **Mantenimiento especializado:** Cada servidor puede ser configurado y mantenido de forma independiente, permitiendo actualizaciones y ajustes sin afectar a los demás servicios.

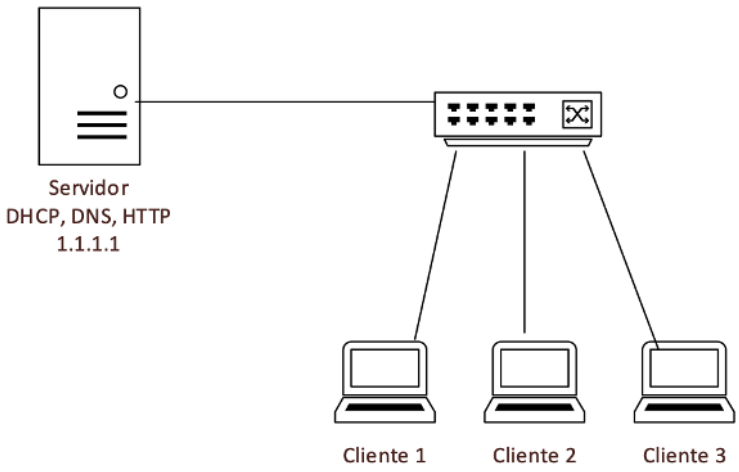
Desventajas:

- Requiere más recursos y una mayor inversión inicial.
- La configuración y administración pueden ser más complejas al tener múltiples puntos de control.

Las Figura 17 y Figura 18, representan el esquema de conexión lógica para la opción 1 y 2 de la simulación respectivamente.

Figura 17

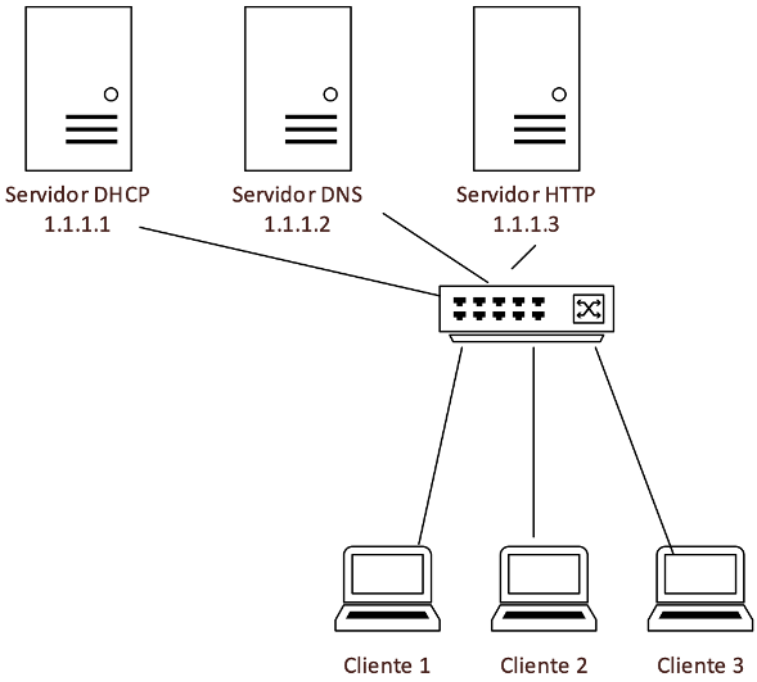
Esquema – Opción 1 de simulación



Nota. Jaramillo, B., 2025.

Figura 18

Esquema – Opción 2 de simulación



Nota. Jaramillo, B., 2025.

Ambos esquemas serán analizados y probados en el simulador, lo que permitirá observar el impacto en términos de rendimiento, disponibilidad y facilidad de administración en distintos escenarios de red.

Requerimientos Técnicos

Sobre los escenarios planteados y tomando en cuenta el uso de cualquier herramienta de simulación disponible, se procederá a implementar los servicios fundamentales de red que se han descrito en las unidades anteriores. Estos servicios, esenciales para la gestión de una infraestructura

de red, comprenden DHCP, DNS, HTTP y HTTPS. A continuación, se detallan los requerimientos específicos para cada servicio, los cuales permitirán validar y probar la funcionalidad y conectividad de la arquitectura de red simulada.

1. **DHCP (Dynamic Host Configuration Protocol)**

   - **Segmento de red:** 1.1.1.0/24
   - **Rango de asignación de IPs dinámicas:** Desde la dirección 1.1.1.50 hasta la 1.1.1.200
   - **IPs reservadas:** Direcciones 1.1.1.1 a 1.1.1.49 para otros servicios como puerta de enlace (gateway), impresoras y servidores críticos.
   - **Tiempo de arrendamiento:** 1 día (24 horas)
   - **Opciones adicionales:** Configurar la dirección de puerta de enlace predeterminada y el servidor DNS en los parámetros de DHCP.

2. **DNS (Domain Name System)**

   - **Configuración básica:**

     - **Entrada tipo A:** Definir el registro A para el dominio arquitecturaderedes.com.
     - **Dirección IP:** El registro debe apuntar a la dirección IP asignada al servidor DNS según el esquema de conexión (ver Figuras 16 y 17).

   - **Validación:** Comprobar que el nombre de dominio resuelva correctamente en toda la red simulada.
   - **Entradas adicionales:** Crear alias o registros CNAME si es necesario para pruebas complementarias.

3. **HTTP (Hypertext Transfer Protocol)**

   - **Configuración básica:** Activar y habilitar el servicio HTTP en el servidor asignado.
   - **Implementación de contenido:**

     - Consultar un script de ejemplo con una página web simple en HTML.

- El contenido debe ser texto estático con un mensaje básico como "Bienvenido a la Arquitectura de Redes".

- **Ruta de acceso:** Definir un directorio raíz público (/var/www/html en servidores Linux, o equivalente).

4. **HTTPS (Hypertext Transfer Protocol Secure)**

   - **Configuración:**

     - Activar HTTPS en el servidor web.
     - Generar o instalar un certificado SSL/TLS de prueba para el dominio arquitecturaderedes.com.

   - **Validación:** Asegurarse de que el acceso a la página web sea posible tanto mediante HTTP como HTTPS, verificando la seguridad del certificado.
   - **Redirección:** Configurar la redirección automática de HTTP a HTTPS si se requiere.

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Acceda al Curso de Networking Basics ofrecido por Cisco NetAcad, es un curso gratuito diseñado para reforzar y ampliar su comprensión de los protocolos de red. Este curso le permitirá explorar en profundidad cómo funcionan los protocolos básicos y avanzados en una red, así como su aplicación en escenarios reales, complementando los temas de redes multimedia, seguridad y conectividad en la nube que hemos trabajado.

¡Felicitaciones por completar la unidad 2 sobre protocolos básicos de la capa de aplicación! Ha dado un paso importante en su estudio, comprendiendo el funcionamiento de protocolos clave como DNS, DHCP, HTTP y HTTPS, así como sus roles en la conectividad y seguridad de la red. Ahora que ha alcanzado este hito, le animo a continuar avanzando con entusiasmo en su camino de aprendizaje.

2. Antes de continuar, le recomiendo que complete la autoevaluación 2, esta actividad le permitirá evaluar su comprensión de los conceptos abordados en esta unidad. No dude en dedicarle tiempo y esfuerzo, ya que le ayudará a consolidar lo aprendido.

   Las preguntas están relacionadas con los temas abordados en la Unidad 2, sobre el funcionamiento de protocolos clave como DNS, DHCP, HTTP y HTTPS, así como sus roles en la conectividad y seguridad de la red.

¡Mucho éxito en su camino de aprendizaje!

Autoevaluación 2

Seleccione la opción de respuesta que considere correcta.

1. ¿Cuál es una de las mejoras principales introducidas en HTTP/2 respecto a HTTP/1.1?

   A. Uso exclusivo del método GET
   B. Multiplexación de solicitudes y respuestas
   C. Dependencia de conexiones persistentes obligatorias

2. ¿Qué puerto estándar utiliza HTTPS para establecer una conexión segura?

   A. Puerto 21

B. Puerto 80
C. Puerto 443

3. ¿Qué función cumple el protocolo DNS en la arquitectura de red?

A. Proporciona seguridad en el transporte de datos
B. Traduce nombres de dominio a direcciones IP
C. Configura dinámicamente direcciones IP

4. ¿Qué tipo de servidor DNS mantiene la información de dominios específicos, como .com o .edu?

A. Servidor DNS raíz
B. Servidor DNS autoritativo
C. Servidor DNS de nivel superior (TLD)

5. En un proceso HTTPS, ¿qué información contiene el certificado SSL/TLS enviado por el servidor?

A. La dirección IP del cliente
B. La clave pública del servidor
C. El historial de conexiones

6. ¿Cuál es la característica principal del protocolo DHCP?

A. Asignación manual de direcciones IP
B. Configuración dinámica de direcciones IP
C. Resolución de nombres de dominio

7. ¿Cuál de las siguientes es una desventaja del protocolo HTTP/1.0?

A. Soporte limitado para métodos HTTP
B. Cierre de conexión después de cada solicitud
C. Falta de autenticación de usuarios

8. ¿Cómo asegura HTTPS la confidencialidad de los datos transmitidos?

   A. Utilizando el protocolo de resolución de nombres
   B. Cifrando las comunicaciones con SSL/TLS
   C. Implementando conexiones no persistentes

9. ¿Qué ocurre si un servidor DNS local no encuentra una dirección IP en su caché?

   A. Rechaza la solicitud del cliente
   B. Reenvía la solicitud a un servidor DNS raíz
   C. Asigna una dirección IP aleatoria

10. ¿Cuál es una de las principales diferencias entre HTTP y HTTPS?

    A. HTTPS utiliza UDP mientras que HTTP utiliza TCP
    B. HTTPS utiliza cifrado mediante SSL/TLS
    C. HTTP permite solo métodos GET y POST

Ir al solucionario

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño. En caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje.

Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 5

¡Bienvenido a la Unidad 3: " Desarrollo y Monitoreo de Aplicaciones de Red "!

En esta unidad, profundizaremos en los conceptos y técnicas para implementar y gestionar aplicaciones que operan en redes orientadas a datos. Exploraremos la programación de sockets tanto con UDP como con TCP, lo que le permitirá comprender cómo se construyen aplicaciones de red desde la base. Posteriormente, estudiaremos el diseño de aplicaciones con arquitecturas basadas en REST y GraphQL, enfocándonos en la interacción eficiente entre servicios. Finalmente, revisaremos las redes de distribución de contenido (CDN), analizando un caso de estudio práctico con servicios como Netflix.

Prepárese para aplicar lo aprendido en escenarios de redes reales. ¡Es momento de fortalecer sus habilidades en diseño y gestión de aplicaciones de red!

Unidad 3. Desarrollo y Monitoreo de Aplicaciones de Red

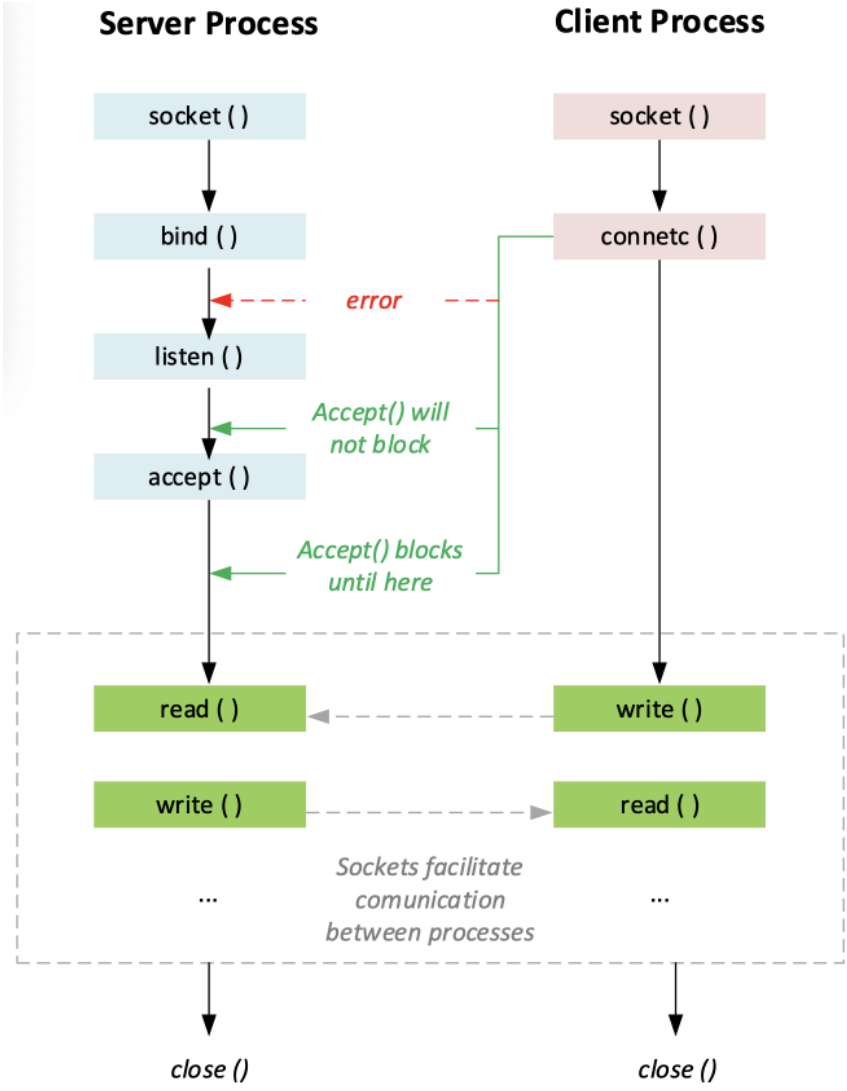
3.1 Programación de sockets.

La programación de sockets permite conectar dos nodos en una red para que se comuniquen entre sí. En este proceso, un socket actúa como servidor, escuchando en un puerto específico de una dirección IP, mientras que otro socket, que actúa como cliente, se conecta al servidor para establecer la comunicación (IBM I 7.3, s. f.).

Muchas aplicaciones de red se componen de dos programas: uno cliente y otro servidor, que operan en distintos sistemas terminales. Al ejecutarse, estos programas generan procesos cliente y servidor, los cuales se comunican intercambiando datos a través de sockets. La tarea principal del desarrollador es, por tanto, crear el código necesario para estos programas. A continuación, la Figura 19 representa el modelo de estados de la programación en sockets.

Figura 19

Diagrama de estados del modelo cliente-servidor en programación de sockets



Nota. Adaptado de *Socket programming* [Ilustración], por IBM I 7.3, 2023, IBM, CC BY 4.0.

En la figura anterior se muestra un diagrama ilustra el flujo de operaciones de comunicación entre un proceso servidor y un proceso cliente utilizando programación de sockets.

1. **Proceso del Servidor:**
    - **socket():** Crea un socket para comunicarse.
    - **bind():** Asigna el socket a una dirección IP y un puerto específico.
    - **listen():** Configura el socket para escuchar conexiones entrantes.
    - **accept():** Espera hasta que un cliente se conecte. Una vez que la conexión es aceptada, el servidor puede iniciar la lectura y escritura de datos.
2. **Proceso del Cliente:**
    - **socket():** Crea un socket para establecer la conexión.
    - **connect():** Intenta conectarse al servidor. Si el servidor acepta la conexión, se establece el canal de comunicación.
3. **Comunicación:**
    - Tras establecer la conexión, ambos procesos pueden intercambiar mensajes a través de operaciones de lectura (read()) y escritura (write()).
4. **Cierre:**
    - Al finalizar la comunicación, ambos procesos cierran sus sockets mediante la llamada close().

Este modelo es fundamental para aplicaciones de red basadas en el protocolo TCP, asegurando una conexión fiable y bidireccional entre cliente y servidor

3.2 Programación de sockets con UDP.

En esta sección, se presenta una aplicación cliente-servidor simple que utiliza programación de sockets con el protocolo UDP. Los sockets son un mecanismo que permite la comunicación entre procesos en una red. Un

proceso emisor (cliente) envía datos a un proceso receptor (servidor) a través de estos sockets. Cada socket se identifica mediante una dirección compuesta por la dirección IP del host y el número de puerto, lo que permite a los routers en la red enrutar correctamente los paquetes.

En este ejemplo, el servidor UDP se mantiene a la escucha de mensajes en un puerto específico. Cuando el cliente envía un mensaje, el servidor lo recibe y muestra la información en la consola. Este proceso es esencial para aplicaciones de red que requieren comunicación sin conexión, en la que los datos se envían de forma independiente sin garantizar su entrega.

Código del servidor UDP (servidor.py)

Este script abre un puerto UDP para recibir mensajes de cualquier cliente.

```
import socket

def open_udp_port(port):
    # Crear un socket UDP
    udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    # Vincular el socket al puerto especificado
    udp_socket.bind(('', port))
    print(f'Servidor UDP escuchando en el puerto {port}...')
    while True:
        # Esperar por datos
        data, addr = udp_socket.recvfrom(1024)  # buffer de 1024 bytes
        print(f'Recibido {data.decode()} de {addr}')
if __name__ == "__main__":
```

```
    port_number = 28  # Especificar el puerto

    open_udp_port(port_number)
```

Código del cliente UDP (cliente.py)

Este script envía un mensaje al servidor UDP y muestra cualquier respuesta recibida.

```
import socket

def send_message(message, server_ip, server_port):

    # Crear un socket UDP

    udp_client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    try:

        # Enviar el mensaje al servidor

        udp_client_socket.sendto(message.encode(), (server_ip, server_port))

        print(f'Mensaje enviado: {message}')

        # Recibir respuesta (opcional)

        response, _ = udp_client_socket.recvfrom(1024)  # Tamaño del buffer

        print(f'Respuesta del servidor: {response.decode()}')

    except Exception as e:

        print(f'Ocurrió un error: {e}')

    finally:

        # Cerrar el socket

        udp_client_socket.close()
```

```
if __name__ == "__main__":
    server_ip = '127.0.0.1'  # IP del servidor (localhost)
    server_port = 28         # Puerto del servidor
    message = "Hola, servidor!"  # Mensaje a enviar
    send_message(message, server_ip, server_port)
```

Descripción del funcionamiento

1. **Servidor**:
   - Crea un socket UDP que escucha en el puerto 28.
   - Espera mensajes de clientes y muestra el contenido del mensaje junto con la dirección del cliente.
2. **Cliente**:
   - Crea un socket UDP para enviar un mensaje al servidor.
   - Envía el mensaje al servidor que escucha en 127.0.0.1:28.
   - Recibe y muestra cualquier respuesta opcional del servidor.

Este ejemplo demuestra los conceptos básicos de comunicación en red utilizando sockets y es útil para aplicaciones en las que no se requiere una conexión persistente.

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Explore la página "Sockets", el cual proporciona una introducción detallada a la programación de sockets en Java. Esta página abarca conceptos fundamentales, ejemplos prácticos y explicaciones que le ayudarán a consolidar su

comprensión sobre los protocolos de comunicación en redes. Revisar este material será una excelente oportunidad para reforzar los conocimientos adquiridos en esta asignatura (Gálvez, s.f.).

2. Investigue las diferencias entre los protocolos TCP y UDP en la programación de sockets. Desarrolle una aplicación cliente-servidor utilizando sockets TCP para enviar y recibir mensajes de texto entre dos nodos.
3. Realice una comparación detallada entre el uso de sockets TCP y UDP, evaluando aspectos como fiabilidad, velocidad y eficiencia en diferentes escenarios prácticos

Nota: Por favor complete las actividades en un cuaderno o documento Word

¡Felicidades por culminar la semana 5! Ha dado un paso importante en la comprensión y práctica de la programación de sockets, tanto con UDP como con TCP. Continúe con ese mismo entusiasmo, ya que en la próxima semana seguiremos profundizando en conceptos fundamentales para el diseño de aplicaciones de red. ¡Adelante, siga así!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 6

Unidad 3. Desarrollo y Monitoreo de Aplicaciones de Red

3.3 Diseño de aplicaciones con REST y GraphQL

REST (Representational State Transfer) es un modelo de arquitectura para la comunicación eficiente entre aplicaciones y servicios en red, basado en el protocolo HTTP. En la arquitectura de redes, REST es fundamental para gestionar interacciones entre sistemas distribuidos, especialmente en

aplicaciones orientadas a datos. Este enfoque se caracteriza por la simplicidad, la escalabilidad y el uso óptimo de recursos en red, lo que lo hace ideal para entornos web y servicios en la nube (What Is RESTful API? - RESTful API Explained - AWS, s. f.).

Principios REST en la comunicación en red

Los principios de REST guían la comunicación en red mediante reglas claras que permiten una interacción eficiente entre clientes y servidores. REST se basa en conceptos como la interfaz uniforme, la identificación de recursos únicos, la comunicación sin estado y el uso de operaciones estándar del protocolo HTTP. Estos principios aseguran la consistencia, escalabilidad y simplicidad en el intercambio de información entre aplicaciones distribuidas. La Figura 20 presenta un detalle de las características de cada principio, ilustrando cómo estos elementos contribuyen a una comunicación estructurada y estándar en la arquitectura de redes.

Figura 20
Principios de REST

Nota. Adaptado de *¿Qué es una API RESTful?* [Ilustración], por AWS, s.f., AWS, CC BY 4.0.

Como se aprecia, la Figura 20 sintetiza visualmente cómo estos principios trabajan en conjunto para establecer una arquitectura robusta y eficiente en aplicaciones de red.

REST en la arquitectura de redes

En redes modernas, REST se utiliza en aplicaciones que requieren comunicaciones rápidas y confiables. Algunos ejemplos incluyen:

- **Servicios web:** APIs REST permiten a diferentes aplicaciones y servicios en la red intercambiar información, facilitando la integración de sistemas.
- **Aplicaciones móviles:** Los dispositivos móviles utilizan REST para sincronizar datos con servidores de backend.
- **Sistemas distribuidos:** REST simplifica la gestión de servicios que están distribuidos en diferentes ubicaciones o regiones.

Ventajas de REST en redes

- **Escalabilidad:** REST es ideal para entornos de alta demanda, como aplicaciones web masivas.
- **Interoperabilidad:** Los sistemas que implementan REST pueden comunicarse sin necesidad de compartir la misma tecnología de desarrollo.
- **Facilidad de implementación:** REST es compatible con cualquier infraestructura que soporte HTTP, sin requerir herramientas adicionales.

GraphQL: Comunicación eficiente en la arquitectura de redes

GraphQL es un lenguaje de consulta y manipulación de datos para APIs desarrollado por Facebook. A diferencia de REST, que entrega recursos completos mediante endpoints específicos, GraphQL permite al cliente definir exactamente qué datos necesita. Este enfoque mejora la eficiencia en la comunicación entre aplicaciones y servicios en una red, optimizando el uso del ancho de banda y reduciendo el volumen de datos transferidos (*Managed GraphQL APIs - Amazon AppSync - AWS*, s. f.).

Características de GraphQL en redes

1. **Consulta Personalizada:**
   Un cliente puede solicitar únicamente los campos de un recurso que necesita. Por ejemplo, en lugar de obtener todos los detalles de un usuario, un cliente puede pedir solo el nombre y el correo electrónico:

```
query {
  usuario(id: "123") {
    nombre
    correo
  }
}
```

   El servidor devuelve solo estos datos:

```
{
  "data": {
    "usuario": {
      "nombre": "Juan Pérez",
      "correo": "juan.perez@example.com"
    }
  }
}
```

2. **Endpoint único:**

A diferencia de REST, que requiere múltiples endpoints para diferentes operaciones, GraphQL utiliza un único endpoint para todas las consultas, mutaciones y suscripciones. Esto simplifica la estructura de la API.

GraphQL en la arquitectura de redes

GraphQL es una solución ideal en arquitecturas de red donde la optimización de datos y la flexibilidad son cruciales. Algunos escenarios donde se utiliza incluyen:

- **Aplicaciones móviles:** Reduce la cantidad de solicitudes a los servidores, lo que es clave en dispositivos con conectividad limitada.
- **Servicios distribuidos:** Permite consolidar información desde múltiples fuentes en una sola respuesta.
- **Desarrollo frontend:** Los equipos de desarrollo pueden modificar consultas sin necesidad de cambiar la lógica del servidor, lo que acelera el desarrollo de interfaces de usuario.

Ventajas de GraphQL en redes

- **Optimización del tráfico:** Minimiza el volumen de datos enviados entre cliente y servidor.
- **Flexibilidad para el cliente:** El cliente tiene control total sobre la forma y cantidad de datos recibidos.
- **Mejora en el rendimiento:** Al reducir el número de solicitudes y la cantidad de datos transferidos, mejora la latencia de red.

A continuación, la Tabla 4 presenta las principales diferencias entre REST y GraphQL, destacando aspectos clave como la flexibilidad en las consultas, la gestión de recursos y la eficiencia en la transferencia de datos. Este análisis permite comparar ambos enfoques para seleccionar la mejor solución según las necesidades de desarrollo.

Tabla 4
Comparación entre REST y GraphQL

Aspecto	REST	GraphQL
Estructura de endpoints	Múltiples endpoints	Un único endpoint
Volumen de datos	Puede entregar más datos de los necesarios	Solo entrega los datos solicitados
Flexibilidad	Fija	Alta
Tipado	No tipado	Fuertemente tipado
Eficiencia	Menor en escenarios complejos	Alta

Nota. Jaramillo, B., 2025.

Con base en nuestro estudio, tanto REST como GraphQL representan enfoques esenciales en la arquitectura de redes para la comunicación entre aplicaciones. REST, con sus principios bien definidos, permite una estructura estándar y escalable sobre el protocolo HTTP, mientras que GraphQL ofrece flexibilidad al permitir a los clientes solicitar solo los datos necesarios, optimizando así el tráfico de red. Desde una perspectiva de redes, estas tecnologías mejoran la eficiencia del consumo de recursos, la escalabilidad y la capacidad de respuesta, factores críticos para aplicaciones distribuidas en infraestructuras modernas. Implementar correctamente estos modelos contribuye a una arquitectura más robusta, reduciendo la latencia y mejorando la experiencia de los usuarios en entornos de gran carga y múltiples servicios conectados.

3.4 Análisis de tráfico de aplicación con Wireshark

El análisis de protocolos de aplicación es una práctica fundamental para comprender y optimizar las comunicaciones que se llevan a cabo en una red. Los protocolos de aplicación, como HTTP, SMTP, DNS y REST, definen las reglas y estructuras que permiten el intercambio de información entre

dispositivos y servicios. Analizar estos protocolos permite diagnosticar problemas, verificar configuraciones y mejorar el rendimiento de las aplicaciones que dependen de la red.

¿Qué es un sniffer?

Un sniffer es una herramienta que captura y monitorea el tráfico de red en tiempo real. Su función es interceptar los paquetes que circulan a través de una red, permitiendo a los usuarios observar y analizar cada comunicación en detalle. Los sniffers son esenciales en el análisis de redes, ya que proporcionan información sobre la estructura de los paquetes, los protocolos utilizados, la dirección de origen y destino, y el contenido de los mensajes (*Wireshark · Go Deep*, s. f.)

Según Wireshark (s. f.), un sniffer es útil para:

- Diagnosticar problemas de conexión.
- Analizar el comportamiento de aplicaciones cliente-servidor.
- Detectar amenazas o vulnerabilidades de seguridad.

Entre las herramientas más conocidas de este tipo se encuentra Wireshark, que se ha consolidado como el estándar en la captura y análisis de tráfico de red.

Introducción a Wireshark

Wireshark es una herramienta de análisis de red y captura de paquetes que permite inspeccionar a fondo el tráfico de aplicaciones en diferentes capas del modelo OSI. Ofrece funcionalidades avanzadas para el monitoreo y diagnóstico de problemas en redes locales e Internet, siendo una de las herramientas más utilizadas por profesionales de TI (*Wireshark · Go Deep*, s. f.).

La Figura 21 representa las características de Wireshark orientadas al análisis de protocolos de aplicación.

Figura 21

Características de Wireshark

Nota. Adaptado de *The world's most popular network protocol analyzer* [Ilustración], por Wireshark Go Deep, s. f., Wireshark, CC BY 4.0.

Con base en la figura 21, podemos indicar las características clave de Wireshark que permitirán el análisis de tráfico en la capa de aplicación.

1. **Captura en tiempo real:** Wireshark permite capturar tráfico en múltiples interfaces de red (Ethernet, Wi-Fi, Bluetooth), identificando los diferentes paquetes transmitidos y recibidos por el sistema.
2. **Soporte para múltiples protocolos:** La herramienta reconoce y analiza cientos de protocolos, incluidos los más relevantes para las aplicaciones de red orientadas a datos, como HTTP, DNS, DHCP y REST.
3. **Visualización detallada:** Los paquetes capturados se presentan en un formato organizado por capas, mostrando detalles como la cabecera del protocolo, la dirección IP de origen y destino, el número de puerto y los datos transmitidos.
4. **Filtros avanzados:** Los filtros permiten reducir el volumen de información capturada, facilitando el enfoque en sesiones específicas. Por ejemplo, es posible aplicar un filtro para visualizar únicamente el tráfico HTTP o capturar solicitudes DNS en tiempo real.
5. **Análisis de tráfico de aplicaciones:** Al estudiar aplicaciones basadas en sockets (UDP/TCP) o servicios web (REST, GraphQL), Wireshark es capaz de identificar las interacciones, verificar respuestas del servidor y analizar tiempos de respuesta, errores y datos transferidos

Ejemplo práctico:

Imaginemos un escenario donde se desarrolla una aplicación cliente-servidor que utiliza sockets TCP. Con Wireshark, es posible capturar el tráfico generado por la aplicación para verificar:

- Si las solicitudes del cliente llegan al servidor.
- Si el servidor responde con la información correcta.
- La cantidad de datos enviados y recibidos en cada transacción.

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

Actividades de aprendizaje recomendadas

1. comparación práctica de sus diferencias en aspectos como eficiencia, flexibilidad y simplicidad. Desarrolle una API sencilla para gestionar un recurso, implementando operaciones básicas como lectura, creación y eliminación de datos.
2. Siga los cursos de enseñanza gratuitos disponibles en la página oficial de Wireshark. Realice ejercicios prácticos donde capture y analice el tráfico de una aplicación de red, identificando las solicitudes y respuestas de un servicio web.

Estas actividades le permitirán afianzar los conceptos de diseño de aplicaciones orientadas a datos y el análisis del tráfico generado por dichas aplicaciones, proporcionando una comprensión sólida de su funcionamiento en un entorno de red real.

¡Felicidades por culminar la semana 6! Ha reforzado su conocimiento en el diseño de aplicaciones con REST y GraphQL, además de adquirir habilidades esenciales para el análisis de tráfico en redes. Siga adelante con el mismo compromiso y entusiasmo, ya que en las próximas semanas seguiremos explorando y consolidando estos conocimientos. ¡Continúe así!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 7

Unidad 3. Desarrollo y Monitoreo de Aplicaciones de Red

3.5 Laboratorio de análisis de protocolos de capa de aplicación con Wireshark

Una manera efectiva de profundizar en la comprensión de los protocolos de red es observarlos en funcionamiento e interactuar con ellos. Esto implica analizar la secuencia de mensajes intercambiados entre las entidades, examinar los detalles de su operación y provocar ciertas acciones en los protocolos para luego estudiar sus efectos y resultados. Esta práctica se puede llevar a cabo tanto en entornos simulados como en redes reales, como Internet (Kurose & Ross, 2017).

A través de este laboratorio, se espera que desarrolle habilidades prácticas para identificar el flujo de comunicación entre cliente y servidor, evaluar tiempos de respuesta, analizar consultas DNS y reconocer otros elementos esenciales en la transmisión de datos en redes.

En este caso práctico, se realizará una captura de tráfico al visitar un sitio web (por ejemplo, https://utpl.edu.ec), con el objetivo de examinar los protocolos de comunicación, analizar los paquetes generados y estudiar cómo se comporta la red durante la carga de una página web.

Instrucciones del Laboratorio

1. **Preparación del entorno:**
    - Asegúrese de que Wireshark esté instalado en su equipo, puede descargarlo desde el sitio oficial de Wireshark.
    - Realice una configuración básica de Wireshark seleccionando la interfaz de red activa para realizar la captura.
2. **Iniciar la captura de tráfico:**
    - Abra Wireshark y seleccione la opción de captura en la interfaz de red correspondiente.
    - Comience la captura y, en paralelo, desde su navegador acceda al sitio web que va a evaluar (por ejemplo, https://utpl.edu.ec).
3. **Detener la captura:**
    - Una vez que la página haya cargado por completo, detenga la captura en Wireshark.
    - Guarde el archivo de captura para análisis posterior.
4. **Análisis del tráfico capturado**
    - **Identificación del protocolo:** Filtre los paquetes por protocolos (HTTP, HTTPS, DNS) usando los filtros de Wireshark (ejemplo: http, dns o tcp).
    - **Análisis de consultas DNS:** Localice las solicitudes y respuestas DNS que resuelven la dirección IP del dominio utpl.edu.ec.
    - **Evaluación del tráfico HTTP o HTTPS:**
        - Identifique los paquetes relacionados con la comunicación HTTP o HTTPS.
        - Examine los métodos HTTP utilizados (GET, POST, etc.) y los códigos de respuesta.

- **Tiempos de respuesta:**
  - Analice los tiempos de respuesta del servidor para las solicitudes realizadas.
  - Determine el tiempo entre el envío de una solicitud y la recepción de la respuesta.

5. **Conclusiones:**
   - Resuma sus hallazgos, describiendo el comportamiento de los protocolos analizados.
   - Mencione los tiempos de respuesta, tipo de contenido descargado y otros aspectos relevantes.

¡Excelente trabajo! Ha completado el estudio de la tercera unidad. Ahora es momento de avanzar a la unidad 4. Pero antes de hacerlo, me gustaría invitarle a realizar las siguientes actividades para reforzar y aplicar los conocimientos adquiridos.

Actividades de aprendizaje recomendadas

1. Ejecute el laboratorio desarrollado en clase utilizando otros sitios web de su elección o aplicaciones personalizadas. Analice el tráfico capturado para identificar protocolos, tiempos de respuesta, direcciones IP, y cualquier otro dato relevante que le permita comprender mejor la estructura y operación de las comunicaciones en red.
2. Investigue las funcionalidades adicionales que Wireshark ofrece. Explore características avanzadas como filtros de captura, análisis estadístico, reconstrucción de sesiones y decodificación de protocolos. Aplique estos conocimientos para mejorar sus habilidades en el monitoreo y diagnóstico del tráfico de red.

Estas actividades le permitirán reforzar y ampliar su comprensión sobre el análisis de protocolos de aplicación, así como familiarizarse con el uso avanzado de herramientas de monitoreo en entornos reales. ¡Continúe explorando y aprendiendo!

3. Antes de continuar, le recomiendo que complete la autoevaluación 3, esta actividad le permitirá evaluar su comprensión de los conceptos abordados en esta unidad. No dude en dedicarle tiempo y esfuerzo, ya que le ayudará a consolidar lo aprendido. Por favor, dedique atención a las preguntas que se presentan, las cuales están relacionadas con los temas abordados en la Unidad 3, que incluye el desarrollo y monitoreo de Aplicaciones de Red.

¡Mucho éxito en su camino de aprendizaje!

Autoevaluación 3

Seleccione la opción de respuesta que considere correcta.

1. ¿Cuál es el propósito principal de los sockets en la comunicación en red?

   A. Realizar encriptación de datos de extremo a extremo
   B. Establecer una interfaz de comunicación entre procesos en diferentes nodos de la red
   C. Autenticar usuarios en el servidor

2. En la programación con sockets UDP, ¿qué característica es un desafío importante?

   A. La falta de fiabilidad en la entrega de mensajes
   B. La necesidad de cifrado manual
   C. La limitación en el tamaño de los mensajes

3. ¿Cuál de los siguientes es un principio clave de REST en la arquitectura de aplicaciones?

A. Almacenamiento de sesiones en el servidor
B. Comunicación sin estado
C. Uso exclusivo de XML para la representación de recursos

4. ¿Qué ventaja tiene GraphQL sobre REST en términos de eficiencia de consultas?

A. Utiliza menos ancho de banda
B. Permite a los clientes solicitar exactamente los datos que necesitan
C. Requiere menos servidores para ejecutarse

5. En Wireshark, ¿cuál es el propósito de los filtros de captura?

A. Seleccionar solo ciertos paquetes para ser almacenados durante la captura
B. Modificar el contenido de los paquetes capturados
C. Bloquear el tráfico de red no deseado

6. Al capturar tráfico HTTP en Wireshark, ¿qué puerto suele indicar la comunicación no cifrada?

A. 443
B. 21
C. 80

7. ¿Cuál es el rol principal del método GET en una API REST?

A. Modificar un recurso existente
B. Crear un nuevo recurso en el servidor
C. Obtener información sobre un recurso

8. ¿Qué tipo de tráfico es más fácil de analizar en Wireshark utilizando filtros específicos?

   A. Tráfico encriptado HTTPS
   B. Tráfico de texto sin cifrar, como HTTP o DNS
   C. Tráfico P2P cifrado

9. ¿Qué característica distingue a los sockets TCP en comparación con los UDP?

   A. Garantizan el orden y la entrega de los mensajes
   B. No requieren conexión previa entre los nodos
   C. Ofrecen una comunicación no confiable pero rápida

10. ¿Cuál es un objetivo típico de ejecutar un laboratorio de análisis de tráfico de red en Wireshark?

    A. Modificar configuraciones de enrutamiento en el servidor
    B. Identificar protocolos y tiempos de respuesta en la comunicación
    C. Interrumpir el servicio en aplicaciones web

Ir al solucionario

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño. En caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje.

Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Semana 8

Actividades finales del bimestre

Es hora de reforzar los conocimientos adquiridos resolviendo las siguientes actividades:

1. Cree un mapa conceptual interactivo: identifique los temas relevantes de las unidades, como los fundamentos de la computación en la nube, los modelos de servicio o los despliegues en la nube; luego, organice los conceptos y relaciones clave, agrega enlaces a recursos adicionales y proporcionar ejemplos prácticos.
2. Revise nuevamente en la guía didáctica, los temas relacionados con la unidad 1 “Introducción a la capa de aplicación”, unidad 2 “Protocolos básicos de la capa de aplicación” y unidad 3 “Desarrollo y Monitoreo de Aplicaciones de Red”, para ello se recomienda retomar sus apuntes del primer bimestre y prepararse para la evaluación correspondiente.
3. Visualice los REA expuestos en el plan docente de los temas abordados.
4. Recuerde, si no alcanzó a participar de la actividad síncrona, está a tiempo de recuperar la misma desarrollando la actividad suplementaria.

¡Felicitaciones por finalizar con éxito el primer bimestre de estudio!

Quiero reconocer su dedicación y esfuerzo al completar satisfactoriamente este primer periodo académico. Ha demostrado un gran compromiso y perseverancia al enfrentar los retos académicos, y eso merece un reconocimiento especial.

¡Le deseo mucha suerte en su próximo examen bimestral! Confíe en sus habilidades y tenga la seguridad de que ha hecho todo lo necesario para destacar en su desempeño.

¡Felicidades nuevamente!

Segundo bimestre

Resultado de aprendizaje 2, 3 y 4 :

- Diseña aplicaciones de red orientada a datos
- Describe el funcionamiento de redes multimedia y de tiempo real.
- Esquematiza estrategias de seguridad básica en redes de computadoras.

Los resultados de aprendizaje propuestos están orientados a desarrollar habilidades en torno al diseño de aplicaciones de red orientadas a datos, la gestión de redes multimedia, y la seguridad en redes de computadoras. Los temas sobre redes en la nube, opciones de conectividad y configuración de recursos en entornos virtuales (Unidad 4) proporcionan a los estudiantes las herramientas para crear arquitecturas que soportan aplicaciones modernas de procesamiento de datos, adaptadas a entornos empresariales y servicios en la nube.

La Unidad 5 aborda fundamentos de seguridad, supervisión de redes, firewalls, y protocolos seguros como IPsec y VPNs. Estos contenidos capacitan al estudiante para identificar vulnerabilidades, implementar medidas de seguridad, y gestionar redes seguras en distintos escenarios tecnológicos.

En la Unidad 6 se exploran tecnologías y protocolos multimedia esenciales, como HLS, RTMP y WebRTC, así como redes de distribución de contenido (CDN). Estos conceptos son fundamentales para el soporte de aplicaciones que requieren transmisión en tiempo real o en vivo, como plataformas de streaming y videoconferencias.

A través del desarrollo de estos temas, usted logrará una comprensión integrada de las redes de computadoras, preparándose para enfrentar desafíos en aplicaciones de red, multimedia y seguridad informática.

Contenidos, recursos y actividades de aprendizaje recomendadas

Recuerde revisar de manera paralela los contenidos con las actividades de aprendizaje recomendadas y actividades de aprendizaje evaluadas.

Semana 9

¡Bienvenido al segundo bimestre de estudio! En esta etapa nos centraremos en el diseño, la implementación y la seguridad de infraestructuras de red modernas, abarcando tanto aplicaciones en la nube como redes multimedia. Aprenderá a configurar recursos de red en la nube, gestionar opciones de conectividad y aplicar estrategias de diseño para redes orientadas a datos. Además, profundizaremos en la importancia de la seguridad en redes de computadoras, incluyendo la implementación de cortafuegos, VPNs e IPSec, y la supervisión constante para garantizar un entorno protegido.

Un enfoque relevante de este bimestre será el análisis y la comprensión de redes multimedia, incluyendo los protocolos de transmisión de video (HLS, RTMP, WebRTC), la adaptación de calidad de servicio (QoS) y el uso de redes de distribución de contenido (CDN) para mejorar la eficiencia y experiencia de usuario en aplicaciones que demandan alto rendimiento.

Para complementar su aprendizaje, contará con recursos didácticos diversos, incluyendo prácticas de laboratorio, estudios de caso como el de Netflix y actividades interactivas. También participará en proyectos colaborativos, desarrollo de infografías y sesiones síncronas de estudio. Estas actividades tienen como objetivo profundizar en el contenido teórico, fomentar el trabajo en equipo y facilitar la resolución de dudas en un entorno participativo.

Le animo a sumergirse plenamente en esta experiencia de aprendizaje. Aproveche los recursos, participe activamente en cada actividad y consulte a su docente para reforzar sus conocimientos. Todo lo que aprenda en este bimestre será fundamental para su desarrollo profesional en el ámbito de las redes, la computación en la nube y la seguridad digital. ¡Éxitos en esta etapa de su formación!

Unidad 4. Diseño e Implementación de Redes en la Nube para Aplicaciones de Datos

Esta unidad tiene como objetivo profundizar en la arquitectura de redes en entornos de nube, explorando conceptos esenciales para diseñar y gestionar aplicaciones de red orientadas a datos. A lo largo de esta unidad, comprenderá los componentes, topologías y servicios fundamentales que ofrecen los proveedores de nube, así como las estrategias de seguridad, monitoreo y escalabilidad necesarias para mantener un rendimiento óptimo. Este enfoque práctico le permitirá adquirir habilidades clave para crear entornos de red eficientes y seguros, adaptados a las necesidades de aplicaciones modernas.

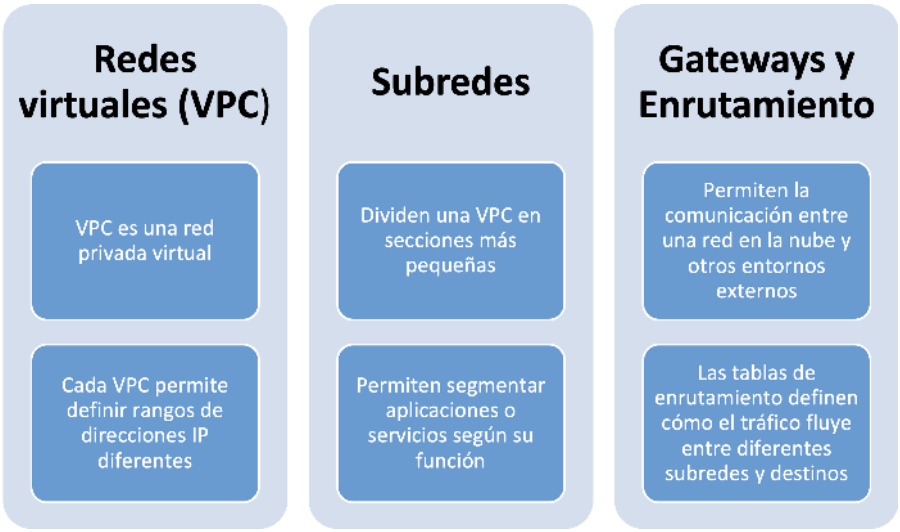
4.1 Conceptos básicos de redes en la nube

Las redes en la nube permiten a las organizaciones diseñar, implementar y administrar aplicaciones y servicios sin la necesidad de infraestructura física dedicada. Este modelo aprovecha recursos virtualizados y gestionados por proveedores de servicios en la nube como AWS, Azure y Google Cloud Platform, entre otros, garantizando flexibilidad, escalabilidad y seguridad (Dutt, 2020).

La Figura 22 representa los conceptos esenciales de las redes en la nube:

Figura 22

Conceptos básicos de redes en la nube



Nota. Adaptado de *Opciones de Conectividad de Red A Amazon VPC* [Ilustración], por Amazon Virtual Private Cloud, s. f., AWS, CC BY 4.0.

Adicional a lo indicado en la Figura 22, existen conceptos complementarios relacionados a la seguridad y escalabilidad, a continuación, la definición desde la perspectiva de redes en la nube.

- **Seguridad en la Nube, l**os grupos de seguridad y listas de control de acceso (ACL) son herramientas que permiten controlar qué tipos de tráfico (entrante o saliente) pueden acceder a los recursos de la red. Además, las soluciones en la nube suelen incluir opciones avanzadas de monitoreo y protección contra amenazas.
- **Escalabilidad y Alta Disponibilidad, l**as redes en la nube permiten el escalado dinámico de recursos según la demanda, así como la implementación de arquitecturas redundantes en múltiples regiones y zonas de disponibilidad, garantizando la continuidad del servicio frente a fallos.

Funcionamiento de las redes en la nube

En el pasado y, algunas actualmente, las empresas alquilan/implementan servidores/aplicaciones en centros de datos locales y utilizan enlaces de red directa proporcionados por operadores de telecomunicaciones. Hoy en día, las redes en la nube permiten que las empresas utilicen servidores virtuales en diferentes regiones y establezcan conexiones seguras a través de redes privadas virtuales (VPN) y gateways basados en la nube.

Las redes en la nube consisten en componentes, topologías y configuraciones virtuales que operan sobre la infraestructura física de un proveedor de servicios en la nube. Estas redes se gestionan y definen mediante software, permitiendo a las organizaciones crear redes virtuales de área local (LAN) y de área extendida (WAN) con recursos en la nube (*Opciones de Conectividad de Red A Amazon VPC - Opciones de Conectividad de Amazon Virtual Private Cloud*, s. f.).

Virtualización

La virtualización es el fundamento que hace posible los servicios de redes en la nube. A través del software, es posible definir y gestionar elementos de red como enrutadores, firewalls, equilibradores de carga virtuales e incluso topologías completas. Esta capacidad permite crear redes escalables y flexibles, donde los límites están determinados por la infraestructura física subyacente del proveedor de servicios.

Por ejemplo, un proveedor de nube puede dividir un cable de red de alta capacidad en múltiples enlaces de redes privadas virtuales, aislados lógicamente por software. Estos enlaces se pueden personalizar para ofrecer distintas capacidades según el paquete de servicios contratado, optimizando recursos según la demanda

Nubes privadas virtuales (VPC)

La integración de recursos de red con otros servicios en la nube se conoce como nube privada virtual (VPC). En una VPC, una organización puede definir una red virtual privada dentro de un entorno en la nube, manteniendo sus recursos aislados lógicamente del resto de la infraestructura pública. A través de una VPN y una puerta de enlace definida por software, se permite el acceso remoto seguro.

Por ejemplo, una empresa puede establecer una subred virtual en una región y zona de disponibilidad específicas de cualquier proveedor de nube. En esta subred, pueden ejecutarse instancias de servidores virtuales, bases de datos y otros servicios, permitiendo la comunicación segura entre diferentes subredes y recursos en distintas zonas geográficas.

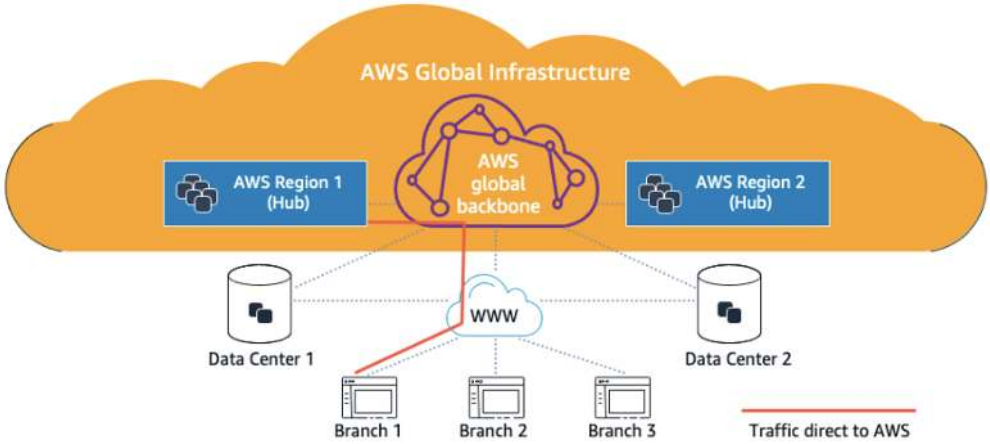
Redes en la nube híbridas

No todos los componentes de una red en la nube deben estar basados completamente en entornos virtuales. Las redes en la nube híbridas permiten la integración de infraestructuras locales (on-premise) con recursos en la nube. Esta combinación se realiza mediante conexiones a través de redes tradicionales o mediante la red pública de internet, posibilitando una interoperabilidad eficiente entre ambos entornos.

Para ofrecer una perspectiva integral de la conectividad en la nube, la Figura 23 muestra la infraestructura global de un proveedor de servicios en la nube, destacando la interconexión entre regiones, centros de datos y sucursales remotas.

Figura 23

Infraestructura de red global de proveedor de servicios de nube (caso AWS)



Nota. Tomado de *¿Qué Son las Redes En la Nube? - Explicación Sobre las Redes En la Nube* [Ilustración], por AWS, s. f., AWS, CC BY 4.0.

La figura ilustra la infraestructura global de red de un proveedor de servicios en la nube (en este caso AWS, aunque puede aplicarse a otros proveedores). A continuación, se detallan los elementos y la relación entre ellos:

1. **Infraestructura Global:** Representada por la nube naranja, simboliza la red principal que conecta regiones de la nube en distintas partes del mundo a través de un backbone de alta capacidad y baja latencia.
2. **Regiones y Hubs**: Dentro de la nube, se muestran dos regiones distintas, denominadas "AWS Region 1 (Hub)" y "AWS Region 2 (Hub)", que funcionan como centros regionales donde se alojan y gestionan servicios y recursos.
3. **Centros de Datos Locales (Data Centers):** Las ubicaciones físicas locales están representadas como "Data Center 1" y "Data Center 2", conectados a la infraestructura global de la nube mediante redes privadas o públicas.
4. **Sucursales (Branches):** Varias oficinas o sucursales ("Branch 1", "Branch 2", "Branch 3") están interconectadas con la infraestructura a través de redes públicas (WWW) o mediante conexiones directas. Estas sucursales pueden acceder a recursos distribuidos en múltiples regiones de la nube.
5. **Conexiones directas:** Las líneas rojas indican el tráfico directo entre las sucursales y los recursos alojados en la infraestructura global. Esto implica

que la comunicación puede ser optimizada mediante enlaces privados y seguros, mejorando el rendimiento de las aplicaciones críticas.

4.2 Opciones de conectividad en la Nube

Según *Opciones de Conectividad de Red A Amazon VPC - Opciones de Conectividad de Amazon Virtual Private Cloud* (s. f.), los proveedores de servicios en la nube pública ofrecen diversas opciones de conectividad que permiten a las organizaciones integrar sus recursos en la nube con sus centros de datos locales (*on-premise*), garantizando seguridad, rendimiento y escalabilidad. A continuación, se describen las principales alternativas.

1. **Conexión mediante Internet Pública**

   Es la opción más accesible y rápida de configurar. Los recursos en la nube se conectan a la red pública a través de direcciones IP accesibles globalmente. Aunque sencilla, esta opción requiere medidas de seguridad adicionales, como VPNs y cifrado, para proteger la transmisión de datos.

   - **Ventaja:** Configuración rápida y bajo costo inicial.
   - **Desafío:** Vulnerabilidad a posibles interrupciones y ataques de red si no se implementan controles de seguridad adecuados.

2. **Red Privada Virtual (VPN) sobre Internet**

   Una VPN establece un túnel cifrado entre el centro de datos local y la nube. Los datos se transmiten de manera segura sobre la red pública, protegiéndolos de interceptaciones.

   - **Aplicación común:** Para conexiones seguras temporales o de bajo a moderado volumen de tráfico.
   - **Proveedores relevantes:** AWS VPN, Azure VPN Gateway, Google Cloud VPN.

3. **Conexión Directa o Enlace Dedicado**

Proveedores de servicios en la nube ofrecen enlaces físicos dedicados para una conectividad directa entre la infraestructura on-premise y la nube. Esto elimina el uso de Internet, mejorando la latencia, la seguridad y el rendimiento.

- **Ventaja:** Mayor estabilidad, velocidad y seguridad en la conexión.
- **Proveedores relevantes:** AWS Direct Connect, Azure ExpressRoute, Google Cloud Interconnect.

4. **Red Híbrida con SD-WAN**

La tecnología de red definida por software (SD-WAN) permite a las empresas gestionar múltiples tipos de conexiones (Internet, MPLS, VPN) a través de una única interfaz. Las soluciones SD-WAN optimizan el enrutamiento del tráfico entre la nube y el centro de datos local.

- **Beneficio**: Flexibilidad para equilibrar el tráfico en diferentes rutas según criterios de rendimiento.
- **Uso típico:** Empresas con múltiples sucursales y servicios distribuidos.

5. **Conexiones Multicloud**

Empresas con estrategias multicloud pueden integrar recursos entre diferentes proveedores mediante redes interconectadas, asegurando la comunicación fluida entre varias plataformas.

- **Proveedores relevantes**: AWS Transit Gateway, Azure Virtual WAN, Google Cloud Network Connectivity Center.
- **Caso de uso**: Organizaciones que combinan servicios especializados de varios proveedores para maximizar beneficios.

Para reforzar los conceptos sobre opciones de conectividad entre la nube y un data center on-premise, le recomiendo explorar la documentación oficial de varios proveedores. Por ejemplo:

- **AWS:** Proporciona información sobre conexiones híbridas a través de AWS Direct Connect y VPN, disponible en AWS Direct Connect.

- **Microsoft Azure:** Explica las opciones de conectividad híbrida como Azure ExpressRoute en Azure ExpressRoute.
- **Google Cloud Platform (GCP):** Detalla las soluciones híbridas como Interconnect y VPN en Google Cloud Interconnect.

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Investigue la infraestructura global de los principales proveedores de nube pública (AWS, Azure, GCP). Identifique y describa los conceptos de región, zonas de disponibilidad y cómo estos afectan la disponibilidad, latencia y escalabilidad de los servicios en la nube.
2. Investigue un caso de uso donde se haya implementado una conexión entre un data center on-premise y una nube pública. Analice las necesidades de negocio, los retos técnicos y las soluciones aplicadas para lograr una conectividad eficiente, segura y escalable.
3. Participe en el siguiente juego de arrastrar y soltar donde seleccionará la opción más adecuada de tipo de conexión (VPN, enlace dedicado, conexiones redundantes) para distintos escenarios de conectividad entre la nube y un data center on-premise. Esta actividad le ayudará a aplicar los conceptos revisados en la sección 4.2.

Opciones de Conexión para la Nube

Estas actividades le permitirán explorar y aplicar los conceptos clave de redes en la nube, como la infraestructura global de los proveedores, las regiones, zonas de disponibilidad y las opciones de conectividad. De este modo, fortalecerá su comprensión de

cómo se diseñan, estructuran y operan las redes en la nube para garantizar escalabilidad, seguridad y comunicación eficiente entre recursos locales y en la nube

¡Felicitaciones por culminar la primera semana de estudio del segundo bimestre! Ha comenzado con gran determinación y se ha esforzado para alcanzar sus metas académicas. ¡Siga así y continúe con ese excelente trabajo en las próximas semanas!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 10

Unidad 4. Diseño e Implementación de Redes en la Nube para Aplicaciones de Datos

Una nueva semana de aprendizaje comienza, y en esta ocasión se adentrará en la configuración y gestión de redes virtuales en la nube. Explorará conceptos clave como la creación de VPC, subredes, gateways y mecanismos de seguridad, esenciales para diseñar y administrar infraestructuras de red híbridas. Este conocimiento le permitirá comprender cómo estructurar conexiones seguras y eficientes entre recursos en la nube y centros de datos locales. ¡Es hora de fortalecer sus habilidades en redes para aplicaciones de datos!

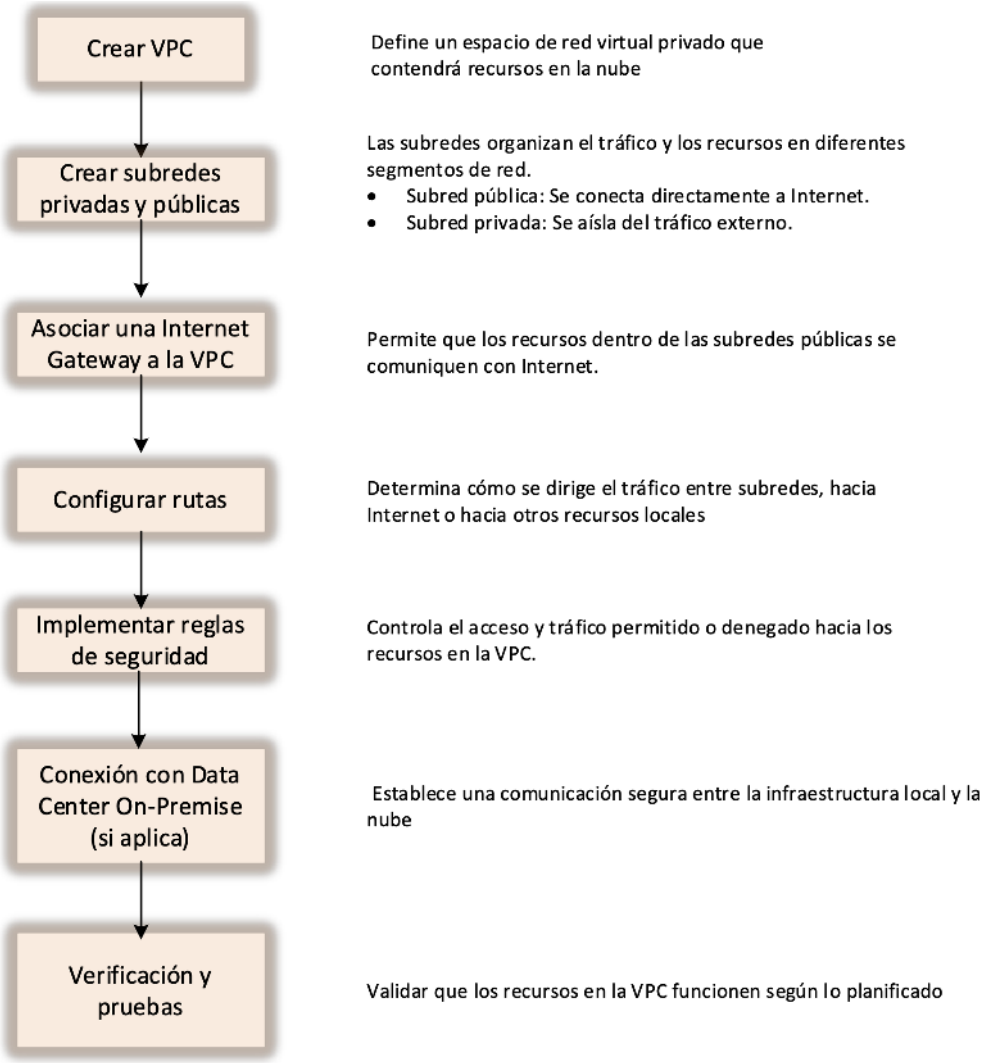
4.3 Configuración de Redes Virtuales en la Nube

Las redes virtuales en la nube permiten segmentar y controlar los recursos conectados de forma segura. A continuación, se detallan los elementos esenciales en la configuración de una VPC (Virtual Private Cloud) con base en los servicios de nube pública (Dutt, 2020).

A continuación, la Figura 24 ilustra el flujo de actividades, desde la definición inicial de la red hasta la configuración de sus principales componentes, como subredes, gateways de Internet, rutas y reglas de seguridad. Este esquema tiene como propósito facilitar la comprensión del proceso, permitiendo visualizar cómo cada paso contribuye a una infraestructura de red en la nube eficiente y segura.

Figura 24

Creación de Virtual Private Cloud (VPC)



Nota. Tomado de *¿Qué Son las Redes En la Nube? - Explicación Sobre las Redes En la Nube* [Ilustración], por AWS, s. f., AWS, CC BY 4.0.

El flujo para la creación de una VPC se compone de etapas clave que permiten estructurar una red en la nube desde cero. Inicia con la definición del espacio de direcciones IP (CIDR), lo que establece el rango de direcciones que la red podrá asignar. Posteriormente, se crean subredes públicas y privadas para

segmentar el tráfico según su propósito. A continuación, se configuran componentes esenciales como el gateway de Internet para la conectividad externa, tablas de rutas para definir el flujo de tráfico y reglas de seguridad que protegen los recursos. Este proceso asegura una red escalable y segura, adaptable a las necesidades de una infraestructura en la nube.

4.4 Práctica de laboratorio: Creación y configuración de recursos de red en la nube

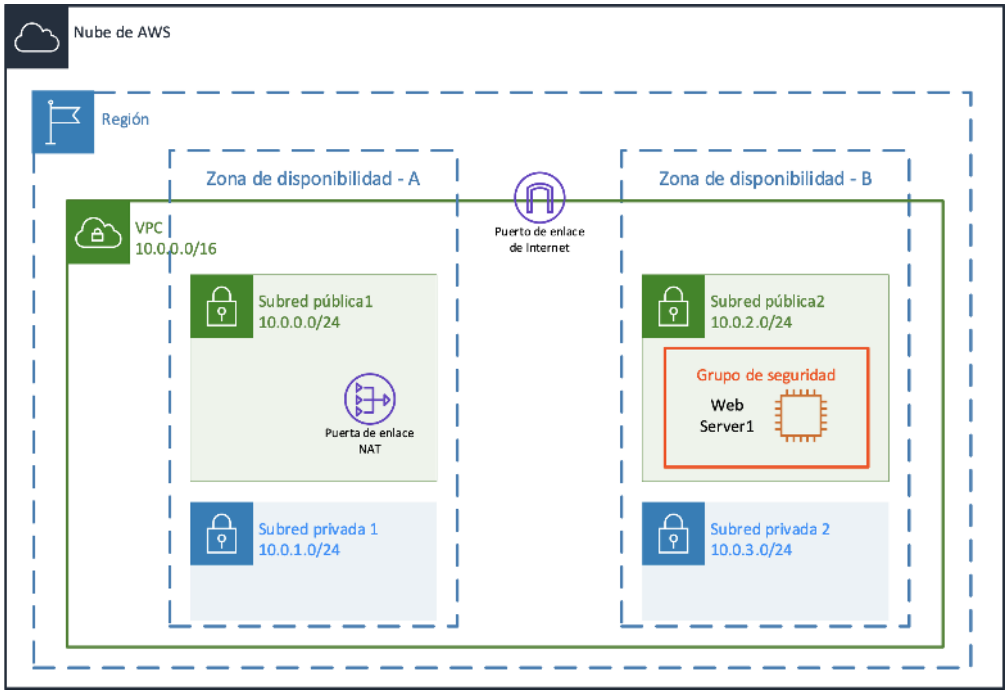
En este laboratorio, con el apoyo de su docente, usted creará una red virtual privada personalizada utilizando servicios de red en la nube ofrecidos por un proveedor público. Comenzará por definir una red privada virtual (VPC), incluyendo la configuración de componentes esenciales como subredes públicas y privadas, gateways de Internet y tablas de rutas. También configurará un grupo de seguridad que controlará el acceso a los recursos dentro de la red. Posteriormente, desplegará una instancia virtual, la cual será configurada para ejecutar un servidor web. Esta instancia se lanzará dentro de la subred definida, permitiendo así la comprensión y aplicación práctica de los conceptos y configuraciones estudiados

Escenario

La Figura 25 indica el escenario a implementar en este laboratorio.

Figura 25

Escenario de práctica de laboratorio – creación de VPCs



Nota. Tomado de *¿Qué Son las Redes En la Nube? - Explicación Sobre las Redes En la Nube* [Ilustración], por AWS, s. f., AWS, CC BY 4.0.

La Figura 25 representa una red privada virtual (VPC) distribuida en dos zonas de disponibilidad (A y B) dentro de una región específica de la nube. El estudiante configurará los siguientes componentes:

1. **VPC (10.0.0.0/16):** La red principal que contiene los recursos de la infraestructura.
2. **Subredes:**
    - *Subred pública 1* (10.0.0.0/24) en la zona de disponibilidad A, con un gateway NAT que permite a los recursos internos acceder a Internet sin ser directamente accesibles desde el exterior.
    - *Subred privada 1* (10.0.1.0/24) en la misma zona de disponibilidad A, diseñada para alojar recursos que no necesitan exposición pública.

- *Subred pública 2* (10.0.2.0/24) en la zona de disponibilidad B, con acceso controlado mediante un grupo de seguridad.
- *Subred privada 2* (10.0.3.0/24) en la zona B, que alojará recursos internos.

3. **Puerta de enlace de Internet:** Permite la comunicación entre las subredes públicas y recursos externos en la web.
4. **Grupo de seguridad:** Configurado para definir las reglas de acceso a una instancia de servidor web que se desplegará en la subred pública 2.
5. **Instancia de servidor web (Web Server1):** Implementada en la subred pública, será accesible a través de Internet mediante las configuraciones de seguridad establecidas.

Durante la ejecución de este laboratorio, el docente estará disponible para brindar apoyo, aclarar dudas y guiar en cada paso del proceso. Consulte cualquier aspecto que considere necesario para una mejor comprensión o para asegurar el correcto desarrollo de la configuración. Aproveche para adquirir experiencia práctica en la gestión de redes en la nube, ¡estamos aquí para ayudarle!

¡Excelente trabajo! Ha completado el estudio de la cuarta unidad. Ahora es momento de avanzar a la unidad 5. Pero antes de hacerlo, le invito a realizar las siguientes actividades para reforzar y aplicar los conocimientos adquiridos:

Actividades de aprendizaje recomendadas

1. Investigue y explore cómo se crean redes virtuales en al menos dos proveedores de nube pública (por ejemplo, Azure y GCP). Compare las diferencias y similitudes en el proceso, prestando atención a la creación de subredes, puertas de enlace de Internet y grupos de seguridad.
2. Elabore un informe que explique la base tecnológica de la creación de redes virtuales, conocida como virtualización de

redes. Analice conceptos clave como enrutadores virtuales, firewalls virtuales y la segmentación lógica de recursos en la nube. Identifique ventajas y desafíos de esta tecnología en la administración de redes.

Nota: Por favor complete las actividades en un cuaderno o documento Word

3. Explore y revise sobre "AWS Cloud Practitioner Essentials" disponible en la plataforma de formación de Amazon Web Services (AWS). Este curso le proporcionará una visión general sobre conceptos fundamentales de computación en la nube, como la infraestructura, modelos de servicio, opciones de conectividad, seguridad en la nube y redes virtuales. Además, es ideal para consolidar su comprensión de los temas relacionados con la configuración de redes en la nube, opciones de conectividad y estrategias de implementación de redes virtuales privadas (VPC).

   Estas actividades le permitirán afianzar los conocimientos adquiridos sobre la creación y gestión de redes en la nube, así como comprender los fundamentos técnicos que las hacen posibles.

   ¡Felicitaciones por haber culminado la Unidad 4: Diseño e Implementación de Redes en la Nube para Aplicaciones de Datos! Su esfuerzo y dedicación en el aprendizaje sobre redes en la nube le han permitido construir una sólida base de conocimientos en este importante ámbito de la tecnología. Le animo a seguir explorando y aplicando lo aprendido para dominar aún más este fascinante tema.

4. Antes de continuar, le invito a poner a prueba sus conocimientos con la autoevaluación 4. Este cuestionario es una excelente oportunidad para consolidar conceptos, evaluar su comprensión y reflexionar sobre los temas clave de la

unidad. Recuerde que los errores son valiosas oportunidades para el aprendizaje, así que enfrente este desafío con actitud positiva y confianza.

¡Siga adelante y continúe demostrando todo su potencial en este camino hacia la excelencia en redes en la nube!

Autoevaluación 4

Lea atentamente las preguntas propuestas en relación con el diseño e implementación de redes en la nube para aplicaciones de datos y, seleccione la opción de respuesta correcta.

1. ¿Cuál es una característica principal de las redes en la nube?

   A. Pueden configurarse y gestionarse mediante software.
   B. No admiten componentes virtuales.
   C. No pueden conectarse a redes locales.

2. ¿Cuál de las siguientes opciones es un método común para conectar una red en la nube con un centro de datos on-premise?

   A. Protocolo sin estado (stateless).
   B. Conexión directa privada.
   C. VPN segura.

3. ¿Qué tipo de red permite aislar recursos dentro de una nube pública?

   A. Red privada virtual (VPC).
   B. Red de acceso abierto.
   C. Red de prueba.

4. ¿Cuál es un componente esencial en la configuración de una nube virtual privada?

   A. Subredes públicas y privadas.
   B. Servicio FTP compartido.
   C. Conexión Wi-Fi pública.

5. ¿Qué es una puerta de enlace de Internet en el contexto de una VPC?

   A. Un componente virtual que permite el acceso a Internet desde una VPC.
   B. Un dispositivo físico conectado al proveedor de servicios.
   C. Un servicio que restringe todo acceso a la red.

6. ¿Cuál es una ventaja del uso de subredes privadas en la nube?

   A. Permiten el acceso sin autenticación.
   B. Proporcionan acceso directo a Internet.
   C. Aumentan la seguridad al restringir el acceso desde fuera de la red.

7. ¿Qué recurso se utiliza para aplicar políticas de acceso en una VPC?

   A. Grupo de seguridad.
   B. Puerta de enlace NAT.
   C. Protocolo UDP.

8. ¿Qué componente se necesita para habilitar la comunicación de salida desde una subred privada hacia Internet?

   A. Puerta de enlace NAT.
   B. Redirección directa.
   C. Firewall físico.

9. ¿Qué implica la configuración de una instancia de servidor web en una subred pública de una VPC?

   A. Que la instancia será accesible solo desde la red interna.
   B. Que la instancia podrá ser accesible públicamente si tiene configuradas las reglas de seguridad adecuadas.
   C. Que la instancia está protegida automáticamente sin necesidad de configuración.

10. ¿Qué tarea debe realizarse para completar el laboratorio de configuración de recursos de red en la nube?

    A. Crear una VPC, subredes y configurar un servidor web.
    B. Ejecutar una instancia sin definir políticas de acceso.
    C. Omitir la configuración de grupos de seguridad.

Ir al solucionario

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño. En caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje.

Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 11

¡Bienvenido a la semana 11 de estudio! En esta nueva etapa, exploraremos el apasionante mundo de la seguridad en las redes de datos, un aspecto crucial en la era digital. A lo largo de estas semanas, se adentrará en conceptos fundamentales que le permitirán comprender cómo proteger las comunicaciones, prevenir intrusiones y garantizar la integridad de la información.

Unidad 5. Seguridad en las redes de computadoras

5.1 Fundamentos de la seguridad en la red

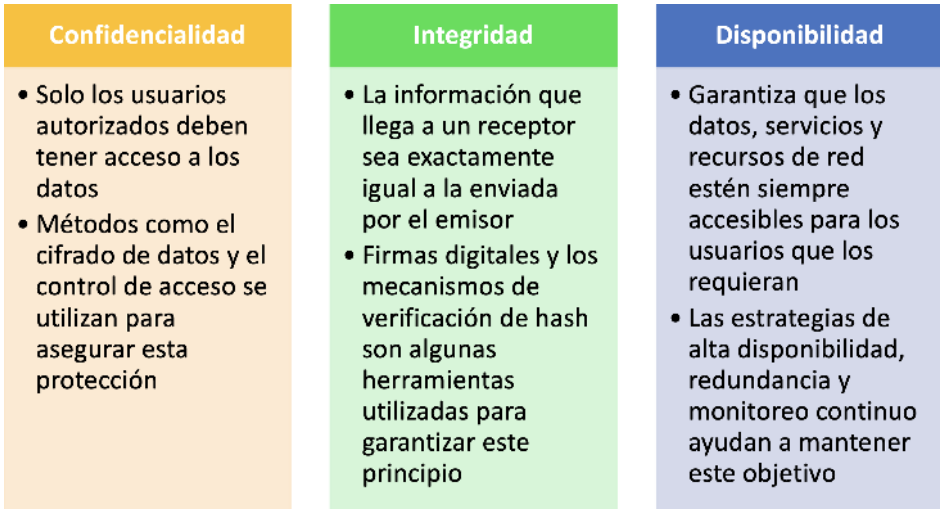
La seguridad en redes es una rama de la informática dedicada a proteger los distintos componentes de una red de computadoras. Su propósito es prevenir accesos no autorizados, el robo de datos, el uso indebido de conexiones, la modificación de información y otros riesgos similares. El objetivo principal es implementar métodos y mecanismos de defensa proactiva para proteger la red frente a amenazas tanto internas como externas (Siddiqui, 2020).

Objetivos fundamentales de la seguridad en redes

La seguridad en redes se fundamenta en tres pilares esenciales que aseguran la protección de los datos, la integridad de la información y la disponibilidad de los servicios. Estos pilares se ilustran en la Figura 26 (Kurose & Ross, 2017).

Figura 26

Pilares de la seguridad en redes



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Como se muestra en la Figura 26, la interacción de estos tres principios es clave para una arquitectura de red segura. Sin confidencialidad, los datos podrían ser expuestos; sin integridad, la información se vuelve poco confiable; y sin disponibilidad, los servicios quedarían inoperativos. Por ello, estos objetivos trabajan juntos para proteger la infraestructura de comunicación en un entorno digital cada vez más amenazado.

Códigos maliciosos: tipos y características

Para Siddiqui (2020), los códigos maliciosos, conocidos como malware, son herramientas empleadas por hackers para dañar, espiar o interferir en sistemas de red. Entre los tipos más comunes se encuentran:

- **Virus**: son programas que se adhieren a otro software con el objetivo de realizar funciones no deseadas. Generalmente requieren que el usuario los active, aunque también pueden permanecer en modo inactivo durante largos periodos para evitar ser detectados.

- **Gusanos**: a diferencia de los virus, los gusanos son programas independientes que se propagan automáticamente, explotando vulnerabilidades conocidas. No requieren la intervención del usuario y se replican para infectar otros dispositivos, ralentizando la red.
- **Spyware:** este software espía recopila información personal sin permiso del usuario. Se utiliza para monitorear la actividad de navegación, influir en las decisiones de compra y redirigir solicitudes HTTP hacia sitios de publicidad predefinidos.
- **Adware:** el adware es un software que muestra anuncios no deseados, generalmente en forma de ventanas emergentes, sin el consentimiento del usuario.
- Scareware: este tipo de software intenta engañar al usuario haciéndole creer que su sistema está infectado. Su objetivo es convencerlo de comprar programas de "seguridad" falsos.
- **Caballo de Troya (Trojan horse):** son programas que aparentan ser útiles, pero en segundo plano ejecutan acciones maliciosas, como permitir acceso no autorizado al dispositivo donde se instalan.
- **Ransomware:** diseñado para bloquear el acceso a un sistema o cifrar sus archivos, el ransomware exige el pago de un rescate para restaurar el acceso.

Estos tipos de malware representan una amenaza significativa para la seguridad de las redes. La identificación y prevención efectiva de estos códigos maliciosos es fundamental para proteger la integridad, confidencialidad y disponibilidad de los sistemas informáticos.

En este contexto, es crucial plantearnos cómo mitigar el acceso no autorizado y la afectación por códigos maliciosos. Según Sadiqui (2020), existen diversas recomendaciones para abordar estos problemas. Le invito a realizar el siguiente juego de relacionar, con el propósito de ampliar sus conocimientos y reflexionar sobre este tema.

Medidas de seguridad en redes

5.2 Supervisión de redes de computadoras

La supervisión en redes de computadoras es un aspecto fundamental para garantizar el correcto funcionamiento, la seguridad y el rendimiento de los sistemas. Esta tarea se lleva a cabo a través del plano de gestión, el cual administra el tráfico generado por protocolos de monitoreo, como Syslog (System Logging Protocol) y SNMP (estudiado en el bimestre anterior). Además, el protocolo NTP (Network Time Protocol) desempeña un papel crucial al sincronizar el tiempo en los distintos elementos de la red, asegurando así la precisión y confiabilidad de los datos recopilados durante el proceso de supervisión (Siddiqui, 2020).

Protocolo de tiempo en red (NTP)

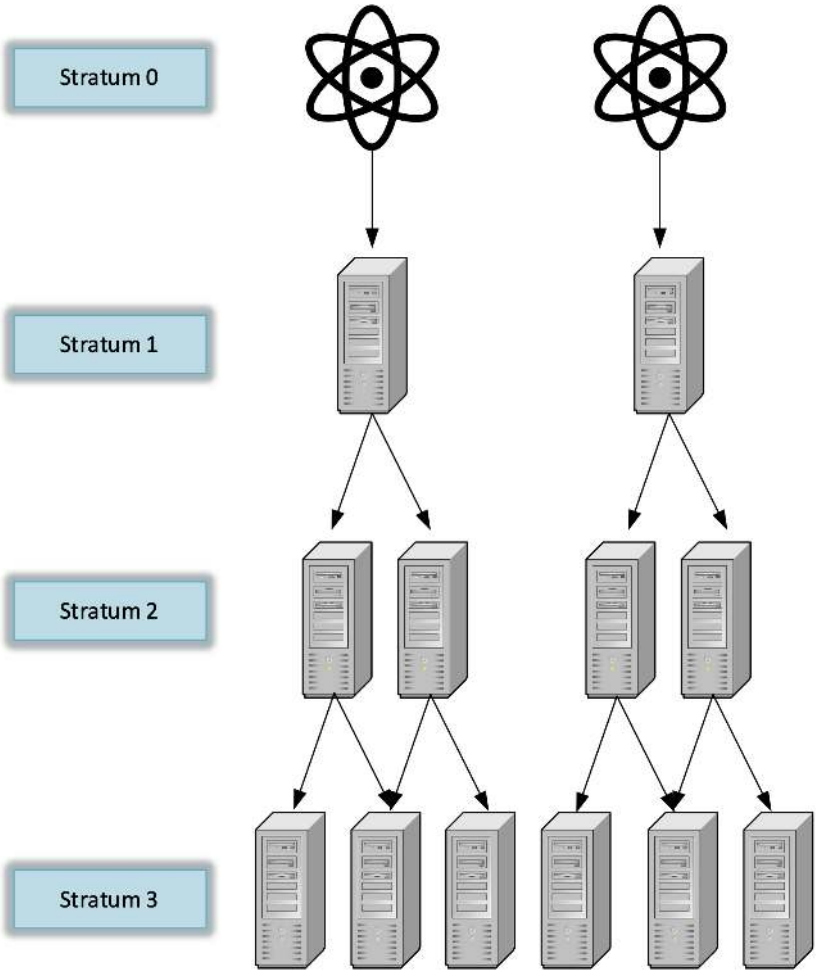
El Protocolo de Tiempo en Red (Network Time Protocol, NTP) permite sincronizar el reloj local de un componente de la red (dispositivos) con el de un servidor de referencia, que puede ser un servidor público de tiempo en Internet o una fuente de tiempo interna.

Niveles jerárquicos del NTP

El NTP opera mediante servidores organizados en diferentes niveles o estratos, según sus características. Los relojes atómicos conforman el Estrato 0 y están conectados directamente a los servidores de Estrato 1. Estos, a su vez, solo pueden ser consultados por servidores de niveles inferiores, como los de Estrato 2 o Estrato 3. Algunos de estos servidores son de acceso libre, la Figura 27 representa la estructura jerárquica de NTP.

Figura 27

Estructura jerárquica de NTP



Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

La correcta sincronización del tiempo en una red es esencial para garantizar la coherencia y precisión en la gestión de eventos y registros. El NTP desempeña un papel crucial al permitir que todos los dispositivos de la red mantengan un reloj uniforme, lo que resulta indispensable para la supervisión efectiva. Tener los eventos organizados cronológicamente permite identificar patrones, detectar anomalías y correlacionar incidentes de manera más eficiente. Sin una sincronización adecuada, los registros pueden presentar inconsistencias

temporales, dificultando el análisis forense, el monitoreo en tiempo real y la resolución de problemas. Por ello, implementar y mantener el NTP es una medida clave para la seguridad y el control operativo en redes de computadoras.

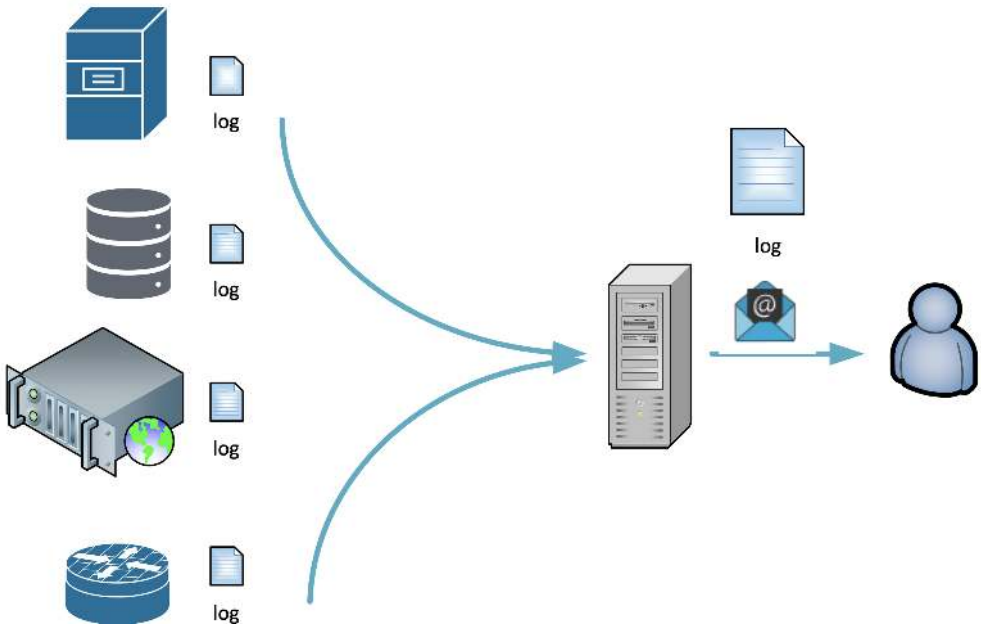
Servidor Syslog

El protocolo Syslog se utiliza para recopilar mensajes de registro (logs) generados por dispositivos o aplicaciones. Estos mensajes pueden ser, en muchos casos, la única fuente de información para identificar las causas de fallos en el equipo. Los mensajes de Syslog se pueden enviar a distintas ubicaciones:

- Consolas locales.
- Terminales remotos.
- Un servidor Syslog.

La Figura 28 muestra la arquitectura de centralización de archivos de registro en un servidor Syslog, ilustrando cómo los dispositivos de red envían sus logs para una gestión y análisis centralizado.

Figura 28
Centralización de logs en un Syslog Server



Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

Como se puede apreciar en la Figura 28, un servidor Syslog permite realizar tareas esenciales para la gestión y supervisión de la red, tales como:

- Centralizar los archivos de registro de dispositivos como routers, switches y servidores.
- Archivar los logs en una ubicación segura donde puedan ser procesados.
- Facilitar la búsqueda y clasificación de los registros, mejorando su análisis.

Niveles de severidad en Syslog

Los niveles de severidad en Syslog indican la naturaleza de los mensajes de error, clasificándolos en una escala de 0 a 7. El nivel 0 representa los errores más críticos, mientras que el nivel 7 corresponde a mensajes informativos de depuración. A continuación, la Tabla 5 detalla estos niveles:

Tabla 5
Niveles de seguridad en Syslog

Código	Severidad	Descripción
0	Emergencias	El sistema es inoperable.
1	Alertas	Se requiere una intervención inmediata.
2	Crítico	Error crítico del sistema.
3	Errores	Errores de operación.
4	Advertencias	Advertencia (podría ocurrir un error).
5	Notificaciones	Evento normal que debe ser informado.
6	Informativo	Mensaje con información relevante.
7	Depuración	Mensaje de depuración para diagnóstico.

Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

La clasificación por niveles de severidad ayuda a priorizar la atención de errores críticos, mejorando así la capacidad de respuesta ante incidentes. Sin un sistema de supervisión centralizado, el diagnóstico de fallos se vuelve más complejo, lo que aumenta el riesgo de interrupciones prolongadas en los servicios.

Protocolo Simple de Gestión de Red (SNMP)

Para Kurose & Rose (2017), el Protocolo Simple de Gestión de Red (Simple Network Management Protocol, SNMP) es una herramienta esencial que permite monitorear, diagnosticar y gestionar de forma remota los dispositivos

en una red. Este protocolo proporciona capacidades avanzadas para el mantenimiento y supervisión de la infraestructura de red, cumpliendo los siguientes objetivos:

- Monitorear el rendimiento de la red y conocer el estado general de los dispositivos, ya sea activo, inactivo, parcialmente operativo, operativo o en congestión.
- Detectar problemas en la red y gestionar eventos excepcionales, como la pérdida de un enlace de red o el fallo repentino de un equipo.
- Configurar remotamente los equipos de la red.

El SNMP opera en la capa de aplicación del modelo OSI y utiliza el protocolo UDP a través del puerto 162.

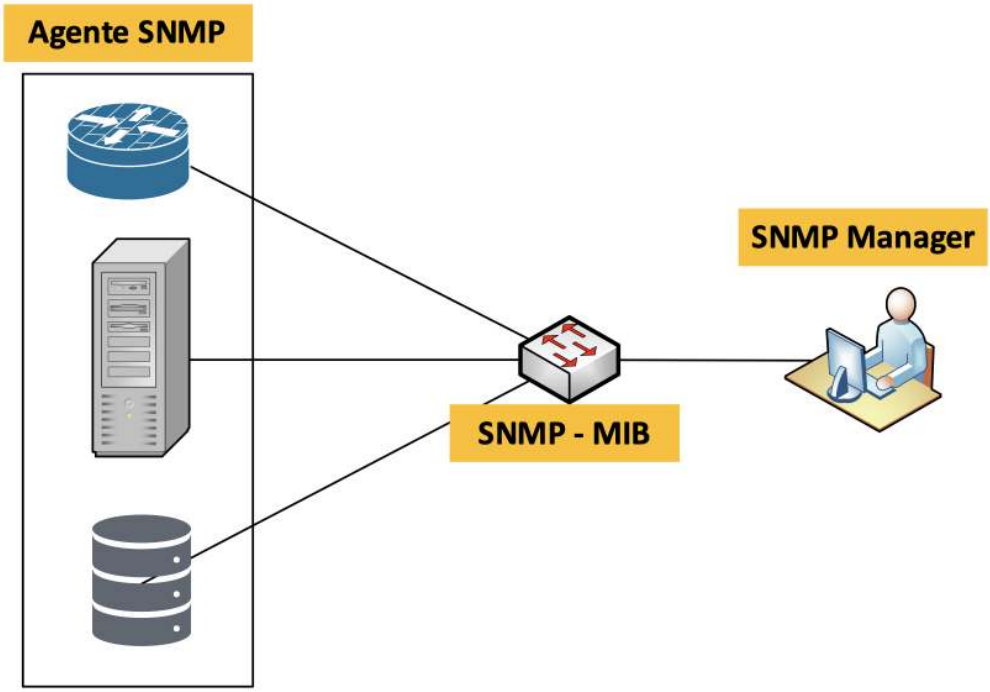
Herramientas basadas en SNMP

Existen múltiples aplicaciones de software que aprovechan las capacidades del SNMP para generar gráficos y reportes que muestran la evolución del tráfico en la red o el estado de los sistemas. Entre las herramientas más destacadas se encuentran PRTG, Centreon, NetCrunch 5, MRTG, entre otras.

Este protocolo es fundamental para una gestión eficiente de la red, ya que permite la detección temprana de fallos, optimiza la capacidad de respuesta ante incidentes y proporciona un control preciso de los recursos disponibles. Para su correcta operación, el SNMP se basa en tres componentes esenciales, los cuales se ilustran en la Figura 29.

Figura 29

Componentes de un sistema de monitoreo basado en SNMP



Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

Como se indica en la Figura 29, un sistema SNMP (Protocolo Simple de Gestión de Red) se compone de tres elementos principales:

- **Gestor SNMP:** Es un equipo que ejecuta el software de administración SNMP. Generalmente, se trata de un ordenador utilizado para supervisar y gestionar la red.
- **Agente SNMP:** Software instalado en dispositivos de red como routers, switches o servidores. Su función es permitir que estos dispositivos sean monitoreados por el gestor SNMP.
- **Base de Información de Gestión (MIB):** Es un conjunto de colecciones de objetos administrados por el agente SNMP. Estas colecciones contienen variables que el gestor SNMP puede consultar o modificar según sea necesario.

El monitoreo de redes es crucial para la seguridad, ya que permite detectar actividades inusuales, identificar vulnerabilidades y responder de manera oportuna a posibles amenazas. Al supervisar constantemente el tráfico y los dispositivos, las organizaciones pueden prevenir accesos no autorizados, minimizar riesgos y garantizar la disponibilidad y protección de los datos, manteniendo así la integridad y estabilidad de sus infraestructuras de red.

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Observe el video "Ataque de Hombre en el Medio": Realice una investigación sobre este tipo de ataque, cómo se lleva a cabo y qué medidas de seguridad pueden implementarse para mitigarlo. Reflexione sobre cómo esta vulnerabilidad afecta la seguridad en redes.

   Nota: Por favor complete la actividad en un cuaderno o documento Word.

2. Descargue e instale una herramienta de monitoreo, como PRTG Network Monitor. Configure la herramienta para supervisar el tráfico y los dispositivos de su red local, identificando posibles anomalías y generando informes sobre el estado del sistema.

Estas actividades le permitirán aplicar los conceptos aprendidos y desarrollar habilidades prácticas en la detección y respuesta ante amenazas en la red.

¡Felicitaciones por completar la semana 11 de estudio! Ha sido realmente importante explorar los fundamentos de seguridad en las redes de computadoras y la importancia

de su supervisión. Ha ampliado su conocimiento sobre cómo aprovechar al máximo estos conceptos y aplicarlos a sus necesidades.

Pero el aprendizaje no se detiene aquí. ¡La semana 12 está a la vuelta de la esquina, lista para brindarle nuevos conocimientos y desafíos! Prepárese para sumergirse en un nuevo tema emocionante que le permitirá seguir creciendo y ampliando sus habilidades en arquitectura de redes.

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 12

Unidad 5. Seguridad en las redes de computadoras

Bienvenido a la semana 12 de estudio! En esta etapa, exploraremos dos temas esenciales para la protección de las redes de datos: los cortafuegos (firewalls) y los mecanismos de seguridad basados en IPSec y VPNs. Estas tecnologías desempeñan un papel crucial en la defensa de la infraestructura de red, al permitir el control de acceso, la encriptación de información y la creación de conexiones seguras. Prepárese para profundizar en su conocimiento de estas herramientas fundamentales para la seguridad en redes. ¡Adelante!

5.3 Cortafuegos o Firewalls

Según Kurose & Rose (2917), un cortafuegos, o firewall, es un sistema de hardware o software que actúa como una barrera entre una red confiable y otra no confiable. Su propósito principal es filtrar y bloquear el tráfico no deseado, previniendo el acceso no autorizado. Para cumplir su función de manera efectiva, un firewall debe seguir estas recomendaciones:

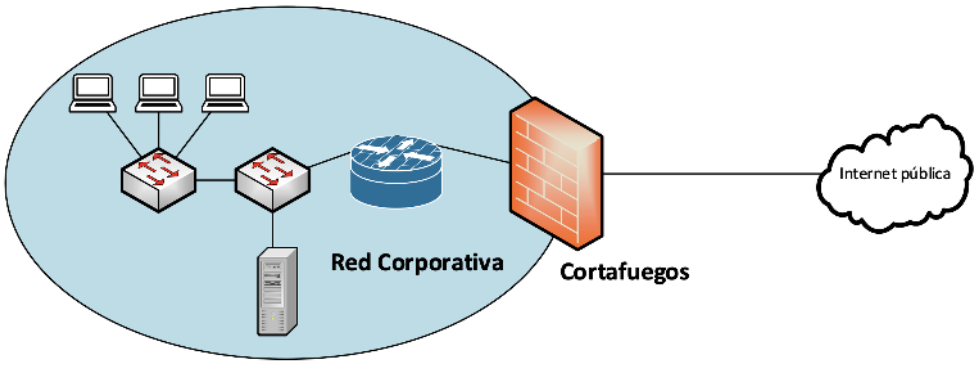
- Ser resistente a ataques.
- Servir como el único punto de tránsito entre las redes.

- Garantizar la aplicación de la estrategia de control de acceso de la organización.

La Figura 30 representa a un cortafuegos en una red institucional. En esta arquitectura, se encuentra ubicado en la zona perimetral, donde actúa como la primera línea de defensa entre la red interna de la organización y las redes externas, como Internet. Su función es filtrar el tráfico entrante y saliente, aplicando reglas de control de acceso para garantizar que únicamente el tráfico autorizado pueda atravesar la red, protegiendo así los recursos críticos de posibles amenazas externas.

Figura 30

Cortafuegos perimetral en una red corporativa



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Tipos de cortafuegos

Existen varios tipos de cortafuegos, diseñados para diferentes necesidades de seguridad:

- **Firewall NAT:** Oculta las direcciones IP privadas mediante la traducción a una dirección IP pública, proporcionando seguridad y anonimato.
- **Firewall de filtrado de paquetes:** Filtra paquetes en las capas 3 y 4 del modelo OSI, permitiendo controlar el tráfico en función de direcciones IP y

puertos. Es fácil de configurar, aunque es vulnerable a ataques de suplantación de identidad.

- **Firewall con estado (stateful):** Además del filtrado de paquetes, mantiene el estado de las conexiones, como los números de secuencia TCP o UDP, ofreciendo mayor seguridad.
- **Firewall de aplicación (proxy):** Realiza el filtrado de información en las capas 3, 4, 5 y 7, actuando generalmente como intermediario entre clientes y servidores.

Estrategias de cortafuegos

Las reglas de acceso suelen implementarse mediante listas de control de acceso (ACL). Estas reglas pueden definirse a partir de varios criterios:

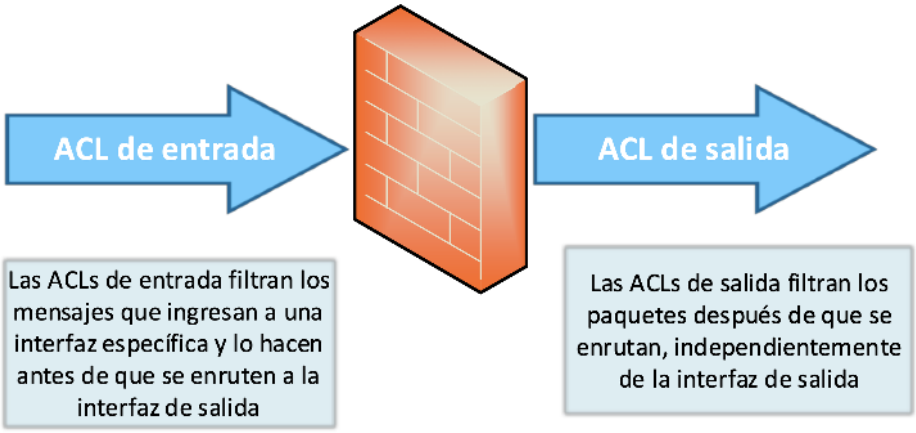
- **Reglas basadas en servicios:** Determinan los servicios que pueden ser accesibles, dependiendo de direcciones IP y números de puerto TCP.
- **Reglas basadas en dirección:** Especifican la dirección (entrada o salida) desde donde se puede iniciar o permitir tráfico hacia un servicio.
- **Reglas basadas en comportamiento:** Controlan cómo se utilizan ciertos servicios, como el filtrado de correos electrónicos para bloquear spam.

Cortafuegos basados en ACL

Las listas de control de acceso (ACL) son un conjunto secuencial de instrucciones que permiten o deniegan el acceso en función de direcciones o protocolos de capa superior. Estas listas pueden gestionar el tráfico entrante o saliente de una red; la Figura 31 representa los conceptos de ACLs entrantes y salientes.

Figura 31

Aplicación de ACLs entrantes y salientes en un cortafuegos



Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

Cortafuegos basados en zonas

Un cortafuegos basado en políticas de zonas permite proteger una red frente a amenazas externas al dividir los diferentes tipos de redes en zonas de seguridad distintas, donde se especifica el tipo de tráfico que puede fluir entre ellas. Una de las principales ventajas de este enfoque es la facilidad para definir estrategias de control de acceso (Siddiqui, *2020).*

Tipos de zonas de seguridad en una red

Generalmente, una red organizacional puede estar compuesta por tres tipos principales de zonas de seguridad:

- **Zona interna:** Corresponde a la red de producción de la organización. Esta zona requiere un acceso altamente controlado, donde no se permite acceso directo o sin restricciones desde otras zonas.
- **Zona externa:** Representa, por lo general, la conexión a Internet. Esta es una zona no segura que constituye una fuente constante de riesgo para la red interna de la organización.

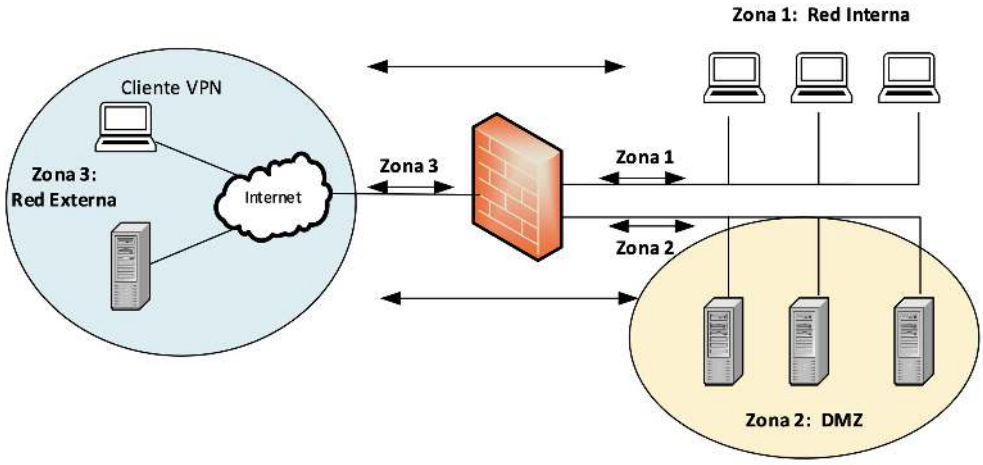
- **Zona desmilitarizada (DMZ):** Esta zona se utiliza para proporcionar acceso desde Internet a ciertos servidores (por ejemplo, servicios web o de mensajería) sin que haya una conexión directa con la red interna. Debido a los posibles riesgos de seguridad, el tráfico entre la DMZ y otras zonas debe ser estrictamente controlado:

- Tráfico entrante desde Internet hacia los hosts en la DMZ.
- Tráfico saliente de los hosts en la DMZ hacia Internet.
- Tráfico entrante desde la red interna hacia los hosts en la DMZ.
- Tráfico saliente de los hosts en la DMZ hacia la red interna.

La Figura 32 presenta el enfoque basado en zonas, proporcionando una estructura más organizada y segura, permitiendo políticas de acceso granular entre diferentes partes de la red

Figura 32

Separación de diferentes tipos de redes en distintas zonas



Nota. Adaptado de *Computer Network Security*, por Sadiqui, A., 2020.

La Figura 32 presenta al cortafuegos central como el regulador del tráfico entre las tres zonas definidas, asegurando que solo las conexiones autorizadas puedan transitar entre ellas, lo que minimiza los riesgos de acceso no autorizado o ataques externos.

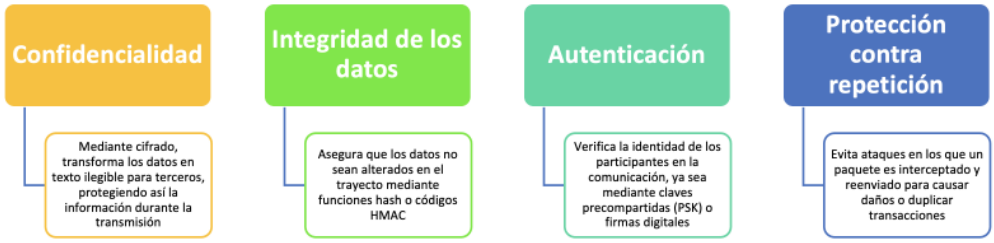
5.4 Seguridad del Protocolo de Internet (IPSec) y Redes Privadas Virtuales (VPN)

En un entorno de redes moderno, donde las organizaciones dependen de la conectividad a Internet para operaciones críticas, garantizar la seguridad de las comunicaciones es vital. Aquí es donde entra en juego IPsec (Internet Protocol Security), un conjunto de protocolos diseñados para proteger el tráfico de red mediante cifrado, autenticación y control de integridad. IPsec se utiliza ampliamente en conexiones VPN (Redes Privadas Virtuales) para proteger datos sensibles frente a amenazas internas y externas (Kurose & Rose, 2017).

IPsec tiene como finalidad proteger las comunicaciones a través de redes públicas como Internet, la Figura 33 presenta los objetivos principales de IPSec.

Figura 33

Objetivos principales de IPSec



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Figura 33 resalta la importancia de IPSec y sus elementos fundamentales, que garantizan la integridad, confidencialidad y autenticidad de las comunicaciones entre distintos puntos de la red.

Protocolos básicos de IPsec

IPsec incluye tres protocolos principales que trabajan juntos para ofrecer seguridad en las comunicaciones:

- **AH (Authentication Header):** Proporciona integridad y autenticación de origen, pero no ofrece cifrado. Se utiliza cuando la confidencialidad no es un requisito.
- **ESP (Encapsulating Security Payload):** Ofrece tanto cifrado como autenticación, lo que lo convierte en la opción más común para proteger comunicaciones sensibles.
- **IKE (Internet Key Exchange):** Este protocolo es responsable de negociar las configuraciones de seguridad antes de establecer una conexión IPsec, asegurando que ambas partes acuerden los parámetros necesarios.

Como indica Kurose & Rose (2017), una institución con presencia en varias regiones puede necesitar una red propia para que sus servidores y dispositivos intercambien datos de manera segura y confidencial. Una opción sería construir una red privada completamente independiente de Internet, con sus propios routers, enlaces y servidores DNS. Sin embargo, esta infraestructura resulta costosa, ya que implica adquirir, instalar y mantener toda la infraestructura física.

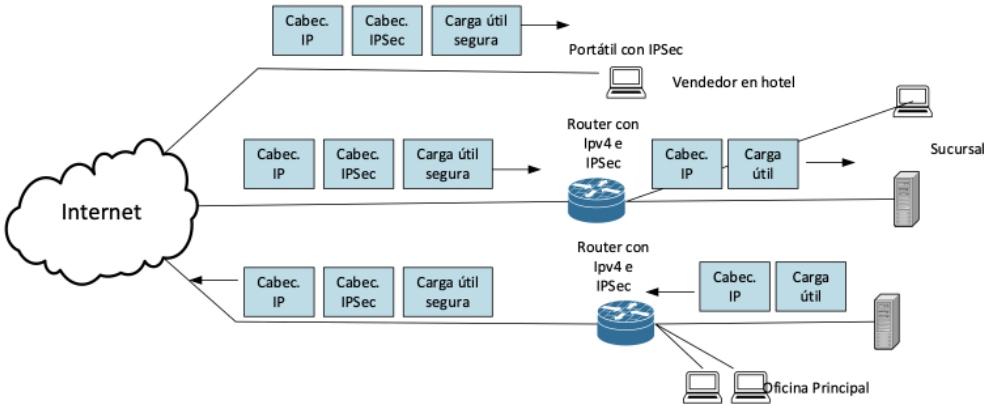
Para evitar estos gastos, muchas organizaciones optan por una Red Privada Virtual (VPN). Con una VPN, el tráfico entre las sucursales de la institución viaja a través de Internet, pero antes de hacerlo se cifra, garantizando la seguridad y confidencialidad de los datos. Así, la VPN utiliza la red pública, pero mantiene un nivel de seguridad similar al de una red privada.

Por ejemplo, imaginemos una institución con una oficina principal, una sucursal y empleados que acceden a Internet desde diferentes ubicaciones, como hoteles. Si dos equipos dentro de la misma oficina se comunican, el tráfico fluye internamente usando el protocolo IPv4 normal. Sin embargo, si un dispositivo en la oficina principal debe comunicarse con uno remoto, como el

portátil de un empleado en un hotel, el router de la oficina principal cifra el datagrama IP mediante IPsec antes de enviarlo a través de Internet, la Figura 34 representa esta interacción.

Figura 34

Red Privada Virtual con IPSec



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Este datagrama cifrado conserva una cabecera IPv4 normal, por lo que los routers en Internet lo procesan como cualquier otro datagrama. Sin embargo, dentro del datagrama IPsec se encuentra una cabecera adicional específica para la seguridad, así como la carga útil cifrada. Al llegar al destino, como el portátil del vendedor, el sistema operativo descifra los datos, verifica su integridad y luego los entrega al protocolo de la capa superior, como TCP o UDP.

Este proceso permite a las instituciones aprovechar la infraestructura de Internet pública sin comprometer la seguridad de sus comunicaciones. Aunque hemos simplificado el concepto, el uso de IPsec en VPNs es una solución clave para proteger datos en tránsito.

¡Excelente trabajo! Ha completado el estudio de la cuarta unidad. Ahora es momento de avanzar a la unidad 5. Pero antes de hacerlo, le invito a realizar las siguientes actividades para reforzar y aplicar los conocimientos adquiridos:

Actividades de aprendizaje recomendadas

1. Explorar las diferentes opciones de firewalls, enfocándose en soluciones de tipo UTM (Unified Threat Management). Deberá comparar sus características, funcionalidades y enfoques de seguridad, así como sus ventajas en entornos empresariales.
2. Investigar cuáles son los requisitos técnicos necesarios para implementar una VPN, además de los diferentes tipos de VPN existentes (como VPN de acceso remoto, VPN de sitio a sitio, y VPN con IPSec). El análisis debe incluir escenarios donde cada tipo de VPN sea el más adecuado.

   Nota. Por favor complete las actividades en un cuaderno o documento Word.

3. Acceder al "Curso Introducción a la Ciberseguridad" ofrecido por Netacad. Este curso le permitirá profundizar en conceptos esenciales de seguridad en redes, tales como amenazas cibernéticas, protección de datos, gestión de riesgos y mejores prácticas de seguridad. Este curso es ideal para complementar su comprensión de los temas revisados sobre cortafuegos, IPSec, VPNs y estrategias de protección en la infraestructura de red.

¡Felicitaciones por haber culminado la Unidad 5: Seguridad en las redes de computadoras! Su dedicación y esfuerzo para adquirir conocimientos sobre este fundamental ámbito tecnológico son dignos de reconocimiento. Ahora posee una comprensión más sólida sobre conceptos clave como cortafuegos, IPSec y VPNs, esenciales para proteger y gestionar redes de forma segura.

4. Le invito a consolidar sus aprendizajes enfrentando la autoevaluación 5, donde podrá aplicar los conocimientos adquiridos y evaluar su dominio de estos temas críticos. No tema equivocarse, recuerde que los errores son oportunidades valiosas para mejorar. Mantenga una actitud positiva y continúe avanzando con determinación en su formación.

¡Siga adelante y continúe demostrando su capacidad y compromiso!

Autoevaluación 5

Revise cuidadosamente las preguntas planteadas sobre seguridad en redes de computadoras y seleccione la opción de respuesta correcta.

1. ¿Cuál de los siguientes es un pilar fundamental de la seguridad en redes?

   A. Escalabilidad
   B. Integridad
   C. Compatibilidad

2. ¿Qué tipo de malware recopila información personal del usuario sin su consentimiento?

   A. Adware
   B. Spyware
   C. Gusano

3. ¿Cuál de los siguientes niveles en Syslog indica un error crítico del sistema?

   A. 2
   B. 6

C. 7

4. ¿Cuál es la función de un firewall basado en estados (stateful)?

   A. Filtrar solo por direcciones IP
   B. Mantener información sobre el estado de las conexiones
   C. Aumentar la velocidad de transferencia de datos

5. ¿Qué protocolo se utiliza para sincronizar el tiempo en los dispositivos de una red?

   A. FTP
   B. NTP
   C. SMTP

6. ¿Cuál de las siguientes opciones describe correctamente una VPN?

   A. Una red pública sin seguridad adicional
   B. Una conexión cifrada sobre la infraestructura de Internet
   C. Una red física dedicada

7. ¿Cuál es un protocolo básico de IPsec que proporciona cifrado y autenticación?

   A. ESP
   B. AH
   C. IKE

8. ¿Qué componente de SNMP es responsable de monitorear la red desde un equipo centralizado?

   A. Agente SNMP
   B. Gestor SNMP
   C. MIB

9. ¿Cuál es la función de la zona DMZ en una red organizacional?

A. Permitir acceso directo a la red interna
B. Aislar servidores accesibles desde Internet
C. Optimizar el rendimiento de la red interna

10. ¿Qué mecanismo utiliza IPsec para prevenir la retransmisión de paquetes capturados?

A. Cifrado asimétrico
B. Protección contra reenvíos (anti replay)
C. Limitación de ancho de banda

Ir al solucionario

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño. En caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje.

Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 13

¡Bienvenidos a la última etapa de nuestro curso! En esta unidad exploraremos las redes multimedia, esenciales para la transmisión eficiente de audio, video y otros contenidos interactivos en Internet. Durante las próximas tres semanas, analizaremos cómo las aplicaciones multimedia utilizan protocolos de red,

cómo se garantizan la calidad y el rendimiento en la transmisión de datos y la manera en que los proveedores de contenido, como Netflix, utilizan redes de distribución global para mejorar la experiencia del usuario; ¡Iniciemos!

Unidad 6. Redes multimedia

Una aplicación de red multimedia se refiere a cualquier aplicación que utiliza audio, video o ambos como parte de su funcionalidad principal. Estas aplicaciones están diseñadas para transmitir y recibir contenido en tiempo real o bajo demanda, como ocurre en plataformas de streaming, videollamadas y juegos en línea. Cada tipo de aplicación multimedia presenta un conjunto específico de requisitos técnicos, como baja latencia, ancho de banda constante y calidad de servicio (QoS), así como desafíos en su diseño e implementación.

Es esencial comprender primero las propiedades inherentes de los medios de audio y video. El audio requiere tasas de muestreo regulares y compresión para minimizar el consumo de datos sin sacrificar la calidad. Por su parte, el video, al ser más intensivo en datos, necesita algoritmos avanzados de compresión, como H.264 o VP9, y es especialmente sensible a retrasos y pérdidas de paquetes.

A medida que las aplicaciones multimedia se han vuelto parte integral de la vida digital, es crucial que las redes sean capaces de soportar estas demandas mediante tecnologías modernas, como protocolos especializados (RTP, RTMP) y garantías de calidad, lo que permite una experiencia fluida y estable para los usuarios.

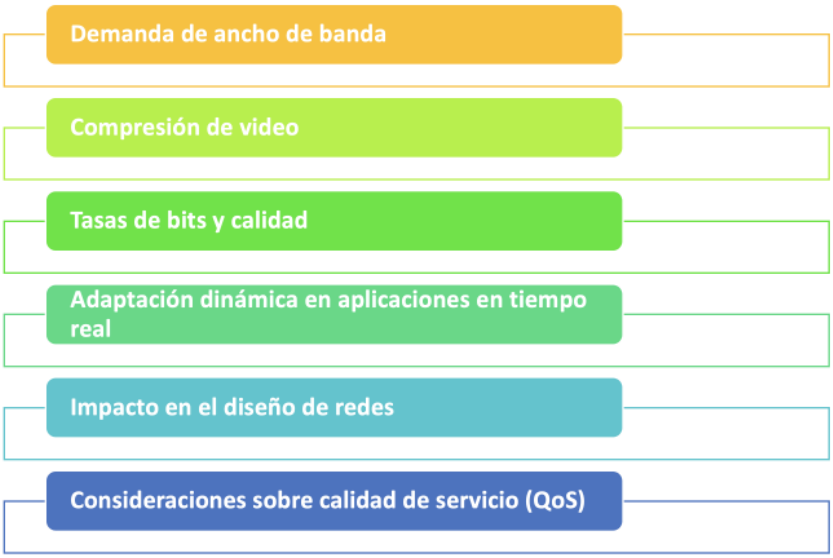
6.1 Propiedades del audio y video

Video en aplicaciones multimedia de red

Según Kurose & Rose (2017), las aplicaciones multimedia que emplean video requieren una gestión adecuada del ancho de banda debido a la gran cantidad de datos que transmiten. A continuación, la Figura 35 detalla las principales características del video en redes y cómo afectan el diseño y rendimiento de las aplicaciones multimedia.

Figura 35

Características del video en aplicaciones multimedia



Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

A continuación, detallamos cada una de las características del video en aplicaciones multimedia.

1. **Demanda de ancho de banda**

Una de las principales características del video es su alta tasa de bits. Las aplicaciones de video, dependiendo de su propósito, necesitan diferentes niveles de ancho de banda:

- **Videoconferencias de baja calidad:** Pueden funcionar con tasas de 100 kbps.
- **Transmisiones de video en alta definición:** Requieren más de 3 Mbps.

Para ilustrar la diferencia, la Tabla 6 presenta diferentes escenarios de consumo de ancho de banda (bw) por tipo de aplicación multimedia. Cada caso ilustra cómo las necesidades de datos varían dependiendo del tipo de contenido, ayudando a comprender mejor las demandas de red para aplicaciones de audio, video e imágenes (Kurose & Rose, 2017).

Tabla 6

Consumo de bw por tipo de aplicación multimedia

Tipo de Actividad	Tasa de Consumo de Datos	Explicación
Navegación en fotos	200 kbytes (1,6 megabits) cada 10 segundos	Los usuarios que navegan por fotos en redes sociales consumen datos de manera intermitente. Este tipo de actividad es moderado en términos de ancho de banda.
Streaming de música	128 kbps constante	Escuchar música en streaming tiene un flujo constante de datos. Sin embargo, el consumo es relativamente bajo en comparación con el video.
Visualización de video	2 Mbps constante	Las aplicaciones de video requieren un alto ancho de banda debido a la transmisión continua de contenido visual, lo que resulta en un consumo significativamente mayor.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

2. **Compresión de video**

El video digital sin compresión es muy voluminoso, ya que cada fotograma contiene millones de píxeles, y cada píxel almacena información de color y luminosidad. Para hacer posible la transmisión en redes, se utilizan algoritmos de compresión que reducen drásticamente la cantidad de datos sin afectar demasiado la calidad.

La compresión se basa en dos tipos de redundancia:

- **Redundancia espacial:** Dentro de un solo fotograma, áreas con información repetitiva, como grandes zonas con un mismo color, pueden comprimirse. Por ejemplo, una imagen con un fondo blanco tiene menor cantidad de información que una imagen con muchos detalles.
- **Redundancia temporal:** Ocurre entre fotogramas consecutivos. Si dos fotogramas son idénticos o similares, el sistema no necesita recodificar el segundo fotograma completo, sino que solo guarda las diferencias entre ambos.

3. **Tasas de bits y calidad**

Los algoritmos de compresión modernos permiten ajustar la tasa de bits del video. Cuanto mayor sea la tasa de bits, mejor será la calidad de imagen, aunque también se requerirá mayor capacidad de transmisión.

Las plataformas de video ofrecen versiones con diferentes tasas de bits para adaptarse a las conexiones de los usuarios:

- Por ejemplo, un video puede tener versiones comprimidas a 300 kbps, 1 Mbps y 3 Mbps.
- Los usuarios con conexiones rápidas eligen la versión a 3 Mbps, mientras que quienes acceden desde dispositivos móviles con conexiones de menor capacidad, optan por una versión de menor calidad.

4. **Adaptación dinámica en aplicaciones en tiempo real**

En aplicaciones como las videoconferencias, el video se ajusta automáticamente según el ancho de banda disponible. Este proceso de adaptación en tiempo real mejora la experiencia del usuario, ya que minimiza cortes o interrupciones; la codificación de video sobre la marcha permite ajustar la calidad en función del tráfico de red, asegurando que la comunicación fluya sin interrupciones.

5. **Impacto en el diseño de redes**

   Debido a su alto consumo de recursos, el video representa un desafío significativo para las infraestructuras de red. Las redes deben implementar mecanismos que garanticen el equilibrio entre la entrega eficiente de video y el rendimiento general de la red. Una solución común es el uso de redes de distribución de contenido (CDN), que almacenan copias de videos en servidores distribuidos geográficamente. Esto reduce la latencia y mejora la calidad de la transmisión al acercar los contenidos a los usuarios finales, este tema, lo revisaremos en mayor profundidad más adelante.

6. **Consideraciones sobre calidad de servicio (QoS)**

   Las aplicaciones de video requieren calidad de servicio para asegurar que el ancho de banda, la latencia y la pérdida de paquetes sean controlados. Las redes deben priorizar el tráfico multimedia para mantener una experiencia fluida. Sin QoS, el video puede experimentar interrupciones, retrasos o una disminución drástica en la calidad, lo que afecta negativamente a la experiencia del usuario.

Propiedades del audio en aplicaciones multimedia

El audio, a diferencia del video, requiere un menor ancho de banda para su transmisión en redes. Sin embargo, es más sensible a errores como retrasos o pérdidas de datos, los cuales afectan directamente la percepción del usuario. Por esta razón, su manejo en aplicaciones multimedia presenta tanto ventajas como desafíos específicos.

Las aplicaciones de audio, como las llamadas VoIP (por ejemplo, Zoom o Microsoft Teams) y los servicios de streaming de música, requieren un ancho de banda significativamente menor en comparación con las aplicaciones de video. Sin embargo, el audio es más sensible a los errores de transmisión, como retrasos o pérdidas de datos, lo que puede afectar notablemente la experiencia del usuario. La Tabla 7 presenta un resumen de las propiedades clave del audio en aplicaciones multimedia por Internet.

Tabla 7
Propiedades del audio en aplicaciones multimedia

Propiedad	Descripción
Menor ancho de banda requerido	Las aplicaciones de audio, como VoIP y música en streaming, necesitan entre 32 y 320 kbps, mucho menos que el video.
Mayor sensibilidad a errores	Los retrasos o pérdidas de datos en el audio se traducen en cortes o distorsiones, siendo perceptibles para el usuario.
Dependencia de la estabilidad de la red	Aunque requiere menos ancho de banda, el audio es vulnerable a interrupciones si la red presenta congestión o variaciones.
Adaptación dinámica	Las aplicaciones ajustan la calidad del audio según el ancho de banda disponible para evitar interrupciones en la transmisión.
Optimización mediante compresión	Formatos como AAC o Opus permiten mantener una calidad aceptable con tasas de bits bajas, optimizando el uso del ancho de banda.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Tabla 7 resalta que, aunque el audio consume menos recursos que el video, su calidad depende críticamente de una transmisión estable y de baja latencia. Las aplicaciones multimedia implementan diversas estrategias, como la compresión y la adaptación dinámica, para mejorar la experiencia del usuario frente a condiciones adversas de red.

6.2 Tipos de aplicaciones multimedia

Internet ofrece una gran diversidad de aplicaciones multimedia, tanto útiles como de entretenimiento. Estas aplicaciones pueden agruparse en tres categorías principales:

1. Flujos de audio y video almacenado
2. Flujos de comunicación en tiempo real (voz y video)
3. Flujos de audio y video en vivo

Cada una de estas categorías presenta sus propios requisitos de servicio y desafíos de diseño (Kurose y Rose, 2017).

Flujos de audio/vídeo almacenado

Según Kurose & Rose (2017), los flujos de audio y video almacenado consisten en la transmisión de contenido pregrabado, como películas, programas de televisión, eventos deportivos o videos generados por los usuarios (por ejemplo, en plataformas como YouTube). Este contenido se almacena en servidores y los usuarios lo solicitan para reproducirlo bajo demanda. Las aplicaciones de transmisión de video almacenado, utilizadas por servicios como YouTube, Netflix y Amazon Prime, presentan tres características clave que se resumen en la Tabla 8.

Tabla 8
Propiedades de los flujos de video almacenado

Propiedad	Descripción
Flujos continuos (streaming)	• El cliente inicia la reproducción pocos segundos después de comenzar la descarga. • Se evita descargar el archivo completo, lo que reduce el tiempo de espera.
Interactividad	• El contenido pregrabado permite acciones como pausar, avanzar o retroceder.

Propiedad	Descripción
	• El sistema debe responder en menos de unos pocos segundos para mantener la fluidez.
Reproducción sin interrupciones	• Los datos deben llegar sincronizados con el tiempo de reproducción. • Retrasos pueden causar congelaciones o saltos en la imagen, afectando la experiencia.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Las propiedades descritas en la tabla 8 permiten que las aplicaciones de video almacenado ofrezcan una experiencia óptima al usuario. El streaming evita largos tiempos de espera, la interactividad mejora la usabilidad, y la reproducción sincronizada es crucial para mantener la continuidad en la visualización.

Flujos de comunicación en tiempo real (voz y video)

Las conversaciones de voz y video a través de Internet, conocidas como Voz sobre IP (VoIP) o videoconferencias, permiten a los usuarios comunicarse en tiempo real de manera similar a los servicios tradicionales de telefonía. Empresas como Skype, Zoom, Microsoft Teams y Google Meet han popularizado estas aplicaciones, alcanzando cientos de millones de usuarios activos diarios (Kurose & Rose, 2017).

La Tabla 9 resume las propiedades más importantes de las aplicaciones de conversación de voz y video sobre IP, destacando su impacto en el rendimiento y los requerimientos de red.

Tabla 9
Características de los flujos de comunicación en tiempo real

Propiedad	Descripción
Sensibilidad a los retardos	• La latencia debe ser mínima para mantener una comunicación fluida. • Los retardos aceptables son inferiores a 150 ms. Retardos superiores a 400 ms pueden hacer que la conversación sea ininteligible.
Tolerancia a la pérdida de datos	• Pérdidas ocasionales no afectan gravemente la comunicación. • Los cortes breves en el audio o video pueden compensarse mediante técnicas de corrección de errores y ocultación de pérdidas.
Reproducción adaptativa	• Las aplicaciones ajustan la calidad del audio y video según el ancho de banda disponible. • Esto garantiza continuidad incluso en situaciones de congestión o fluctuaciones en la red.
Comparación con aplicaciones de datos	• Las aplicaciones de voz y video priorizan la latencia y la fluidez sobre la integridad de los datos. • En aplicaciones como el correo electrónico, la integridad es más importante que los tiempos de respuesta.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Las aplicaciones de datos, como el correo electrónico o la navegación web, se caracterizan por priorizar la integridad de la información transferida. En estos casos, aunque los retardos puedan ser molestos, no afectan gravemente el funcionamiento de la aplicación. Por el contrario, en las aplicaciones de voz y video en tiempo real, la prioridad es mantener una comunicación fluida y con

baja latencia. Para ello, estas aplicaciones pueden tolerar la pérdida ocasional de pequeños fragmentos de datos sin que se vea comprometida la continuidad de la conversación.

Flujos de audio y video en vivo

Las aplicaciones de transmisión en vivo, similares a las emisoras de radio y televisión tradicionales, permiten a los usuarios acceder a contenido en tiempo real a través de Internet. Con este tipo de aplicaciones, es posible sintonizar eventos globales como noticias, deportes o conciertos. Miles de emisoras alrededor del mundo ofrecen actualmente servicios de streaming en vivo, apoyándose en tecnologías de redes avanzadas para garantizar una experiencia de usuario fluida. La Tabla 10 presenta las características principales de la transmisión en vivo por Internet.

Tabla 10
Propiedades de la transmisión en vivo

Propiedad	Descripción
Acceso simultáneo por múltiples usuarios	• Muchos usuarios pueden sintonizar una misma transmisión en tiempo real. • Se utilizan redes de distribución de contenido (CDN) para mejorar la eficiencia y reducir la latencia.
Requerimientos de tasa de transferencia	• La red debe mantener una tasa de transferencia media suficiente para evitar interrupciones. • Congelaciones o interrupciones ocurren si el flujo no es constante.
Tolerancia a la latencia inicial	• Un retraso inicial de hasta 10 segundos es aceptable para permitir la precarga de datos en el buffer. • Esto mejora la continuidad de la reproducción durante la transmisión.

Propiedad	Descripción
Similitudes con flujos almacenados	• Técnicas como el almacenamiento en buffer, la adaptación de ancho de banda y el uso de CDNs también se aplican en las transmisiones en vivo.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

A continuación, le invito a completar el siguiente juego de arrastrar y soltar, diseñada para reforzar sus conocimientos sobre los tres tipos de flujos multimedia por Internet; este juego le permitirá comprender las diferencias clave entre cada tipo de aplicación multimedia, así como sus requerimientos y desafíos técnicos.

Comparativa entre tipos de flujos multimedia por internet

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Realice una investigación sobre los principales formatos de compresión de audio y video (por ejemplo, MP3, AAC, MP4, H. 264). Compare las tasas de bits, calidad y requisitos de ancho de banda para cada formato. Reflexione sobre cómo estas propiedades afectan la reproducción en aplicaciones multimedia a través de diferentes tipos de conexión (fibra, 4G, 5G, etc.).
2. Identifique una aplicación multimedia que utilice cada tipo de flujo (almacenado, en vivo y comunicación en tiempo real). Analice sus características técnicas, tales como tasas de bits, gestión de retardos, tolerancia a la pérdida de datos e interactividad. Prepare un breve informe o presentación en el

que destaque las diferencias de rendimiento y la experiencia de usuario en distintos escenarios de red.

Nota: por favor complete las actividades en un cuaderno o documento Word

¡Felicidades por completar una semana más de estudio! Cada avance refuerza su conocimiento y habilidades en la materia. Es importante reconocer que el aprendizaje es un proceso continuo, y usted está demostrando compromiso y constancia. Aunque puedan surgir retos, estos son oportunidades para crecer y fortalecer su comprensión. Prepárese para una nueva semana llena de descubrimientos. ¡Continúe con ese entusiasmo y siga avanzando hacia sus metas con confianza!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 14

Unidad 6. Redes multimedia

Esta semana nos adentraremos en aspectos técnicos esenciales para la transmisión eficiente de contenido multimedia en redes. Analizaremos cómo los flujos de video almacenado utilizan protocolos como UDP y HTTP, revisaremos los principales protocolos multimedia modernos, y exploraremos las estrategias para garantizar la calidad de servicio (QoS) en entornos de red. Estos conceptos son fundamentales para comprender cómo se logra una transmisión de datos estable, fluida y adaptada a las necesidades de los usuarios. ¡Es hora de seguir aprendiendo!

6.3 Flujos de video almacenado (UDP y HTTP)

Para Kurose & Rose (2017), los flujos de video almacenado son una forma común de transmisión multimedia, donde los usuarios pueden acceder a contenido pregrabado en servidores, como películas, series, eventos

deportivos o videos generados por usuarios. Este tipo de aplicaciones permite al usuario disfrutar de la reproducción de principio a fin, así como interactuar con el contenido mediante acciones como pausa, avance rápido o retroceso.

Los sistemas de transmisión de video se dividen en tres categorías principales:

1. **Flujos UDP:** Utilizan el Protocolo de Datagrama de Usuario (UDP), que ofrece baja latencia, pero no garantiza la entrega completa de datos, lo que puede resultar en pérdida de calidad si no se gestiona adecuadamente.
2. **Flujos HTTP:** Transmiten los datos a través del protocolo HTTP. Este enfoque se basa en la confiabilidad del protocolo TCP, asegurando que todos los paquetes lleguen correctamente, aunque a veces con mayor retraso.
3. **Flujos HTTP adaptativos:** Ajustan la calidad del video en tiempo real según el ancho de banda disponible, lo que permite ofrecer una experiencia más fluida a los usuarios, incluso en condiciones variables de red.

Importancia del buffer en la reproducción

Un aspecto clave en estos sistemas es el uso de un buffer en el lado del cliente. Este almacenamiento temporal tiene dos funciones principales:

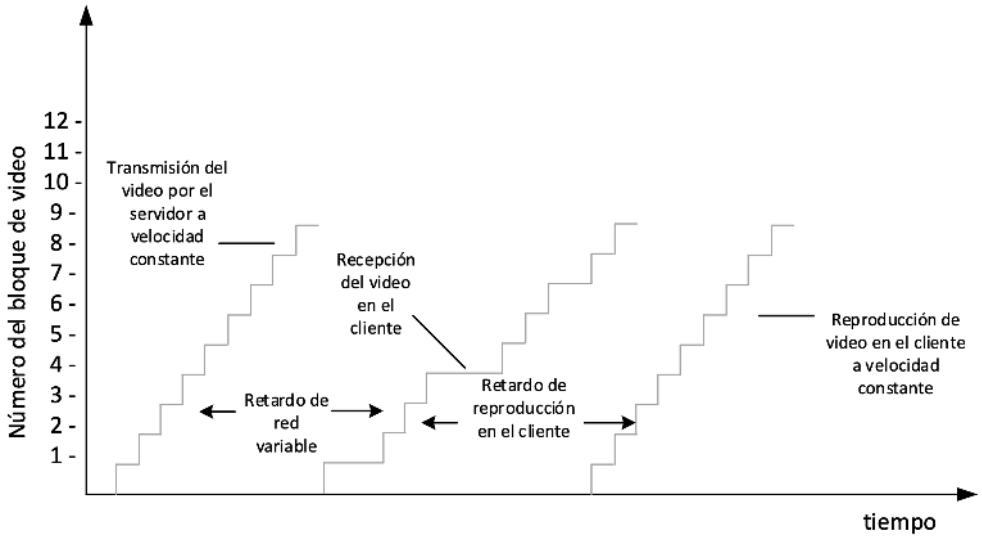
- **Absorber variaciones en el retardo:** Los datos de video no siempre llegan en tiempos uniformes debido a la naturaleza de la red. El buffer permite que el cliente espere unos segundos antes de comenzar la reproducción, evitando interrupciones.
- **Mantener la reproducción continua:** Si el ancho de banda disminuye temporalmente, el buffer proporciona una reserva de datos que garantiza que la visualización no se interrumpa, siempre que no se vacíe completamente.

Comportamiento del buffer

El proceso de almacenamiento en buffer se ilustra en la Figura 36, ahí se puede identificar bloques de video que se transmiten y reciben con diferentes tiempos de llegada. Si el cliente inicia la reproducción tan pronto como llegue el primer bloque, podría enfrentarse a retrasos o saltos de fotogramas. Retrasar la reproducción hasta que se haya acumulado una cantidad suficiente de bloques permite una experiencia continua y sincronizada.

Figura 36

Retardo y buffer en la reproducción de un flujo de video en el cliente



Nota. Adaptado de *Redes de computadoras. Un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Este enfoque se aplica en muchas plataformas de streaming actuales, como Netflix o YouTube, que optimizan la entrega de contenido utilizando redes de distribución (CDN) y técnicas adaptativas que ajustan la calidad del video para mejorar la experiencia del usuario.

Flujos de video almacenado usando UDP

Los flujos de video mediante el Protocolo de Datagrama de Usuario (UDP) se caracterizan por transmitir el video a una tasa constante, sincronizada con la velocidad de consumo del cliente. Sin embargo, este enfoque presenta ventajas y desventajas importantes (Kurose & Rose, 2017). A continuación, la Tabla 11 presenta las características principales de los flujos con UDP.

Tabla 11
Flujos de video mediante UDP

Aspecto	Descripción
Velocidad de transmisión	Transmisión constante a la velocidad de consumo del video por el cliente.
Protocolo de transporte	Utiliza UDP y puede encapsular datos en RTP u otros esquemas similares.
Tamaño del buffer	Buffer pequeño, generalmente suficiente para almacenar menos de un segundo de video.
Conexión de control	Conexión de control paralela con comandos de estado como pausa, continuación, reposicionamiento.
Desventajas principales	Dependencia de ancho de banda constante, complejidad de un servidor de control de medios y problemas con cortafuegos que bloquean UDP.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

A pesar de estas limitaciones, los flujos UDP son utilizados en sistemas que priorizan la baja latencia, aunque su implementación requiere soluciones robustas para gestionar variaciones en el ancho de banda y la interactividad.

Flujos HTTP

Para Kurose & Rose (2017), los flujos de video a través de HTTP utilizan servidores estándar donde el contenido se almacena como archivos con URL específicos. Este método ha ganado popularidad gracias a su capacidad para

atravesar cortafuegos y redes NAT, así como por su simplicidad en comparación con otras alternativas. A continuación, la Tabla 12 detalla las principales características de este enfoque.

Tabla 12
Características de los flujos HTTP para transmisión de video

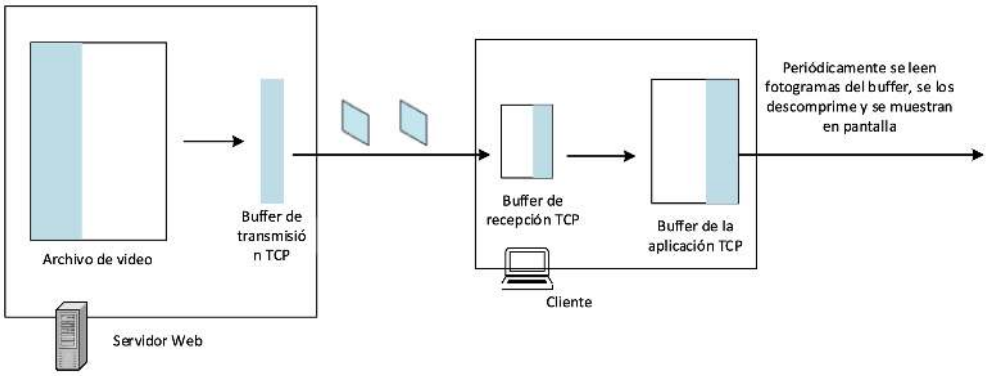
Aspecto	Descripción
Almacenamiento	El video se guarda como un archivo estándar en un servidor HTTP, accesible a través de una URL específica.
Conexión	Se establece una conexión TCP entre el cliente y el servidor, donde el cliente envía una solicitud GET para acceder al archivo de video.
Velocidad de transmisión	Controlada por TCP, influida por mecanismos de flujo, congestión y retransmisión, lo que puede ocasionar variaciones en la velocidad.
Buffer en el cliente	Los datos se acumulan en el buffer del cliente antes de comenzar la reproducción, evitando interrupciones debidas a fluctuaciones de la red.
Compatibilidad	Es compatible con cortafuegos y sistemas NAT, que suelen bloquear el tráfico UDP pero permiten el tráfico HTTP, lo que mejora la accesibilidad.
Costo de implementación	Al no requerir un servidor de control de medios, se reducen los costos asociados a la implementación a gran escala.
Ejemplos	Utilizado por plataformas populares como YouTube y Netflix para la transmisión de video.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Esta tabla proporciona una visión clara y estructurada de las ventajas y consideraciones del uso de flujos HTTP para la transmisión de video en aplicaciones multimedia. La Figura 37 muestra el proceso completo de transmisión de video mediante el protocolo HTTP sobre TCP. El esquema detalla las etapas que atraviesan los datos desde el servidor hasta su reproducción en el cliente.

Figura 37

Flujo de video a través de HTTP sobre TCP



Nota. Adaptado de *Redes de computadoras. Un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

La Figura 37 muestra el proceso completo de transmisión de video mediante el protocolo HTTP sobre TCP. El esquema detalla las etapas que atraviesan los datos desde el servidor hasta su reproducción en el cliente:

1. **Archivo de video en el servidor web:** El contenido del video se almacena como un archivo estándar en un servidor HTTP, disponible mediante una URL.
2. **Transmisión de video a través de TCP:** El servidor transmite el archivo al cliente, dividiéndolo en fragmentos de datos que son enviados al buffer de transmisión TCP. Estos fragmentos son enviados tan rápido como lo permita el control de flujo y congestión de TCP.
3. **Buffer de recepción en el cliente:** Los fragmentos de video llegan al cliente donde se almacenan temporalmente en el buffer de recepción TCP. Este buffer ayuda a manejar las variaciones en el tiempo de llegada de los datos debido a posibles fluctuaciones de red.
4. **Buffer de la aplicación cliente:** Desde el buffer de recepción, los fragmentos son trasladados al buffer de la aplicación cliente. Este buffer almacena varios segundos de video antes de iniciar la reproducción, asegurando una experiencia continua y sin interrupciones.

5. **Reproducción del video:** La aplicación cliente extrae periódicamente fotogramas del buffer, los descomprime y los muestra en la pantalla del usuario.

Este flujo es esencial para mantener la sincronización y calidad del video en aplicaciones multimedia. Gracias al uso de buffers, se mitigan los efectos de las variaciones en la red, mejorando la experiencia del usuario

6.4 Protocolos multimedia modernos para la transmisión de video

Para Wilbert (2023), los protocolos multimedia modernos juegan un papel esencial en el flujo de contenido audiovisual en Internet. Estos protocolos regulan la transmisión de video dividiendo los archivos en fragmentos para su entrega eficiente a los usuarios. La Tabla 13 presenta los principales protocolos utilizados en la transmisión de video en línea, detallando sus características, usos y ventajas.

Tabla 13

Principales protocolos de transmisión de video

Protocolo	Descripción	Ventajas	Desventajas
HTTP Live Streaming (HLS)	Desarrollado por Apple, es el protocolo más usado hoy. Divide el video en fragmentos y utiliza HTTP para la entrega.	Alta compatibilidad con navegadores y dispositivos. Transmisión adaptativa.	Alta latencia. No es ideal para la ingestión directa.
RTMP (Real-Time Messaging Protocol)	Inicialmente diseñado para Flash, se usa actualmente para la ingestión de video a plataformas de streaming.	Baja latencia. Compatible con múltiples formatos de video.	No compatible con HTML5. Vulnerable a problemas de ancho de banda.
WebRTC	Protocolo de comunicación en tiempo real diseñado para videoconferencias y chat de voz/video.	Latencia casi en tiempo real. Código abierto y adaptable.	Tecnología reciente con algunos problemas de compatibilidad.
SRT (Secure Reliable Transport)	Protocolo seguro y confiable para streaming de baja latencia. Fue desarrollado por Haivision.	Alta seguridad. Compatible con múltiples dispositivos.	Limitada adopción en la industria.
RTSP (Real-Time Streaming Protocol)	Utilizado para el control de sesiones multimedia, especialmente en dispositivos de IoT y cámaras de seguridad.	Permite personalización del flujo. Soporta streaming segmentado.	Baja popularidad. Incompatibilidad con HTTP y navegadores web.
MPEG-DASH	Protocolo basado en HTTP que ofrece transmisión adaptativa con alta calidad de video.	Agnóstico en cuanto a códecs. Transmisión adaptativa.	No compatible con dispositivos Apple. Adopción limitada.

Nota. Adaptado de *Protocolos de transmisión de vídeo: 6 formatos preferidos para la radiodifusión profesional*, por Wilbert, M., 2023.

Estos protocolos trabajan en conjunto para proporcionar una experiencia de transmisión fluida y segura, adaptándose a diferentes necesidades de calidad, latencia y compatibilidad. HLS y RTMP son las combinaciones más utilizadas

en configuraciones de transmisión actuales, aunque protocolos como SRT y WebRTC continúan ganando terreno en la industria del streaming; para reforzar su aprendizaje lo invito a desarrollar el siguiente juego de relacionar.

Explicación de los protocolos de video más relevantes

Actividades de aprendizaje recomendadas

Para fortalecer sus conocimientos, a continuación, lo invito a desarrollar las siguientes actividades recomendadas:

1. Configure un entorno de simulación utilizando herramientas como Wireshark o un servidor local. Genere flujos de video almacenado tanto por UDP como por HTTP; analice las diferencias en términos de transmisión, tiempos de respuesta y posibles problemas de latencia o pérdida de paquetes. Realice capturas de tráfico para documentar sus observaciones.
2. Realice una investigación sobre al menos tres protocolos multimedia modernos (HLS, RTMP y WebRTC); cree una tabla comparativa detallando sus características, ventajas, desventajas y casos de uso específicos. Incluya ejemplos de plataformas o aplicaciones que utilicen estos protocolos.

Nota: Por favor complete las actividades en un cuaderno o documento Word

¡Felicidades por haber concluido con éxito la semana 14 de estudio! Durante estos días, ha profundizado en conceptos esenciales sobre los flujos de video almacenado y los protocolos multimedia modernos, conocimientos fundamentales en el ámbito de las redes multimedia. Este avance demuestra su compromiso y dedicación hacia el aprendizaje constante.

Ahora nos encontramos en la recta final. ¡La semana 15 está a la vuelta de la esquina! En esta última etapa, exploraremos temas cruciales como las redes de distribución de contenido (CDN) y estudiaremos un caso práctico con una de las plataformas más importantes del mundo: Netflix.

¡Ánimo! Está a un paso de alcanzar la meta. ¡Continúe con el mismo esfuerzo y dedicación que ha demostrado hasta ahora!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 15

Unidad 6. Redes multimedia

¡Bienvenido a la semana 15 de estudio! En esta etapa culminante de nuestro recorrido académico, nos enfocaremos en las redes de distribución de contenido (CDN), una tecnología clave para mejorar la entrega eficiente de contenido multimedia en todo el mundo. Además, analizaremos un caso práctico sobre Netflix, una de las plataformas más emblemáticas en el uso de CDN.

6.5 Redes de distribución de contenido – CDN

Una Red de Distribución de Contenido (CDN) es una infraestructura de servidores distribuidos geográficamente que se utiliza para optimizar la entrega de contenido digital, como videos, documentos, imágenes o aplicaciones. Su propósito es mejorar la velocidad, disponibilidad y seguridad de la distribución, reduciendo los tiempos de carga y minimizando los problemas relacionados con la latencia (Kurose & Rose, 2017).

Actualmente, empresas como YouTube, Netflix y Amazon utilizan CDN para entregar contenido de video de múltiples megabytes a millones de usuarios simultáneamente. Sin estas redes, las empresas enfrentarían desafíos significativos en términos de rendimiento, capacidad de red y costos operativos.

Problemas de distribución desde un único centro de datos

Distribuir contenido únicamente desde un servidor central puede resultar problemático debido a los siguientes factores:

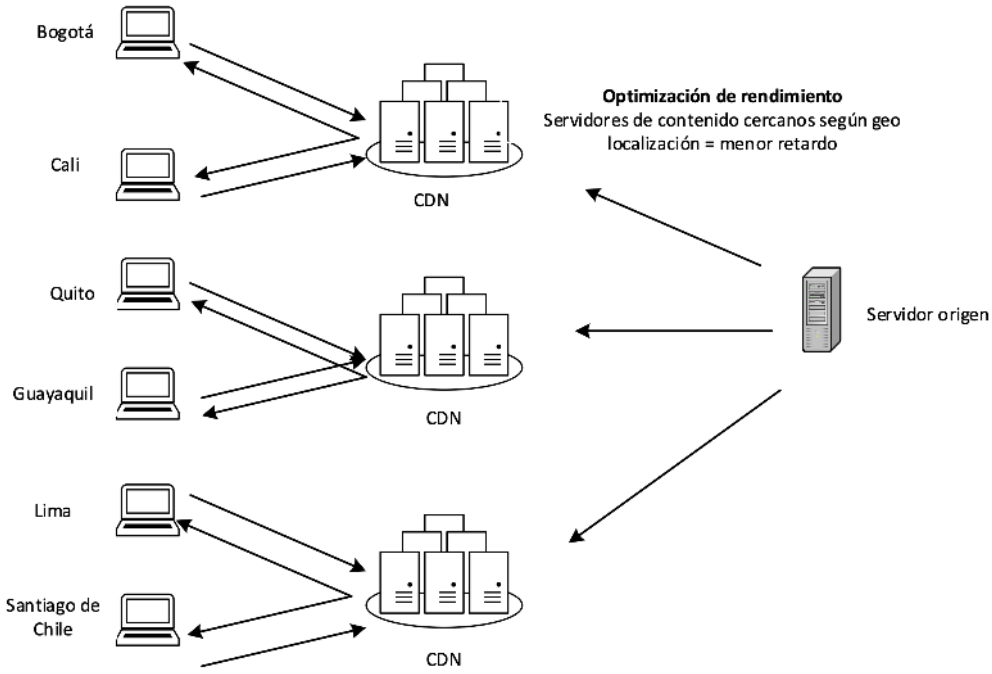
- **Latencia alta y cuellos de botella:** Si los clientes están lejos del servidor, los paquetes atraviesan múltiples enlaces y proveedores de servicios (ISP), lo que reduce la tasa de transferencia y genera interrupciones en la reproducción.
- **Uso excesivo del ancho de banda:** Contenidos populares deben ser enviados repetidamente por los mismos enlaces, incrementando los costos y saturando la capacidad de la red.
- **Fallo único**: Si el servidor central o sus conexiones fallan, todo el sistema de distribución se interrumpe.

Funcionamiento de una CDN

Las CDN resuelven estos problemas mediante servidores distribuidos en diversas regiones. La Figura 38 muestra una CDN y su estructura general.

Figura 38

Redes de entrega de contenido – CDN



Nota. Adaptado de *Que es un CDN (red de distribución de contenido)* [Ilustración], por Sanz, 2016, Comunidad Hosting, CC BY 4.0.

La Figura 38 muestra cómo los servidores almacenan copias de los contenidos y responden a las solicitudes de los usuarios desde el servidor más cercano o eficiente. Este enfoque optimiza la experiencia del usuario al disminuir la latencia y mejorar la velocidad de transferencia de datos.

Tipos de CDN

- **CDN privada:** Propiedad del proveedor de contenido (ejemplo: Google para YouTube).
- **CDN comercial:** Ofrecen servicios a múltiples proveedores de contenido (ejemplo: Akamai, Limelight).

Estrategias de distribución

A continuación, la Tabla 14 explica cómo la CDN utiliza dos enfoques principales para colocar sus servidores.

Tabla 14
Enfoques principales de la CND para colocación de servidores

Enfoque	Descripción	Ventajas	Desafíos
Introducción profunda	Los servidores se colocan dentro de las redes de acceso de múltiples ISP. Ejemplo: Akamai.	Menor retardo y mayor tasa de transferencia.	Alta complejidad y costos de mantenimiento.
Atracción de ISP	Grandes clústeres se ubican en puntos de intercambio de Internet (IXP). Ejemplo: Limelight.	Menores costos de mantenimiento.	Mayor latencia y tasas de transferencia variables.

Nota. Adaptado de *Redes de computadoras. un enfoque descendente*, por Kurose, J. y Ross, K., 2017, Pearson.

Gestión del contenido en una CDN

El contenido se replica dinámicamente entre los servidores de la CDN según la demanda. Si un servidor no tiene un vídeo solicitado, lo obtiene de otro nodo y guarda una copia local para futuras solicitudes. Los vídeos menos demandados son eliminados para liberar espacio (Kurose & Rose, 2017).

Beneficios de las CDN

- **Reducción de latencia**: Los servidores cercanos al usuario permiten tiempos de respuesta más rápidos.
- **Optimización de recursos:** Minimiza el tráfico repetido en los enlaces principales de la red.

- **Alta disponibilidad:** La distribución entre múltiples servidores reduce el riesgo de interrupciones.

6.6 Caso de estudio CDN

Según Fernández (2023), Netflix genera aproximadamente el 13% del tráfico global de Internet, lo que evidencia la alta demanda de contenido en streaming que la plataforma produce a nivel mundial, rivalizando con el tráfico combinado de YouTube y TikTok. Esto ha consolidado a Netflix como el principal proveedor de servicios de películas y series de TV en línea en los Estados Unidos, donde representa una parte importante del tráfico de descarga en los ISP residenciales de Norteamérica. Su infraestructura de distribución de contenido, actualizada a 2025, se compone de dos elementos clave: la nube de Amazon y su red de distribución de contenido (CDN) privada, conocida como "Netflix Open Connect".

Funciones en la nube de Amazon

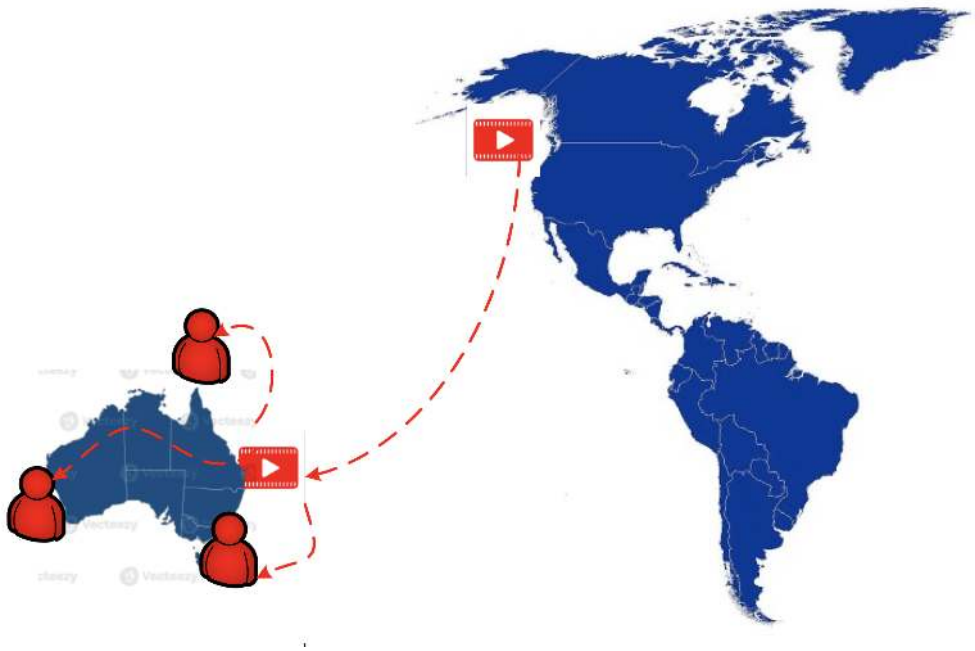
Netflix utiliza la infraestructura de Amazon para manejar diversas funciones críticas, entre ellas:

- **Gestión del sitio web:** El sitio web de Netflix, donde los usuarios registran sus cuentas, inician sesión, exploran el catálogo y reciben recomendaciones, se ejecuta íntegramente en servidores de Amazon.
- **Ingesta y procesamiento de contenido:** Las versiones maestras de las películas se cargan en la nube de Amazon, donde se procesan para crear múltiples formatos adaptados a una variedad de dispositivos, desde computadoras hasta consolas de juegos. Se generan versiones con distintas tasas de bits para permitir la transmisión adaptativa mediante HTTP y el protocolo DASH (Dynamic Adaptive Streaming over HTTP).
- **Carga en la CDN:** Una vez procesados los contenidos, se transfieren desde la nube de Amazon a la infraestructura CDN de Netflix.

Al abordar el preposicionamiento de contenido de esta manera, Netflix ahorra recursos esenciales de Internet. En Australia, por ejemplo, todo el acceso a contenido en Internet que no se origina en ese país llega a través de cables submarinos. En vez de usar este costoso sistema para el tráfico de Netflix, Netflix copia cada archivo una vez desde su repositorio de transcodificación en EE. UU. a las ubicaciones de almacenamiento en Australia. El procedimiento se hace durante horas de bajo consumo, cuando no se compite contra otros tráficos de Internet. Entonces, una vez que cada archivo ya está en el continente, se replica en docenas de servidores Open Connect dentro de cada red de ISP; la Figura 39 representa este proceso.

Figura 39

Preposicionamiento de contenido de Netflix



Nota. Adaptado de *Netflix trabaja con los Proveedores de Servicios de Internet (ISP) de todo el mundo para que la experiencia de los espectadores sea extraordinaria - About Netflix* [Ilustración], por About Netflix, s.f., About Netflix, CC BY 4.0.

Evolución de la CDN de Netflix

Inicialmente, en 2007, Netflix utilizaba servicios de terceros para distribuir sus vídeos. Sin embargo, con el tiempo, desarrolló su propia CDN privada llamada "Netflix Open Connect". Esta infraestructura está compuesta por bastidores de servidores ubicados en puntos de intercambio de internet (IXP) y en redes de ISP residenciales. Actualmente, Netflix cuenta con más de 50 ubicaciones en IXP y cientos de instalaciones en ISP.

Cada bastidor de servidores está diseñado para maximizar el rendimiento, con puertos Ethernet de 10 Gbps y más de 100 terabytes de almacenamiento. En los IXP, los bastidores suelen contener toda la biblioteca de contenido, mientras que en los ISP locales se almacenan solo los vídeos más populares. A diferencia de otros sistemas de distribución que utilizan cachés bajo demanda (pull-caching), Netflix implementa un sistema de carga programada (push-caching), donde los vídeos se actualizan durante las horas de menor tráfico.

Interacción cliente-servidor en la distribución de contenido

Cuando un usuario selecciona una película, el software de Netflix, ejecutándose en la nube de Amazon, determina qué servidor de la CDN tiene la mejor disponibilidad de la película. Si el usuario pertenece a un ISP con un bastidor de servidores que contiene la película solicitada, se elige ese servidor. De lo contrario, se selecciona un servidor en un IXP cercano.

Netflix envía al cliente la dirección IP del servidor seleccionado y un archivo de manifiesto con las URL de las distintas versiones del contenido. A partir de ahí, el cliente y el servidor CDN interactúan directamente utilizando una versión propietaria de DASH. El cliente descarga los segmentos de vídeo de aproximadamente cuatro segundos de duración y ajusta la calidad en función de la velocidad de transferencia recibida.

Optimización del diseño de la CDN

A diferencia de otros servicios que dependen de redirección DNS para vincular clientes a servidores, Netflix utiliza su propia lógica en la nube de Amazon para asignar directamente el servidor CDN adecuado. Este enfoque simplificado le permite controlar la distribución de contenido con mayor eficiencia y planificar mejor el uso de recursos durante períodos de menor tráfico.

Netflix ha logrado así integrar principios avanzados de transmisión adaptativa y distribución de contenidos, mejorando la experiencia del usuario con menores tiempos de carga y mayor calidad de reproducción.

Después de revisar el caso de estudio sobre Netflix y su implementación de redes de distribución de contenido (CDN), te sugiero realizar una investigación adicional sobre las actualizaciones más recientes de este tema en la web. Las tecnologías y estrategias que emplea Netflix, como su infraestructura Open Connect, evolucionan constantemente para mejorar el rendimiento, la escalabilidad y la experiencia del usuario.

¡Felicidades por tu esfuerzo y dedicación! Has culminado exitosamente el estudio de la Unidad 6. Ahora es momento de prepararte para el examen bimestral. Antes de ello, te invito a participar en las siguientes actividades diseñadas para reforzar y aplicar los conocimientos adquiridos durante esta unidad. ¡Aprovecha esta oportunidad para consolidar lo que has aprendido!

Actividades de aprendizaje recomendadas

1. Explore cómo empresas como YouTube (Google), Netflix y Akamai implementan y gestionan sus redes de distribución de contenido. Identifique las diferencias en sus estrategias (por ejemplo, introducción profunda frente a atracción de ISP) y analice cómo estas afectan la calidad del servicio. Puede buscar artículos técnicos, estudios de caso o informes anuales de estas empresas.

2. Investigue los servicios CDN que ofrecen empresas como Akamai, Cloudflare, Amazon CloudFront y Microsoft Azure CDN. Evalúe sus características clave, como escalabilidad, tiempos de latencia, seguridad y costos. Reflexiona sobre cuál podría ser más adecuado según diferentes tipos de negocios o aplicaciones.
3. Antes de avanzar, le recomiendo completar la autoevaluación 6, correspondiente a la última unidad de nuestra asignatura. Lea atentamente las preguntas propuestas en relación con los conceptos de redes multimedia, tipos de flujos y el caso de estudio revisado.

   Esta actividad le permitirá medir su comprensión de los conceptos tratados en esta unidad. Dedique el tiempo y esfuerzo necesarios, ya que será clave para consolidar lo aprendido.

Autoevaluación 6

Seleccione la opción de respuesta correcta para cada pregunta. Estas preguntas le ayudarán a reforzar y evaluar su comprensión de los contenidos estudiados.

1. ¿Cuál es el propósito principal de una Red de Distribución de Contenido (CDN)?

   A. Reducir los costos de almacenamiento en servidores.
   B. Aumentar el número de servidores centrales.
   C. Mejorar la velocidad, disponibilidad y seguridad en la entrega de contenido digital.

2. ¿Cuál de las siguientes es una característica distintiva de los flujos de video almacenado?

   A. Permite acciones como pausa, avance y retroceso.
   B. Solo funciona con conexiones de alta velocidad.

C. No permite la reproducción sin buffer.

3. ¿Qué problema se presenta al distribuir contenido desde un único centro de datos?

   A. Reducción del uso de ancho de banda.
   B. Alta latencia y puntos únicos de fallo.
   C. Mejora en el rendimiento de la red.

4. ¿Qué estrategia de colocación de servidores CDN busca reducir los costos de mantenimiento?

   A. Atracción de ISP.
   B. Introducción profunda.
   C. Sincronización global de servidores.

5. ¿Cuál de los siguientes protocolos es el más utilizado actualmente para la transmisión de video en línea?

   A. RTMP
   B. WebRTC
   C. HTTP Live Streaming (HLS)

6. ¿Qué función tiene el buffer del cliente en aplicaciones de streaming?

   A. Almacenar temporalmente los datos para compensar fluctuaciones en la red.
   B. Limitar el tiempo máximo de reproducción.
   C. Descomprimir los fotogramas antes de enviarlos a la pantalla.

7. ¿Cuál es la ventaja de utilizar flujos HTTP en la transmisión de video?

   A. Permite atravesar cortafuegos y redes NAT.
   B. No requiere conexión TCP para la transmisión.

C. Disminuye la latencia más que otros protocolos.

8. ¿Qué protocolo es utilizado por Netflix para gestionar la calidad adaptativa del video?

   A. Dynamic Adaptive Streaming over HTTP (DASH).
   B. Real-Time Messaging Protocol (RTMP).
   C. Secure Reliable Transport (SRT).

9. ¿Qué ventaja ofrece la transmisión adaptativa en plataformas como YouTube?

   A. Elimina la necesidad de almacenamiento en buffer.
   B. Ajusta automáticamente la calidad del video en función del ancho de banda.
   C. Reduce los costos de implementación de servidores.

10. ¿Cuál es una estrategia común en las CDN para gestionar contenido popular y menos solicitado?

    A. Eliminar los videos populares cuando se llena el almacenamiento.
    B. Replicar todos los videos en todos los servidores.
    C. Almacenar los videos menos demandados solo en servidores principales.

Ir al solucionario

¡Le deseo mucho éxito en su proceso de aprendizaje!

Si al contestar la autoevaluación ha obtenido resultados positivos, ¡FELICITACIONES! Siga adelante con su gran desempeño. En caso contrario, le recomiendo revisar nuevamente el contenido de los ítems que contestó incorrectamente, para reforzar su aprendizaje.

Recuerde que siempre puede consultar con su profesor tutor si tiene alguna pregunta o inquietud. Estoy seguro de que, con un poco de repaso y dedicación, podrá mejorar y alcanzar sus objetivos. ¡No se desanime y siga esforzándose!

Contenidos, recursos y actividades de aprendizaje recomendadas

Semana 16

Actividades finales del bimestre

Estamos cerrando el estudio de nuestra asignatura, Arquitectura de Redes, confiamos en que esta experiencia de aprendizaje le haya motivado a profundizar en el fascinante mundo de las redes multimedia, protocolos, seguridad, y aplicaciones orientadas a datos. Este campo, clave en la infraestructura tecnológica actual, ofrece amplias oportunidades de especialización e investigación, así como múltiples salidas profesionales en el ámbito de las tecnologías de la información.

Durante esta última semana, es esencial que enfoque su dedicación en la preparación para el examen bimestral. Para lograr una preparación efectiva, le recomiendo lo siguiente:

1. **Organice un esquema comparativo de conceptos**: Resuma los protocolos, servicios y arquitecturas estudiados (redes en la nube, distribución de contenido, seguridad en redes), detallando sus aplicaciones prácticas y ventajas.
2. **Repase los contenidos abordados durante el segundo bimestre:** Enfoque su revisión en las unidades 4, 5 y 6, que incluyen temas como:
    - Redes en la nube y configuración de recursos (VPC, opciones de conectividad).
    - Seguridad en redes de computadoras (cortafuegos, IPsec, VPNs).
    - Redes multimedia, incluyendo flujos de video, protocolos de transmisión y CDN.

3. **Revise las actividades prácticas:** Realice un repaso de las simulaciones, análisis de protocolos, prácticas de laboratorio, recursos interactivos y los cuestionarios realizados durante las semanas previas.
4. **Identifique áreas de mejora:** Si nota que hay temas en los cuales necesita reforzar su conocimiento, realice actividades complementarias, investigue más a fondo y anote sus dudas.
5. **Aproveche las sesiones de tutoría:** Comparta sus inquietudes con el tutor para recibir orientación personalizada y resolver cualquier tema pendiente.

Recuerde que el examen estará orientado a evaluar su capacidad para comprender, aplicar y analizar los conceptos clave de la asignatura. Su dedicación y constancia le llevarán al éxito.

¡Confíe en su esfuerzo, continúe avanzando y finalice con excelencia!*¡Le deseamos el mejor de los éxitos y le felicitamos por culminar exitosamente el estudio de todas las unidades de la materia de Computación en la Nube!*

Es un logro destacable haber completado este proceso de aprendizaje, y espero sinceramente que haya sido una experiencia enriquecedora en su formación académica. Ha adquirido conocimientos sobre la capa de aplicación, su diseño, funcionamiento, protocolos y aplicaciones actuales. Aprendió a implementar, gestionar y proteger redes, con un enfoque en datos, multimedia y contenido en la nube.

Sin embargo, este es solo el comienzo. Las redes y las tecnologías asociadas evolucionan constantemente, y cada nueva experiencia será una oportunidad para crecer y especializarse. Le animo a continuar investigando, aplicando lo aprendido y explorando áreas complementarias para fortalecer su formación profesional.

¡Le deseo mucho éxito en sus futuros proyectos y metas! Continúe avanzando con la misma dedicación y confianza que ha demostrado hasta ahora**.**

4. Autoevaluaciones

Autoevaluación 1

Pregunta	Respuesta	Retroalimentación
1	B	La capa de aplicación permite a las aplicaciones comunicarse entre diferentes sistemas, facilitando servicios esenciales como el correo electrónico y la navegación web.
2	B	El modelo OSI organiza las funciones de red en siete capas, donde la capa de aplicación es la más alta.
3	B	HTTPS es un protocolo de la capa de aplicación que garantiza la comunicación segura en la web.
4	B	El protocolo DNS traduce nombres de dominio en direcciones IP, facilitando el acceso a servicios en la red.
5	B	HTTPS cifra las comunicaciones, garantizando la seguridad y confidencialidad de los datos.
6	B	DHCP asigna dinámicamente direcciones IP, simplificando la configuración de los dispositivos conectados a la red.
7	C	HTTPS es un servicio esencial de la capa de aplicación que permite la transferencia segura de datos en la web.
8	B	SMTP se encarga de enviar mensajes de correo electrónico entre servidores.
9	B	El sistema DNS traduce los nombres de dominio en direcciones IP para que los usuarios puedan acceder fácilmente a sitios web y servicios.
10	A	En el modelo TCP/IP, la capa de aplicación agrupa las funciones de las capas de aplicación, presentación y sesión del modelo OSI.

Ir a la autoevaluación

Autoevaluación 2

Pregunta	Respuesta	Retroalimentación
1	B	La multiplexación permite que HTTP/2 maneje múltiples solicitudes y respuestas simultáneamente sobre una única conexión, mejorando el rendimiento.
2	C	HTTPS utiliza el puerto 443, proporcionando cifrado y seguridad a las comunicaciones web.
3	B	DNS traduce nombres legibles para humanos (como www.ejemplo.com) en direcciones IP que los sistemas utilizan para conectarse.
4	C	Los servidores TLD gestionan los dominios de nivel superior, proporcionando direcciones IP de servidores autoritativos.
5	B	El certificado incluye la clave pública del servidor, que es utilizada para cifrar la clave de sesión en la comunicación.
6	B	DHCP permite la asignación automática de direcciones IP a dispositivos en la red, simplificando la configuración.
7	B	En HTTP/1.0, cada solicitud requería una nueva conexión, lo que aumentaba la latencia y reducía la eficiencia.
8	B	HTTPS cifra las comunicaciones mediante SSL/TLS, evitando que terceros puedan interceptar o modificar los datos.
9	B	El servidor DNS local envía la consulta a un servidor DNS raíz para continuar el proceso de resolución de nombres.
10	B	HTTPS añade una capa de seguridad mediante cifrado SSL/TLS, protegiendo las comunicaciones frente a ataques e interceptaciones.

Ir a la autoevaluación

Autoevaluación 3

Pregunta	Respuesta	Retroalimentación
1	B	Los sockets permiten la comunicación bidireccional entre procesos de red, ofreciendo un punto de conexión en aplicaciones cliente-servidor.
2	A	UDP no garantiza la entrega de paquetes ni el orden, lo que requiere que las aplicaciones manejen la fiabilidad por su cuenta.
3	B	REST se basa en una comunicación sin estado, donde cada solicitud es independiente, lo que mejora la escalabilidad y simplicidad del diseño.
4	B	GraphQL permite a los clientes definir la estructura de la respuesta, evitando el envío de datos innecesarios.
5	A	Los filtros de captura permiten enfocar la recolección de datos en paquetes específicos, lo que mejora la eficiencia del análisis.
6	C	El puerto 80 se utiliza para conexiones HTTP no cifradas, mientras que el puerto 443 es usado para HTTPS.
7	C	El método GET recupera datos de un recurso sin realizar cambios en el servidor.
8	B	Los protocolos de texto sin cifrar permiten un análisis más detallado del contenido y estructura de los paquetes.
9	A	TCP proporciona una conexión confiable, garantizando la entrega y el orden correcto de los paquetes.
10	B	Wireshark permite examinar los detalles de los paquetes de red, incluyendo protocolos, tiempos de respuesta y direcciones involucradas.

Ir a la autoevaluación

Autoevaluación 4

Pregunta	Respuesta	Retroalimentación
1	A	Las redes en la nube se basan en la virtualización y son gestionadas mediante software, permitiendo flexibilidad en su configuración.
2	C	La VPN permite una conexión segura y eficiente entre los entornos on-premise y la nube, proporcionando comunicación encriptada.
3	A	Las VPC permiten crear redes privadas dentro de la infraestructura de un proveedor de servicios en la nube, ofreciendo aislamiento lógico.
4	A	Las subredes públicas y privadas son fundamentales para segmentar y controlar el tráfico dentro de una VPC.
5	A	La puerta de enlace de Internet es un recurso virtual que permite el tráfico entrante y saliente hacia y desde la red VPC.
6	C	Las subredes privadas mejoran la seguridad al mantener los recursos internos inaccesibles desde el exterior sin reglas de acceso específicas.
7	A	Los grupos de seguridad definen las reglas de tráfico entrante y saliente para los recursos en una VPC.
8	A	La puerta de enlace NAT permite a los recursos de subredes privadas enviar solicitudes hacia Internet sin recibir conexiones directas desde el exterior.
9	B	Para que una instancia sea accesible desde el exterior, debe estar en una subred pública y tener configuradas las reglas de seguridad adecuadas.
10	A	El laboratorio requiere la creación y configuración de una VPC, subredes, grupos de seguridad e instancias para implementar una red funcional en la nube.
		Ir a la autoevaluación

Autoevaluación 5

Pregunta	Respuesta	Retroalimentación
1	B	La integridad garantiza que los datos recibidos sean idénticos a los enviados, evitando modificaciones no autorizadas.
2	B	El spyware monitorea actividades en línea, recolectando datos personales para distintos propósitos.
3	A	El nivel 2 de Syslog clasifica los mensajes como errores críticos que requieren atención inmediata.
4	B	Los firewalls stateful monitorean el estado de las conexiones, proporcionando una seguridad más robusta.
5	B	NTP sincroniza los relojes de dispositivos en la red, asegurando precisión en los eventos registrados.
6	B	Las VPNs permiten conexiones seguras mediante cifrado, utilizando la red pública de Internet.
7	A	ESP (Encapsulating Security Payload) protege tanto la confidencialidad como la integridad de los datos.
8	B	El gestor SNMP ejecuta el software de administración y se encarga de supervisar el estado de la red.
9	B	La DMZ separa los servicios accesibles externamente del resto de la red, evitando conexiones directas no autorizadas.
10	B	La protección contra reenvíos garantiza que cada paquete sea único, evitando ataques de repetición.

Ir a la autoevaluación

Autoevaluación 6

Pregunta	Respuesta	Retroalimentación
1	C	Una CDN mejora la experiencia de usuario al reducir la latencia, mejorar la disponibilidad del contenido y aumentar la velocidad de transmisión.
2	A	Los flujos de video almacenado permiten interactividad, lo que mejora la experiencia del usuario al otorgar control sobre la reproducción.
3	B	Distribuir desde un único centro de datos puede generar cuellos de botella, alta latencia y depender de un solo punto, lo que pone en riesgo la disponibilidad del contenido.
4	A	La estrategia de atracción de ISP utiliza clústeres en puntos de intercambio de Internet, lo que permite menores costos de mantenimiento comparado con una distribución más profunda.
5	C	HLS es ampliamente utilizado debido a su compatibilidad con múltiples dispositivos, adaptabilidad y seguridad en la transmisión.
6	A	El buffer del cliente acumula varios segundos de datos antes de la reproducción, asegurando una visualización fluida ante fluctuaciones en la red.
7	A	Los flujos HTTP son compatibles con cortafuegos y redes NAT, mejorando la accesibilidad y reduciendo problemas de conectividad.
8	A	Netflix utiliza una versión personalizada de DASH para ajustar dinámicamente la calidad del video según el ancho de banda disponible.
9	B	La transmisión adaptativa mejora la experiencia del usuario al ajustar la calidad del video de manera continua, evitando interrupciones.
10	C	Las CDN optimizan el uso del almacenamiento al replicar contenido popular en servidores cercanos y eliminar videos menos solicitados cuando el espacio es limitado.

Ir a la autoevaluación

5. Glosario

Capa de Aplicación: Es la capa superior del modelo OSI, responsable de proporcionar servicios de red a las aplicaciones del usuario, gestionando protocolos como HTTP, DNS y SMTP.

HTTP/HTTPS: Protocolo utilizado para la transferencia de datos en la web. HTTPS incluye una capa de seguridad basada en SSL/TLS para proteger la comunicación.

DNS (Sistema de Nombres de Dominio): Servicio que traduce nombres de dominio comprensibles para los humanos (como www.google.com) en direcciones IP.

DHCP (Protocolo de Configuración Dinámica de Host): Protocolo que asigna automáticamente direcciones IP y configuraciones de red a los dispositivos conectados.

VoIP (Voz sobre IP): Tecnología que permite realizar llamadas de voz a través de redes IP, utilizando protocolos como SIP o WebRTC.

Programación de Sockets: Técnica que permite la comunicación entre procesos de red mediante sockets, usando protocolos como TCP o UDP.

Calidad de Servicio (QoS): Conjunto de técnicas que garantizan un rendimiento adecuado de las aplicaciones multimedia mediante la priorización del tráfico.

Flujos Multimedia: Categorías de transmisión de contenido que incluyen flujos almacenados, en tiempo real y en vivo, cada uno con sus propios requisitos de red.

RTP (Protocolo de Transporte en Tiempo Real): Protocolo utilizado para la transmisión de medios de audio y video en aplicaciones de tiempo real, como videoconferencias.

Redes de Distribución de Contenido (CDN): Infraestructura distribuida que mejora la entrega de contenido digital al almacenar copias en servidores cercanos a los usuarios.

Firewall: Sistema de seguridad que filtra y controla el tráfico de red entre dos zonas, permitiendo o bloqueando el acceso según reglas definidas.

VPN (Red Privada Virtual): Tecnología que permite crear una conexión segura y cifrada a través de una red pública, como Internet.

IPsec (Seguridad de Protocolo de Internet): Conjunto de protocolos que proporcionan autenticación, cifrado y protección de integridad en las comunicaciones de red.

Protocolo HLS (HTTP Live Streaming): Protocolo de transmisión multimedia que utiliza HTTP para enviar fragmentos de video con tasa de bits adaptativa, optimizando la experiencia del usuario.

Cloud Networking: Gestión y configuración de redes virtuales en infraestructuras de nube pública o híbrida, permitiendo escalabilidad, flexibilidad y conectividad segura.

6. Referencias bibliográficas

Cisco Packet Tracer . (s. f.). https://www.netacad.com/es/cisco-packet-tracer

Dutt, D. G. (2020). Cloud Native Data Center Networking. *O'Reilly Media* . https://openlibrary.telkomuniversity.ac.id/home/catalog/id/160559/slug/cloud-native-data-center-networking.html

Dynamic Host Configuration Protocol (DHCP) . (s. f.). https://w3.ual.es/~vruiz/Docencia/Apuntes/Networking/Protocols/Level-3/05-DHCP/index.html

EVE-NG Ltd. (2025, 24 enero). *Home* - . https://www.eve-ng.net/

Fernández, A. M. (2023, 24 abril). Netflix genera más del 13% del tráfico de Internet que circula por las redes, casi tanto como YouTube y TikTok juntas. *El Español* . https://www.elespanol.com/invertia/observatorios/digital/20230424/netflix-genera-trafico-internet-circula-youtube-tiktok/757924487_0.html?utm_source=chatgpt.com

Gálvez, J. A. S. (s. f.). *Sockets* . Unidades de Apoyo Para el Aprendizaje - CUAIEED - UNAM. https://www.geeksforgeeks.org/socket-programming-in-java/

GeeksforGeeks. (2024, 15 enero). *Explain the Working of HTTPS* . GeeksforGeeks. https://www.geeksforgeeks.org/explain-working-of-https/

Getting Started with GNS3 | GNS3 Documentation . (s. f.). https://docs.gns3.com/docs/

HTTP y HTTPS: diferencia entre los protocolos de transferencia. AWS . (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/compare/the-difference-between-https-and-http/

IBM i 7.3 . (s. f.). https://www.ibm.com/docs/en/i/7.3?topic=communications-socket-programming

Kurose, J. F., & Ross, K. W. (2017). *Redes de computadoras. un enfoque descendente* (7.a ed.). Pearson.

López, P. (2021, 22 septiembre). *¿Qué es SMTP y para qué sirve?* GEEKNETIC. https://www.geeknetic.es/SMTP/que-es-y-para-que-sirve

Managed GraphQL APIs - Amazon AppSync - AWS . (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/graphql/

Netflix trabaja con los Proveedores de Servicios de Internet (ISP) de todo el mundo para que la experiencia de los espectadores sea extraordinaria - About Netflix . (s. f.). About Netflix. https://about.netflix.com/es/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience

Nsnam. (s. f.). *ns-3* . Ns-3. https://www.nsnam.org/

Opciones de conectividad de red a Amazon VPC - Opciones de conectividad de Amazon Virtual Private Cloud . (s. f.). https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-vpc-connectivity-options/network-to-amazon-vpc-connectivity-options.html

¿Qué son las redes en la nube? - Explicación sobre las redes en la nube - AWS . (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/cloud-networking/

Ranasinghe, N. (2023, 24 julio). The Evolution of HTTP from Version 1.0 to 3.0 and the Impact on Modern Web Communication. *Medium* . https://medium.com/@nirajranasinghe/the-evolution-of-http-from-version-1-0-to-3-0-and-the-impact-on-modern-web-communication-cb233b17e118

RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1 . (s. f.). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc2616

RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2) . (s. f.). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc7540

Sadiqui, A. (2020). *Computer Network Security* . John Wiley & Sons.

Sanz. (2016, 16 diciembre). *Que es un CDN (red de distribución de contenido) | Comunidad Hosting México* . Comunidad Hosting. https://comunidadhosting.mx/que-es-un-cdn-red-de-distribucion-de-contenido/

What is RESTful API? - RESTful API Explained - AWS . (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/restful-api/

Wilbert, M. (2023, 8 agosto). *Protocolos de transmisión de vídeo: 6 formatos preferidos para la radiodifusión profesional* . Dacast. https://www.dacast.com/es/blog-es/protocolo-de-transmision-de-video/

Wireshark · Go Deep . (s. f.). Wireshark. https://www.wireshark.org/