UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE TELECOMUNICACIONES

INFORME FINAL DEL TRABAJO DE INTEGRACIÓN CURRICULAR, PROYECTO DE INVESTIGACÓN

TEMA:

"METODOLOGÍA PARA LA IMPLEMENTACIÓN DE SDN EN EL DATA CENTER DE LA UNIVERSIDAD TÉCNICA DEL NORTE"

Trabajo de titulación previo a la obtención del título de Ingeniera en Telecomunicaciones

Línea de investigación: Desarrollo, aplicación de software y cibersecurity (seguridad cibernética

AUTORA:

Grijalva Torres Ana Carolina

DIRECTOR:

Msc. Domínguez Limaico Hernán Mauricio

Ibarra, 2025

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004089676		
APELLIDOS Y NOMBRES:	GRIJALVA TORRES ANA CAROLINA		
DIRECCIÓN:	ATUNTAQUI, CALLE PICHINCHA Y ALEJANDRO ANDRADE		
EMAIL:	acrijalvat@utn.edu.ec		
TELÉFONO FIJO:	062909340	**TELF. MOVIL**	0991679236

DATOS DE LA OBRA	
TÍTULO:	METODOLOGÍA PARA LA IMPLEMENTACIÓN DE SDN EN EL DATA CENTER DE LA UNIVERSIDAD TÉCNICA DEL NORTE
AUTOR (ES):	ANA CAROLINA GRIJALVA TORRES
FECHA:	3 de febrero de 2025
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	■ **GRADO** □ **POSGRADO**
TITULO POR EL QUE OPTA:	INGENIERA EN TELECOMUNICACIONES
DIRECTOR:	MSC. HERNÁN MAURICIO DOMÍNGUEZ LIMAICO
ASESOR:	MSC. CARLOS ALBERTO VÁSQUEZ AYALA

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, GRIJALVA TORRES ANA CAROLINA, con cédula de identidad Nro. 1004089676 en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 3 días del mes de febrero de 2025

EL AUTOR:

.....

Grijalva Torres Ana Carolina

CONSTANCIAS

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

Ibarra, a los 3 días del mes de febrero de 2025

EL AUTOR:

.....

Grijalva Torres Ana Carolina

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 3 de febrero de 2025

MSC. HERNÁN MAURICIO DOMÍNGUEZ LIMAICO

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f) ……

Msc. Hernán Mauricio Domínguez Limaico

C.C.: 1002379301

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular "METODOLOGÍA PARA LA IMPLEMENTACIÓN DE SDN EN EL DATA CENTER DE LA UNIVERSIDAD TÉCNICA DEL NORTE" elaborado por GRIJALVA TORRES ANA CAROLINA, previo a la obtención del título de INGENIERA EN TELECOMUNICACIONES, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f):.....
Msc. Hernán Mauricio Domínguez Limaico
C.C.: 1002379301

(f):.....
Msc. Carlos Alberto Vásquez Ayala
C.C.: 1002424982

DEDICATORIA

Este trabajo de titulación es el resultado de un camino lleno de aprendizajes, retos y momentos que han forjado quién soy. Lo dedico a todas las personas y seres que han dejado una huella imborrable en mi vida, siendo mi apoyo y motivación constante.

A Dios, quien ha sido mi guía en cada paso que he dado. Su presencia ha llenado mi vida de propósito, brindándome fortaleza y sabiduría para superar cada desafío.

A mis padres, Jaime Grijalva y Teresa Torres, quienes con su amor incondicional, esfuerzo incansable y valores sólidos me han enseñado que los sueños se logran con dedicación. Gracias por ser mi pilar y ejemplo de vida.

A mi hermana Teresa Fernanda, mi cómplice y compañera en este viaje. Tus palabras de aliento y tu confianza en mí han sido un faro en los momentos más oscuros. Este logro es una muestra del apoyo mutuo que siempre nos ha unido.

De forma especial, dedico este trabajo a mi querida tía Beatriz Torres, mi ángel en el cielo. Aunque físicamente ya no estés, tu amor, tus enseñanzas y tus valores han marcado profundamente mi vida. Gracias por enseñarme la importancia del trabajo honesto y la bondad. Tu memoria ha sido una fuerza que me ha impulsado a seguir adelante y alcanzar este sueño.

Finalmente, a mis más fieles compañeros : Tony, Zeus, Luna y Lucas. Su amor desinteresado y su compañía incondicional me han brindado alegría y consuelo en los días más difíciles.

A todos ustedes, con infinito cariño y gratitud, dedico este logro que representa el cierre de una etapa y el inicio de nuevos sueños.

AGRADECIMIENTO

Al culminar este capítulo lleno de aprendizajes, quiero expresar mi profundo agradecimiento a quienes han sido fundamentales en mi vida y en la realización de este trabajo:

En primer lugar, agradezco a Dios, quien guía mi vida por caminos de bien y justicia, brindándome fortaleza y sabiduría en cada paso.

A mis padres, Jaime Grijalva y Teresa Torres, por ser el pilar de mi vida. Gracias por su incansable apoyo, por su ejemplo de perseverancia y trabajo honesto, y por enseñarme que con esfuerzo y dedicación todo es posible. Sus sacrificios y amor incondicional han sido mi mayor inspiración para alcanzar mis metas.

A mi hermana, Fernanda Grijalva, por ser mi compañera de vida, mi apoyo incondicional, y aquella que siempre me escucha y me motiva. Gracias por confiar en mí y ser una fuente constante de fortaleza. Sin ti, no sería la persona que soy hoy.

A mis amigos Sofía, David, Anahí, Jorge, Keneth, y Erick, así como a todos los que han compartido este camino conmigo. Su amistad, compañía y los momentos compartidos han marcado mi vida de manera imborrable.

De forma especial, expreso mi gratitud al MSc. Mauricio Domínguez y al MSc. Carlos Vásquez, por su tiempo, dedicación, y valiosos consejos durante el desarrollo de este trabajo de titulación. Su orientación fue crucial para convertir este proyecto en una realidad.

Finalmente, agradezco al Departamento de Desarrollo Tecnológico e Informático de la Universidad Técnica del Norte por su apertura y apoyo en el desarrollo de esta investigación, permitiéndome trabajar en un entorno enriquecedor y colaborativo.

RESUMEN EJECUTIVO

La migración de la arquitectura de red tradicional del datacenter de la Universidad Técnica del Norte hacia una basada en redes definidas por software responde a la necesidad de adoptar tecnologías modernas que optimicen la gestión de recursos, mejoren la escalabilidad y faciliten el manejo del tráfico en redes complejas. El objetivo principal fue desarrollar una guía metodológica que permita al DDTI planificar y ejecutar esta transición, garantizando la continuidad operativa y la integración eficiente de nuevos componentes. Se adoptó un enfoque descriptivo y práctico, que incluyó el análisis de la infraestructura actual, la identificación de equipos compatibles con tecnologías SDN y la evaluación de los requisitos de hardware y software. Entre los resultados se destaca la incorporación de OpenDaylight como controlador central, la configuración del protocolo OpenFlow para la gestión de reglas de flujo, y la integración de un router físico para abordar las limitaciones en el manejo de IPv6, lo que garantiza una red funcional en dual-stack. A partir de la investigación realizada, se determinó que la implementación de esta arquitectura permite una administración más eficiente, reduce la dependencia de configuraciones manuales y establece las bases para futuras actualizaciones tecnológicas, consolidándose como una solución escalable y adaptable a las necesidades actuales y futuras de la institución.

Palabras clave: SDN, OpenFlow, Datacenter, migración, OpenDaylight.

ABSTRACT

The migration from the traditional network architecture of the Universidad Técnica del Norte's datacenter to one based on software-defined networks responds to the need to adopt modern technologies that optimize resource management, improve scalability and facilitate traffic management in complex networks. The main objective was to develop a methodological guide that allows DDTI to plan and execute this transition, ensuring operational continuity and efficient integration of new components. A descriptive and practical approach was adopted, which included the analysis of the current infrastructure, the identification of equipment compatible with SDN technologies and the evaluation of hardware and software requirements. The results include the incorporation of OpenDaylight as a central controller, the configuration of the OpenFlow protocol for flow rule management, and the integration of a physical router to address limitations in IPv6 handling, ensuring a functional dual-stack network. From the research conducted, it was determined that the implementation of this architecture allows for a more efficient administration, reduces dependence on manual configurations and establishes the basis for future technological upgrades, consolidating itself as a scalable and adaptable solution to the current and future needs of the institution.

Keywords: SDN, OpenFlow, Datacenter, migration, OpenDaylight.

LISTA DE SIGLAS

DDTI. Dirección de Desarrollo Tecnológico e Informático

SDN. Software Defined Networking

VXLAN. Virtual Extensible LAN

VLAN. Virtual Local Area Network

WAN. Wide Area Network (Red de Área Amplia)

OpenFlow. Protocolo utilizado en redes definidas por software

DMZ. Demilitarized Zone (Zona desmilitarizada)

IPv6. Internet Protocol version 6

IPv4. Internet Protocol version 4

RADIUS. Remote Authentication Dial-In User Service

SSH. Secure Shell

HX. HyperFlex (tecnología de Cisco para infraestructura hiperconvergente)

RAID. Redundant Array of Independent Disks (Conjunto Redundante de Discos Independientes)

RAM. Random Access Memory (Memoria de Acceso Aleatorio)

SSD. Solid State Drive (Unidad de Estado Sólido)

VMware. Software de máquinas virtuales

NGFW. Next-Generation Firewall (Cortafuegos de Nueva Generación)

FortiOS. Sistema operativo de Fortinet

CEDIA. Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia

ONF. Open Networking Foundation

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I

Antecedentes

1.1 Problema de investigación

Las redes de datos se encuentran en constante evolución, así como los equipos de red que llevan décadas de desarrollo e implementación en diferentes entornos, más estos avances han venido acompañados de la constante necesidad de brindar mayores capacidades y velocidades para los datos, por lo que, sobre todo en los centros de datos requieren de una nueva tecnología que permita satisfacer estas necesidades, como una respuesta a este planteamiento surgen las redes definidas por software (SDN) (Goransson & Black, 2014). Muchos de los problemas que enfrentan los centros de datos actualmente se enfocan únicamente al área física de los mismos, pues se toman consideraciones sobre el espacio útil y como la conexión de nuevos equipos puede traer complicaciones en la gestión energética y de temperatura, ambos factores cruciales para el buen funcionamiento de los equipos de networking (Jennings Michael, 2022), más no se considera que los equipos de redes tradicionales se enfrentan a problemas de capacidad y asignación de recursos, pues con los grandes flujos de tráfico que deben gestionar los centros de datos en situaciones de congestión la provisión de servicios se ve amenazada atentando contra la continuidad del servicio ya que pueden existir fluctuaciones en la capacidad de los enlaces debido al consumo heterogéneo del ancho de banda (Kant, 2009).

Los data center son cada vez más grandes y abundantes con tendencias hacia la virtualización y la expansión de la sociedad de la información, este crecimiento viene acompañado de la necesidad de soportar cada vez mayores cantidades de tráfico y por ende de una asignación de recursos apropiada, por lo que la gestión tradicional de los

centros de datos tiende a quedarse corta en cuanto a la prestación de determinados parámetros como el rendimiento de la red, pues en base a lo expuesto por (Paliwal & Shrimankar, 2019) las arquitecturas de red tradicionales no permiten realizar una adecuada gestión energética, limitan la escalabilidad o capacidad de adaptación y la optimización es baja ya que en condiciones de congestión se generan retardos considerables que afectan directamente a la utilización del ancho de banda de la red. En base a estos planteamientos se establece que las redes definidas por software pueden representar una solución apropiada a estas deficiencias (Sherwin & Cormac, 2021).

Sin embargo, no se puede realizar una migración de una tecnología tradicional a una de redes definidas por software de forma precipitada, pues si bien la implementación de esta arquitectura puede parecer sencilla, realizar este proceso debe respaldarse con un estudio del coste, rendimiento y disponibilidad que se busca dar a los servicios que se albergan en el data center, ya que los mismos no deben verse afectados por la transición, por el contrario deberían optimizar su funcionamiento y ofrecer mejores resultados en la implementación. Por esta razón se considera que, al no contar con una metodología que se adapte a los requerimientos de la arquitectura de la red este proceso no se puede llevar cabo de manera óptima, ya que, el análisis de los elementos de red actuales, su compatibilidad con protocolos vinculados a las redes definidas por software y el soporte respectivo para los mismos no ha sido considerado, y que en consiguiente puede traer consigo varios imprevistos que frenen el desarrollo del proyecto de migración y que incluso afecten a los servicios que presta el data center (OPEN NETWORKING FUNDATION, 2014). Tomando en consideración que la implementación de redes definidas por software requiere de una reestructuración de la red para admitir la incorporación de una controladora que

administre y gestione los procesos de la red, y que este a su vez se acople a un protocolo de comunicación entre la controladora y los equipos de red, la necesidad de un plan estructurado que detalle estas características se vuelve más fuerte y la falta de este representa un problema mucho mayor.

Actualmente dentro de La Universidad Técnica del Norte (UTN) se cuenta con un data center de arquitectura tradicional para respaldar servicios y operaciones propios de la institución, sin embargo, considerando la tendencia evolutiva que siguen las redes con la finalidad de ofrecer una mayor flexibilidad, escalabilidad y eficiencia se plantea el escenario de migración hacia una arquitectura de red definida por software (SDN).

La Universidad Técnica del norte (UTN) no cuenta al momento con un plan definido y estructurado para realizar la migración del data center a una estructura de red definida por software (SDN). La propuesta de diseñar una guía metodológica enfocada a la futura migración del data center permitirá que el personal del departamento de TI que es el encargado de la gestión de este, tenga un panorama más claro de la orientación que se debe tomar para lograr este cometido, así como analizar las ventajas y los procesos a considerar para dar este paso.

El presente trabajo se centrará en la investigación y la revisión de los requerimientos tanto de software como hardware para migrar a una arquitectura de red definida por software (SDN) que se acople a las funciones actuales del data center y que permita mejorar el rendimiento del mismo finalmente se realizará una guía metodológica que detalle de forma clara a los hallazgos de la investigación realizada para de esta forma apoyar al departamento de TI en el proceso de migración en caso de que este se lleve a cabo en el futuro.

1.2 Justificación

El presente proyecto surge con la finalidad de aportar una solución que permita gestionar de forma más eficiente los recursos de la red del data center de la Universidad Técnica del Norte. La implementación de redes definidas por software se presenta como una alternativa viable para este fin, debido a que las redes definidas por software ofrecen una gran flexibilidad en la asignación de recursos en base a la utilización de estos para lograr así mayor eficiencia en los procesos de conmutación y reducen los costos operativos que se mantienen comúnmente dentro de los data centers convencionales (Jennings Michael, 2022). Por lo que, para lograr una migración exitosa hacia esta nueva tecnología, es fundamental desarrollar un estudio detallado basado en protocolos, documentaciones de propuestas similares y sobre todo en de la arquitectura de la red actual, para de este modo, considerar de los elementos claves a considerar para mantener la continuidad en los servicios prestados por el data center y que además permitan incorporar mayores facilidades al gestionar la red, tomando en cuenta que se trata de una tecnología que se halla en constante evolución y que a su vez no se ha dado a conocer de todo en el contexto local es necesario que el proceso se documente de la forma adecuada para sentar las bases correctas para este proceso.

Considerando los objetivos de desarrollo sostenible (ODS) planteados por la Organización de las Naciones Unidas como una guía para determinar el aporte que se realiza con este proyecto, se toma como punto de partida el noveno objetivo: “Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación”, mismo que abarca de forma general en una de sus metas el apoyo al desarrollo tecnológico en el entorno nacional y el acceso

universal a las tecnologías de la información (ONU, 2023). Al hablar de redes definidas por software, este objetivo permite abarcar de forma precisa el concepto de la innovación y el acceso a las tecnologías de la información, debido a que esta tecnología tiene como finalidad mejorar las comunicaciones desde su raíz y a su vez actualizar las arquitecturas tradicionales; por otro lado, al enfocar el estudio de las SDN en un contexto local se enmarca con facilidad en la meta del desarrollo tecnológico en el entorno nacional.

Considerando también la Agenda de Transformación Digital del país comprendida para los años 2022 a 2025 se pone a consideración la integración de nuevas tecnologías y los objetivos de infraestructura digital, el abordar este proyecto de investigación permite aportar de forma positiva al cumplimiento de los objetivos referentes a los sistemas de la información, pues al establecer la necesidad de plantear mejoras tecnológicas a las infraestructuras de red se presenta la iniciativa de realizar cambios sustanciales en la definición actual de las redes en centros de datos en el contexto nacional (Meza et al., 2022).

Tomando como referencia las líneas de investigación manejadas en la Universidad Técnica del Norte, se puede enmarcar al presente proyecto de integración curricular como parte de la rama de Innovación Tecnológica y de productos que se maneja actualmente en la carrera de Telecomunicaciones, misma que a su vez se inclina por la sub línea de Networking que se halla enfocada en la búsqueda de soluciones que permitan mejorar la infraestructura de red. Con este fin, se toma el programa de cloud computing que abarca de forma general los conceptos de infraestructuras como servicios y redes definidas por software.

1.3 Objetivos

1.3.1 Objetivo General

- Desarrollar una guía metodológica enfocada a nuevas tecnologías de red mediante el estudio de la infraestructura de red actual del data center de la Universidad Técnica del Norte que permita su migración a una arquitectura de redes definidas por software.

1.3.2 Objetivos Específicos

- Establecer el estado del arte sobre los conceptos de redes definidas por software, las tecnologías y estándares empleados, así como sus principios de aplicación en centros de datos.
- Realizar una evaluación de la infraestructura de red actual del data center de la Universidad Técnica del Norte estableciendo sus características, funciones y los beneficios de la implementación de redes definidas por software.
- Definir los requerimientos de software y hardware necesarios para la migración de la estructura de red tradicional a una red definida por software manteniendo la funcionalidad actual del data center.
- Elaborar una guía metodológica detallada sobre los puntos clave que permitan la migración del data center a SDN y sirva de referencia al departamento de TI en su futura implementación.

CAPÍTULO II

Fundamentación Teórica

2.1 Software Defined Networks

Las redes tradicionales que se mantienen vigentes son el resultado de una evolución en la que el hardware se consideraba el núcleo principal, dejando al software en segundo plano. Sin embargo, con el avance de la tecnología y la constante demanda, esta estrategia dejó de ser viable. Por ello, se optó por utilizar un software de red mucho más estructurado, que se basa en la aparición de protocolos y tecnologías que permiten organizar las redes en pilas de capas lo cual simplifica el proceso de comunicación (Tanenbaum & Wetherall, 2012). Sin embargo, estas redes han dejado de ser tan eficientes como en el pasado, y las estructuras desarrolladas con anterioridad requieren de nuevos protocolos que permitan ofertar un mejor rendimiento.

Según (Stallings, 2014), una red definida por software (SDN) puede entenderse como una técnica de organización de redes en la que se realiza la separación de las funciones de plano de datos y plano de control de los dispositivos de red como routers, switches o firewalls. Se considera entonces que, una SDN plantea una abstracción de las capas de una red para volverla más flexible y mejorar su rendimiento mejorando los tiempos de respuesta al facilitar que un administrados de red moldee el flujo de tráfico desde una controladora centralizada para dar instrucciones a los dispositivos de red (Rosencrance et al., 2022).

Según (Haleplidis et al., 2015) en el RFC 7426 se define a los dispositivos de red como un equipo que realiza una o varias funciones de red relacionadas con el reenvío y manejo de paquetes, y este puede ser un contenedor de recursos o ser un

recurso por sí mismo. Los dispositivos de red convencionales se hallan conformados por el plano de datos y el plano de control, mismos que se encuentran integrados en un mismo hardware y cumplen funciones de control y gestión de la red o de transporte y conmutación de datos (Stallings, 2014). Para comprender de mejor forma el funcionamiento de una red definida por software es necesario conocer las funciones del plano de control y del plano de datos dentro de un dispositivo de red, la **Tabla 1.** Características de los planos de datos y de control, presenta de forma general la funcionalidad de estos dos planos.

Tabla 1.

Características de los planos de datos y de control

	Plano de datos	**Plano de control**
Características	Encargado del reenvío de paquetes en base a información del plano de control.	Administra información de las interfaces de red.
	Proporciona rutas de alta velocidad y baja latencia.	Gestiona creación y actualizaciones de tablas de enrutamiento.
	Realiza la revisión de direcciones destino de la cabecera IP.	Permite implementación de políticas de seguridad.
	Aplica las políticas establecidas por el plano de control.	Administra protocolos de capa de red.
	Depende de la lógica del plano de control.	Su funcionamiento es independiente del plano de datos.

Fuente: Adaptado de (Awati, 2023)

En base a lo expuesto anteriormente es posible definir que, el plano de datos va a depender siempre de lo que el plano de control establezca, por lo que, una red definida por software busca separa estos elementos que se encuentran trabajando en conjunto dentro de cada dispositivo de red, SDN es un concepto que se lleva trabajando desde los años 90, en los que se estableció la búsqueda de la programabilidad en los dispositivos de red en base a una lógica de control que

logre dicha separación, así como introducir un controlador centralizado que contenga información de todos los recursos disponibles en la red para lograr una mejor utilización del ancho de banda así como garantizar la eficiencia de las transmisiones de tráfico diferenciado por la red (Huang et al., 2017).

Dentro de una red tradicional, se considera la existencia de plano de datos, de control y administración en un mismo dispositivo de red (Rodríguez Herlein et al., n.d.). Estos planos emplean tablas de enrutamiento y tablas de conmutación para determinar rutas por las cuales se enviarán los paquetes en la red. En este contexto, cada dispositivo cuenta con su propia lógica y autonomía de reenvío. De este modo, se considera que, cuando un paquete llega a un equipo de red como un router, este realiza un análisis de la cabecera del protocolo de red para identificar la dirección destino y consultar a su tabla de enrutamiento para determinar los posibles caminos que pueda tomar el paquete para ser reenviado por toda la red. Este proceso se repite para cada nuevo paquete que llegue por una de sus interfaces y en cada equipo intermedio hasta alcanzar su destino.

Por otro lado, en una red definida por software, el proceso de reenvío de paquetes se desvincula del plano de control, estableciendo así que la toma de decisiones se desarrolla en un equipo diferente que opera como controlador de la red. Este controlador se comunica con los demás elementos mediante un protocolo de SDN que actúa como interfaz de control (Bernal & Mejía, 2016).

En una SDN, cuando un paquete llega a un dispositivo de red, este envía la información sobre el paquete al controlador asignado. El controlador toma las decisiones sobre el reenvío del paquete, actualiza las tablas de enrutamiento de todos los dispositivos y coordina el reenvío del paquete en base a las decisiones tomadas por el controlador. Esto permite reenvío dinámico y adaptable a los

posibles cambios en la topología de la red logrando así un manejo más eficiente del tráfico, debido a que cuando el controlador SDN recibe información de más de un paquete dirigido al mismo destino puede tomar decisiones de reenvío iguales o puede optar por nuevas rutas en base a las políticas y la carga de la red actualizando continuamente las tablas de flujo de los dispositivos de red.

Con base en los principios establecidos para las SDN, han emergido protocolos como OpenFlow, OpenDaylight y OnePK, los cuales facilitan la configuración de cómo se transmiten las reglas y políticas desde los controladores hacia los dispositivos de red. Esto posibilita que estos últimos cumplan con sus funciones de reenvío del tráfico a lo largo de la red. OpenFlow se puede definir como una interfaz estándar diseñada específicamente para SDN. Esta interfaz posibilita la separación clara entre el plano de control y el plano de datos, permitiendo que los dispositivos se centren únicamente en las tareas de transporte, mientras que el controlador centralizado toma decisiones de reenvío (Nadeau & Gray, 2013), este protocolo es altamente empleado en diferentes campos como es el caso de redes empresariales y centros de datos, donde se busca mejorar la eficiencia y flexibilidad de las redes.

En base a la primicia del funcionamiento de una red definida por software, es posible distinguir cuatros tipos o modelos de redes definidas por software, mismos que son:

- SDN abierta: constituyen las redes definidas por software en las que se hace uso de protocolos de código abierto como OpenFlow para el control de dispositivos de red (VMware, 2020).

- SDN API[1]: redes definidas por software que hacen uso de APIs para la gestión del tráfico en la red en lugar de protocolos de código abierto (VMware, 2020).
- Modelo de superposición de SDN: este modelo de redes hace uso de túneles dinámicos para ejecutar redes virtuales sobre infraestructuras de red existentes asignando el ancho de banda y los recursos en varios canales para no afectar a esta infraestructura (Roger, 2022).
- SDN híbrida: modelo de red definida por software que integra los principios de SDN con protocolos de networking tradicionales, en este modelo de redes el control del movimiento del tráfico se comparte entre el controlador SDN y los protocolos de red estándar, permitiendo así una introducción paulatina de la SDN en un entorno con hardware heredado[2] (VMware, 2020).

2.1.1 Arquitectura

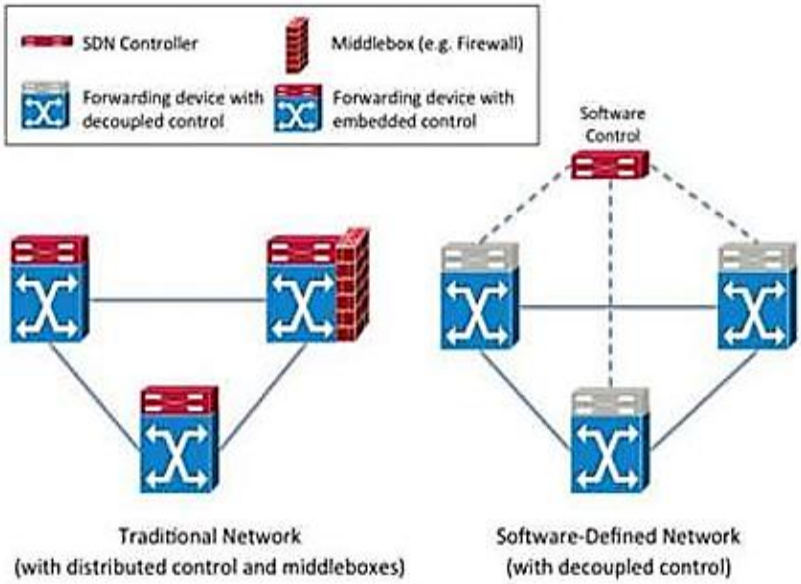
La principal diferencia de una red tradicional y una red definida por software se encuentra en su arquitectura, pues en una red definida por software existe una interfaz de usuario versátil que permite que los administradores de red puedan escribir y definir políticas de servicio de red mediante el uso de lenguajes de programación de alto nivel, algo que no era posible de implementar en los dispositivos de red tradicionales (Kumar et al., 2019), así también se puede decir que la distribución de red centralizada que se maneja en SDN no es la misma que

[1] API: Appication Programming Interface o Interfaz de programación de aplicaciones es un conjunto de protocolos y herramientas diseñadas para la comunicación entre aplicaciones de software.(Segmentify, 2023).
[2] Hardware heredado: término que define a dispositivos de hardware que pueden seguir en uso pero que se vuelven obsoletos pero que no pueden ser reemplazados fácilmente (Icy Science, 2023).

se gestiona en el networking convencional, como ejemplo se tiene la **Figura 1** en la cual se puede distinguir evidenciar como la interconexión de quipos y los componentes de la red cambian al usar una organización diferente en la red, pues en la red tradicional es necesario adjuntar varios equipos con contadores embebidos propios mientras que, en SDN todos los equipos de forwarding se conectan a un mismo controlador.

Figura 1.

Arquitectura de red tradicional vs Arquitectura de SDN



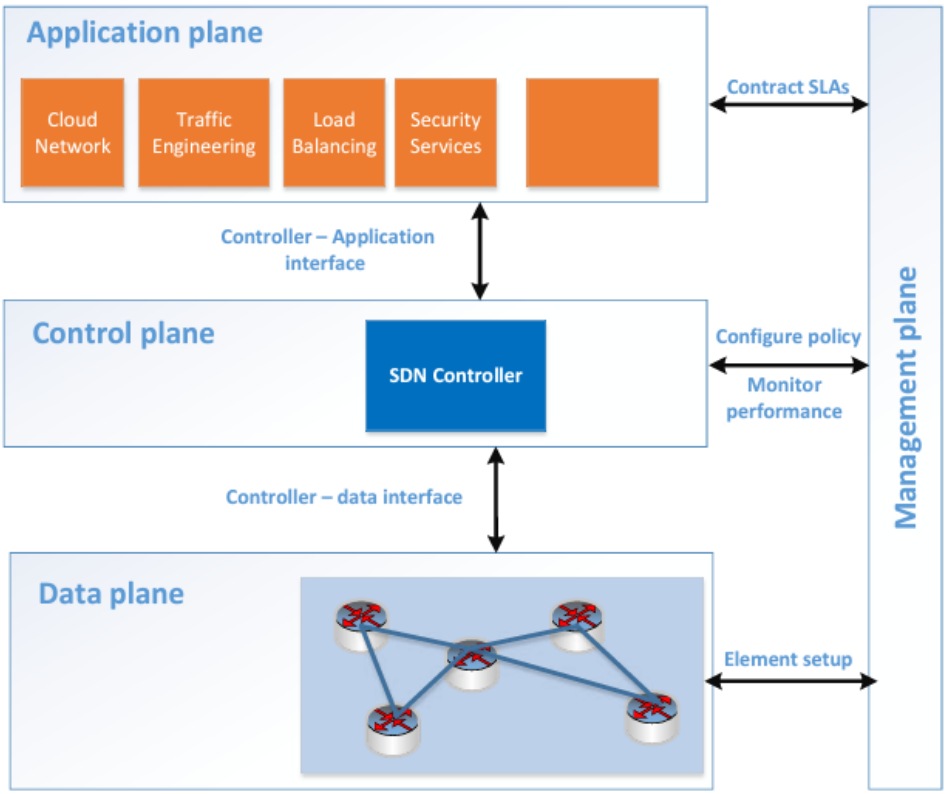
Fuente: Obtenido de (Kumar et al., 2019)

La arquitectura de una red definida por software se puede definir en tres capas principales las cuales se denominan de abajo hacia arriba como: capa de infraestructura o de datos, capa de control y capa de aplicación (Schaller & Hood, 2017), en esta arquitectura se puede definir de cierta forma al plano de control como el cerebro o núcleo de la red, ya que como se puede apreciar en la **Figura 2** esta se encuentra conectada directamente a la capa de aplicación mediante la

llamada interfaz North-Bound[3] y a la capa de infraestructura o datos mediante una interfaz South-Bound [4]que se encarga de la traducción de las instrucciones de alto nivel a comandos de bajo nivel (Cabaj et al., 2014).

Figura 2.

Arquitectura de una red definida por software



Fuente: Obtenido de (Cabaj et al., 2014)

Cada uno de los planos que comprenden la arquitectura básica de una red definida por software cumple con una función específica, existe un cuarto plano dentro de la arquitectura como se evidenció con anterioridad en la **Figura 2.** Arquitectura de una red definida por software , el llamado plano de gestión, este

[3] Interfaz North-Bound: Sistema de software que construye aplicaciones de red en lenguajes de programación como Java o Pyhton para simplificar la programación de la red (Hend Abdelgader et al., 2019).

[4] Interfaz South-Bound: Conexión de enlace entre controlador y dispositivos de forwarding, establecen el método de interacción entre plano de datos y plano de control (Hend Abdelgader et al., 2019).

es aquel que gestiona de forma exterior al plano de control, este no es visible a los usuarios y se encarga de llevar tareas como instalación y configuración de parámetros de la red, este no debe ser programable para mantener la seguridad en la red y es común que se considere en muchos casos como parte del plano de control (Cabaj et al., 2014), las funciones propias de cada uno de los planos son:

- Plano de datos: se compone por los equipos de forwarding o reenvío de tráfico como routers y switches, su objetivo se centra en el reenvío del flujo de tráfico de la red mediante el uso de tablas de enrutamiento o reenvío (Hend Abdelgader et al., 2019).
- Plano de control: es el controlador o núcleo de la red, se encarga de la toma de decisiones del reenvío del flujo de tráfico y gestiona la comunicación de los lenguajes de alto nivel empleados en el plano de aplicación con los comandos de bajo nivel requeridos por los equipos de forwarding, generalmente trabaja de forma centralizada indicando a los dispositivos del plano de datos políticas, actualizaciones e información suministrada por la red física (Hend Abdelgader et al., 2019).
- Plano de aplicación: formado por aplicaciones de servicio de red, mismas que permiten realizar funciones de gestión de red y asignación de recursos pues, son la interfaz por la cual el administrador de red envía instrucciones hacia el controlador de la red SDN para realizar acciones de gestión o balanceo de carga en la red (Hend Abdelgader et al., 2019).

En síntesis, para una red definida por software, la definición de un controlador de red es crucial, ya que la misma se encarga de comunicar a los

dispositivos de reenvío y a los administradores de red para de esta forma lograr el mejor resultado posible en lo que respecta a la organización y asignación de recursos para el funcionamiento óptimo de la red (Kreutz et al., 2015)

2.1.2 Controladores SDN

El controlador de una SDN es el punto central y el cerebro de la organización de la red en cuestión, pues la funciones de este se centran en la gestión del flujo de datos, automatización de tareas de gestión, aprovisionamiento de recursos, configuración de rutas, visibilidad y monitoreo de la red, entre otras. Los controladores pueden ser propietarios como el caso de Cisco, Juniper Networks, Vmware o Nokia, o de código abierto como el caso de OpenDaylight, ONOS o Tungsten Fabric (codilime, n.d.). Los controladores pueden categorizarse en dos clases:

- Controladores centralizados: esta categoría de controladores se caracteriza por implementar toda la lógica del plano de control en un único punto, haciendo que un mismo servidor gestione todas las acciones del plano de control simplificando la gestión, sin embargo, su mayor desventaja se encuentra en limitación de la escalabilidad y al hecho de tener un punto de falla fatal (Paliwal et al., 2018).
- Controladores distribuidos: la segunda categoría de controladores se basa en la subdivisión del control de la red en diferentes equipos o servidores. Estos presentan una mayor escalabilidad y una alta tolerancia a fallos debido a su naturaleza distribuida lo que permite incrementar la seguridad en las redes al dividir la información sensible en diferentes equipos (Paliwal et al., 2018).

Gracias al avance y a la popularización la estructura de organización de redes definidas por softwares, los desarrolladores del área de networking han desarrollado diversas versiones de controladores que ofrecen diferentes prestaciones y que pueden acoplarse a distintas aplicaciones de redes, la **Tabla 2.** Comparación entre controladores de SDN que se plantea a continuación muestra una comparativa de las características más relevantes de algunos de los controladores de red más empleados en la actualidad, misma que se basa en diferentes parámetros.

Tabla 2.

Comparación entre controladores de SDN

Controlador	Asociación	Campo de aplicación	Sistema operativo	Lenguaje de programación	North-Bound	South-Bound	GUI	Flows/s	Documentación	Categoría
ONOS	At&T, Ciena,Cisco, Ericsson,Fujitsu, Huawei,Intel, Nec, Nsf.Ntt Communication, Sk Telecom, ON.LAB	Datacenter, WAN y redes de transporte.	Linux, MAC OS, Windows	Java	REST API	OF1.0, 1.3, NETCONF	Basada en WEB	1M	Buena	Distribuida
OpenDay-Light	Linux Foundation With Memberships Covering Over 40 Companies, Such As Cisco, IBM, NEC	Datacenter	Linux, MAC OS, Windows	Java	REST API	OF1.0, 1.3, 1.4, NETCONF/ YANG, OVSDB, PCEP, BGP/LS, LISP, SNMP	Basada em Web	106K	Muy Buena	Centralizada
NOX	Nicira	Campus	Mayor soporte en Linux	C++	Ad-Hoc API	Nicira	Python + QT4	1.8M	Escasa	Centralizado
RYU	Nippo Telegraph And Telephone Corporation	Campus	Mayor soporte en Linux	Python	REST API	OF 1.0, 1.2, 1.3, 1.4, NETCONF,	GUI BASIC	------	Aceptable	Centralizada

Bacon	Universidad de Standford	Investigación	Linux, MAC OS, Windows	Java	Ad-Hoc API	OFCONFIG OF 1.0	Basada en Web	12.8 M	Aceptable	Centralizada
Onix	Nicira Networks, Google, NEC, ICSI	Campus	Mayor soporte en Linux	C, Python	REST API	OF 1.0	Basada en Web	2.2 M	Escasa	Centralized
Maestro	RICE, NSF	Investigación	Linux, MAC OS, Windows	Java	REST API	OF 1.0	Sin GUI	4.8 M	Poca	Centralizada

Fuente: Adaptado de (Salman et al., 2016)

2.2 SDN Data Centers

Los data centers se crearon como una forma de facilitar la gestión de las grandes redes enfocadas a la prestación de servicios dedicados así como realizar una separación física de los dispositivos de red tradicionales y su almacenamiento asociado de las redes de interconexión con clientes finales (Nadeau & Gray, 2013), de esta forma los centros de datos fueron ganando mayor relevancia dentro de los empresas al punto de que cada una gestiona su red desde un centro específico en el cual mantienen equipos como: servidores, conmutadores de tramas, routers, dispositivos de seguridad (firewalls) y almacenamiento, entre otros; sin embargo, la constante demanda y la expansión de la sociedad de la información ha generado la necesidad de cambiar la organización de las redes que gestionan grandes flujos de tráfico de su estructura tradicional a una que permita gestionar de mejor manera los recursos para de esta forma agregar flexibilidad, escalabilidad y mejorar el rendimiento (Kant, 2009).

2.2.1 Requerimientos

Los data centers ha evolucionado junto con los requerimientos y necesidades para poder responder a las exigencias de las redes actuales, razón por la que el cambio de arquitectura a una SDN puede traer grandes ventajas

incluyendo un mejor rendimiento para los altos flujos de datos que deben gestionar los centros de datos sobre todo en entornos empresariales, algunos de estos requerimientos son:

- Automatización: debido a los constantes cambio y a la tendencia de las redes a expandirse o contraerse las redes deben volver más dinámicas y ágiles para poder adaptarse a estos cambios con la mínima interacción humana posible permitiendo dar mayor independencia a la red y estableciendo así una respuesta más rápida ante la necesidad de reasignación de recursos o de cambios significativos en la topología para mantener así la disponibilidad de los centros de datos en múltiples circunstancias (G̈ransson & Black, 2014).
- Escalabilidad: con el constante crecimiento de los centros de datos, las tablas CAM, de enrutamiento, números de VLANs han repercutido en una limitación importante para el crecimiento de las redes, razón por la cual los data center tienen la necesidad de hallar soluciones que permitan eliminar estas limitaciones como es el uso de túneles de redes virtuales o de nuevas implementaciones de organizaciones que aporten una mejora al inconveniente planteado (G̈ransson & Black, 2014).
- Virtualización de la red: debido a demandas de automatización, arrendamiento múltiple y escalabilidad la virtualización de la red se da poco a poco, esto debido al uso de servidores y almacenamiento virtualizados que han ido tomando lugar en los centros de datos, generando así la necesidad de generar una abstracción virtual de la red que opere sobre la red física para poder responder a las demandas de esta nueva tendencia (G̈ransson & Black, 2014).

- Recuperación de fallos: debido a la gran escala de los centros de datos el proceso de recuperación de fallos es un tarea compleja y delicada por lo que la necesidad de contar con una visión completa de la red y de simplificar el proceso de recuperación de fallos se vuelve una necesidad importante dentro de cualquier data center ya que el objetivo de estos se centra muchas veces en ofertar alta disponibilidad (Göransson & Black, 2014).
- Adición, traspaso y eliminación de recursos: para los centros de datos las actualizaciones en la red tener una rápida respuesta y adaptación, sobre todo al contar con servidores virtualizados, ya que se requiere de un gran sincronismo para que los cambios no generen estragos en el rendimiento de la red, por esta razón el proceso de añadir, quitar o reasignar recursos deben acoplarse a un sistema de automatización que se anticipe a los procesos y pueda responder adecuadamente a estas solicitudes (Göransson & Black, 2014).
- Arrendamiento múltiple: existen muchas empresas que usan sus centros de datos para arrendar espacio ya sea físico o lógico dentro de sus servidores, por lo que la importancia de mantener el aislamiento y la individualidad de los clientes se vuelve crucial en estos centros de datos, por lo que la segregación del tráfico es una necesidad no solo de los recursos asignados sino también de seguridad para mantener el estado de la red y la calidad de servicio ofertado a los usuarios finales (Göransson & Black, 2014).
- Ingeniería de tráfico: debido al crecimiento de los data centers es necesario que se dé un mejor uso de los recursos existentes en la red, es

así que el uso de herramientas que permitan comprender de mejor forma las cargas de tráfico para que así se puedan tomar las decisiones más adecuadas para el reenvío del mismo considerando el máximo aprovechamiento de los enlaces y ancho de banda disponible, razón por la cual las estrategias de ingeniería de tráfico toman mayor relevancia dentro de las redes actuales en los centros de datos (Göransson & Black, 2014).

- Tunelización: si bien la existencia de tecnología de tunelización en centros de datos no es algo reciente, si se destaca que con la aparición de la virtualización de las redes las necesidades de tecnologías de tunelización han ido cambiando, y si bien no son una obligatoriedad, con la finalidad de manejar de mejor forma los grandes volúmenes de tráfico de los centros de datos, se puede destacar la implementación de tres tecnologías que son: Virtual eXtensible Local Area Network (VXLAN), Generic Routing Encapsulation (NVGRE), y Stateless Transport Tunneling (STT) (Göransson & Black, 2014).

2.2.2 Arquitectura

Dentro de un data center su arquitectura de red se organiza generalmente en diferentes capas, pues lo más usual es que se empleen equipos de distintos dispositivos con la finalidad de aportar un mejor rendimiento en la red y aportar características como la reconfigurabilidad dinámica de la red y la capacidad de manejar grandes volúmenes de tráfico, razón por la cual existen muchos tipos de arquitecturas que se pueden considerar, entre ellas se encuentran: topología basada en árbol, topología basada en la recursividad, red híbrida, y red directa, estas se diferencian entre sí en la forma de organizar e interconectar dispositivos de red. Sin embargo, las data centers pueden dividir a sus topologías de red en otras dos

categorías, las cuales son: centradas en conmutadores y centradas en servidores, estas topologías se diferencian en que, la primera centra la conciencia de interconexión en los routers y conmutadores de la red, mientras que la segunda se enfoca en admitir que los servidores también gestionen la lógica de reenvío de paquetes (Ting Wang et al., 2014). Para una SDN esta segunda categoría cobra mayor relevancia y debe realizar un estudio propio para determinar la organización más optima del centro de datos en torno al tipo de reenvío que se vaya a manejar.

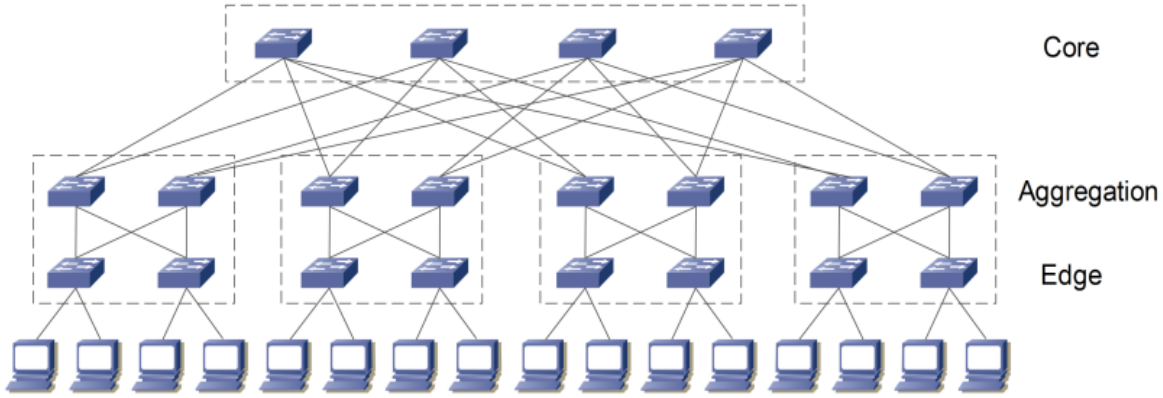
Una aplicación de una nueva estructura de red dentro de un data center debe considerar la existencia de las tres capas de la arquitectura de una red definida por software mencionada en la **Figura 2.** Arquitectura de una red definida por software, por lo que se debe organizar a los equipos compatibles con la tecnología de red definida por software en el plano de datos o de reenvío, esto dentro de una red basada en conmutadores, en la que se considera el uso de múltiples dispositivos de red para el levantamiento de la red, en la que se coloca a los servidores como hosts de estos dispositivos. Una topología de centro de datos bastante empleada la cual se encuentra basada en el concepto de topología de árbol es la topología fat-tree[5] misma que puede evidenciarse en la **Figura 3.** Topología Fat-tree simple para un data center a modo de ejemplo, este tipo de topología considera una organización en tres capas diferentes para distribuir a los equipos de conmutación de paquetes, para esta topología la aplicación de la arquitectura de red definida por software se realizaría mediante le implementación del plano de control de forma distribuida entre los equipos de la capa de core, y la

[5] Fat-Tree: topología de árbol constituida por una gran cantidad de enlaces entre dispositivos de red para brindar grandes capacidades de ancho de banda y baja latencia (Ting Wang et al., 2014).

capa de agregación como una alternativa para usar controladores descentralizados, pues por la organización de los conmutadores es más conveniente contar con varios dispositivos o servidores encargados del control del flujo del tráfico para mantener la capacidad actual del data center.

Figura 3.

Topología Fat-tree simple para un data center



Fuente: Obtenido de (Ting Wang et al., 2014)

2.2.3 Recomendaciones de implementación

La implementación de un data center basado en SDN no representa un inconveniente ya que se puede planificar la red con esta organización desde sus inicios y la adquisición de los equipos se hará centrada en la utilización de cierto controlador y ciertos protocolos según como se haya designado la construcción de la red, sin embargo, debido a que realmente en la mayoría de los casos ya se cuenta con una red estructurada y funcional el cambio de arquitectura no es tan sencillo, razón por la cual se debe considerar un plan de migración que permita realizar el cambio de forma paulatina sin afectar a la red.

El proceso de migración debe considerar algunos factores como los costos, afectaciones de rendimiento, disponibilidad de los servicios y los procesos de gestión y seguridad que requieren mayor atención al pasar de una tecnología a

otra, pues en este proceso se pueden descubrir vulnerabilidades de seguridad, así como falencias a las que se debe dar solución a lo largo de este proceso(Open Netowrking Foundation, 2014).

La migración de una red tradicional a SDN puede incurrir en diferentes etapas que pueden adaptarse a la organización o empresa que requiera de este proceso en sus centros de datos, pues en este punto se requiere que un administrador de red gestione y analice las condiciones actuales de la topología para tomar las mejores decisiones en este proceso, según (Open Netowrking Foundation, 2014) existen ciertos pasos clave a considerar en una migración, estos pasos son:

- Identificar y priorizar recursos básicos de la nueva red: si bien las redes definidas por software son una alternativa de solución a varios de los problemas que enfrentan los centros de datos en la actualidad, puede que, en un principio estas no se acoplen a toda la funcionalidad de la red actual, razón por la cual es importante reconocer los requerimientos prioritarios de la red actual para que estos sean los primeros en implementarse en la nueva red para de esta forma no perder la funcionalidad del data center hasta culminar el proceso de migración.
- Preparación de la red actual para la migración: al considerar el paso a una nueva tecnología se debe realizar un proceso de preparación en el que se pueda mover la red de partida a un estado intermedio, limpio y organizado que permita seguir con el proceso de migración de la red.
- Estructuración de una migración por fases: es importante que se considere que cada dispositivo individual requiere de un proceso de cambio con un método específico, por lo que pensar en un plan de

migración por etapas que permita agrupar los procesos de ciertos dispositivos es la mejor opción.

- Validar resultados: al realizar un proceso de migración se pueden tener varias expectativas sobre el proceso, mismas que deben ser comprobadas en el resultado final de la red, para de esta forma determinar si la migración se puede clasificar como totalmente concluida al obtener así una red que cumple con cada una de las demandas establecidas originalmente como objetivos de la migración a SDN.

En el esquema de aplicación de redes definidas por software dentro de un centro de datos, los diferentes fabricantes han planteado modelos que permitan gestionar este proceso. Cisco, por ejemplo, plantea un modelo basado en tres pilares esenciales, mismos que se basan en:

- Infraestructura basada en aplicaciones (ACI): consiste en una plataforma centralizada para la gestión de las redes. Se debe separar el plano de datos para lograr redes más eficientes, escalables y seguras. Esta plataforma incluye un controlador de infraestructura de políticas de aplicaciones (APIC) que brinda el soporte necesario para las arquitecturas de SDN, así como llevar un enfoque centralizado en aplicaciones para la administración de políticas de red (Huitema, 2015).
- Tejido Programable: este pilar hace referencia a la búsqueda de escalabilidad y simplicidad en las redes para lo cual se busca una convivencia con nuevas tecnologías que aportan mejoras en tecnologías tradiciones, como es el caso de la ampliación de VXLAN (Huitema, 2015).

- Red Programable: la programabilidad de la infraestructura de red es crucial en una red definida por software, pues esta impulsa la automatización que a su vez se dirige hacia la búsqueda de mayor velocidad y eficiencia en la red (Huitema, 2015).

Si bien la elección del software a emplear en una SDN depende de múltiples factores el realizar un análisis de lo que ofertan los distintos fabricantes y como estos recomiendan la implementación y despliegue de esta nueva arquitectura es de las mejores prácticas que pueden existir.

CAPÍTULO III

Evaluación y diagnóstico

3.1 Situación actual

La Dirección de Desarrollo Tecnológico e Informático (DDTI) de la Universidad Técnica del Norte es el departamento encargado de la administración y gestión de redes y servicios dentro de la institución. Hoy, la UTN es parte de la red de CEDIA (Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia), que le da acceso a recursos y servicios de conexión nacionales. Además, mantiene negociaciones directas con su proveedor, TELCONET, quien contribuye al mantenimiento y operación de la infraestructura de red de la UTN.

Al momento, la infraestructura de red existente se basa en una arquitectura de networking tradicional, principalmente compuesta por equipos del fabricante CISCO tanto en el núcleo de la red (core) como en los segmentos de distribución. En la parte lógica de la red, se hace uso de VLANs para segmentar y gestionar de forma eficiente el tráfico de datos.

Un detalle que considerar en cuanto a los servicios que gestiona el DDTI es que, se ha considerado la migración de muchos de estos a entornos virtualizados para mejorar la prestación de servicios a toda la comunidad universitaria, este proceso ha ocasionado que la arquitectura de red dentro del Datacenter se modifique parcialmente teniendo como objetivo concretar el proceso de migración de ciertos servicios, para que de este modo la estructura se acople de mejor manera y se puedan usar plataformas más modernas y eficientes que mejoren el rendimiento y la disponibilidad de estos.

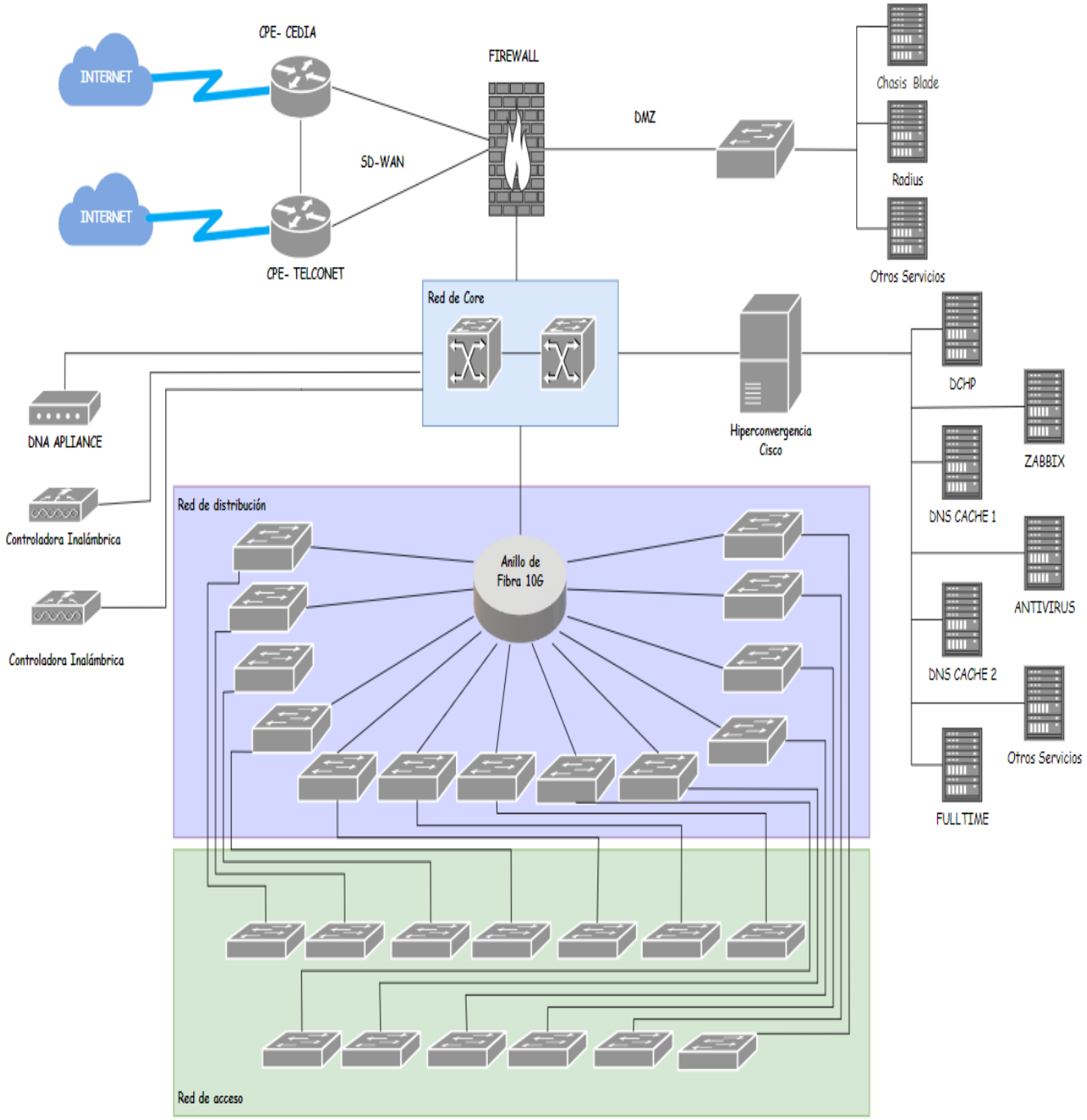
3.2 Topología de red

La arquitectura de red, o topología, es un aspecto fundamental en el análisis del funcionamiento de una red de alta demanda, como la que opera dentro de un datacenter. En este contexto, la organización de los equipos y servidores se realiza en función de las necesidades específicas de la institución. La distribución de estos componentes se basa en la disposición general representada en la **Figura 4.** Topología de red de la Universidad Técnica del Norte, la cual ofrece una visión general de la configuración actual de la infraestructura.

La **Figura 4** representa la disposición física de los equipos de red dentro del Datacenter de la Universidad Técnica del Norte. Esta disposición se ha diseñado para garantizar una conectividad eficiente y confiable entre los diferentes dispositivos. Además, se han considerado aspectos como la redundancia y la escalabilidad para asegurar la disponibilidad y el rendimiento óptimo de la red.

Figura 4.

Topología de red de la Universidad Técnica del Norte



Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

3.2.1 Distribución física de la red

La distribución e interconexión entre los equipos mostrados en la **Figura 4** se puede explicar mediante segmentos de red que permitan comprender de

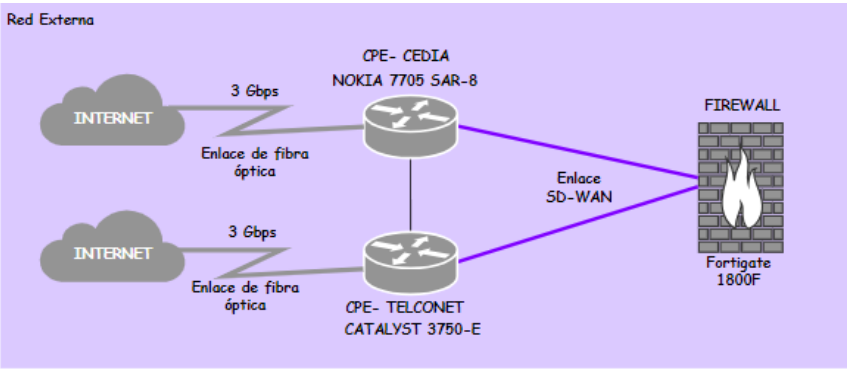
mejor manera el papel que cumple cada uno de estos dentro de la red establecida en el datacenter del DDTI de la UTN, por lo que se considera una subdivisión para cada segmento de la topología planteada anteriormente.

Es importante considerar que la red inicia desde un punto externo que no puede ser controlado ni administrado directamente por el personal del DDTI. La **Figura 5** muestra la extranet, que comprende el acceso a internet a través de un enlace de fibra óptica de 3 Gbps con dos equipos de alto rendimiento proporcionados por TELCONET y CEDIA. Estos equipos, un router Nokia 7705 SAR-8 y un Cisco Catalyst 3750-E, forman un enlace SD-WAN para la gestión del tráfico desde el primer equipo administrable de la UTN hacia la red del proveedor.

El primer equipo al que el DDTI tiene acceso y control total es un firewall, específicamente un Fortigate 1800F, encargado de administrar políticas de seguridad y acceso a la red en general para administrar el tráfico de la red. Las interfaces conectadas al enlace SD-WAN representan las últimas instancias de la red externa.

Figura 5.

Red externa de conexión con el proveedor



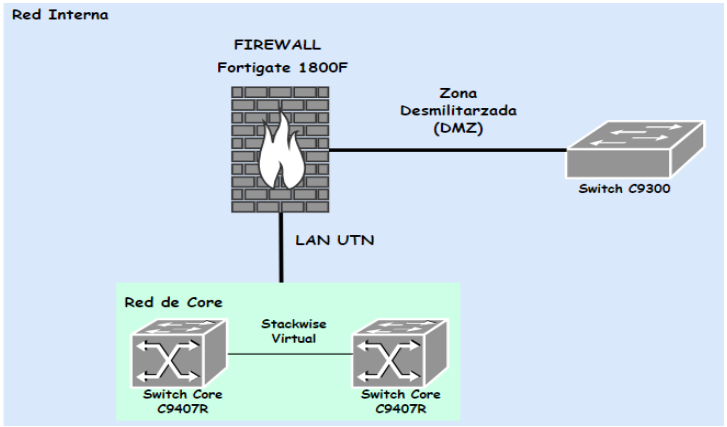
Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

La red interna del datacenter, puede ser considerada a partir del firewall, mismo que es el encargado de controlar el tráfico entrante y saliente de la red, , en este punto se consideran dos de sus interfaces: la primera se dirige hacia una zona desmilitarizada, misma que cuenta con políticas diferentes y menos restringidas para permitir que usuarios externos puedan tener acceso a los servicios ubicados en este segmento de la red. Por otro lado, se encuentra la LAN de la UTN, misma que se enlaza con el Core de la red gestionada por el DDTI.

A partir de lo mostrado en la **Figura 6**, se puede determinar cómo se da esta interconexión, tomando atención especial en el bloque del core de la red. Este bloque se encuentra formado por dos equipos Cisco C9407R que se comunican mediante Stackwise Virtual, una tecnología de Cisco que se emplea para virtualizar varios switches como una sola entidad lógica para su administración, permitiendo mayor flexibilidad, escalabilidad y alta disponibilidad en redes altamente concurridas como un datacenter. El core de la red es el encargado de la dirección del tráfico en la intranet de la institución.

Figura 6.

Red interna del datacenter de la UTN



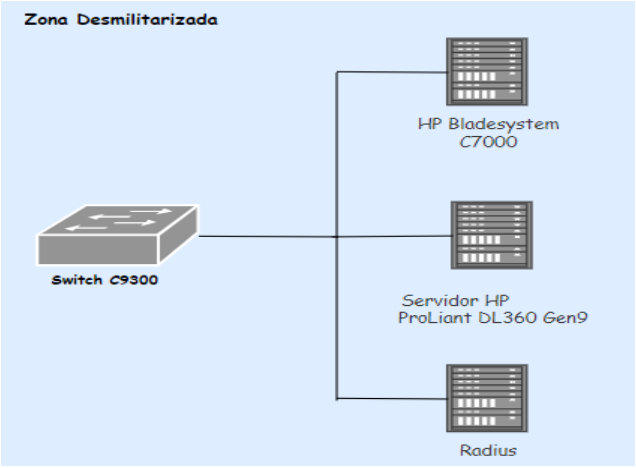
Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

El primer segmento de red representado en la **Figura 7**, la zona desmilitarizada (DMZ). Esta zona se compone por un switch Cisco C9300, que se conecta a un grupo de servidores HP ProLiant DL360, y a un Chasisblade HP Bladesystem C7000, destinados a proporcionar diferentes servicios a usuarios internos y externos. Sin embargo, se debe considerar que estos equipos ya no se encuentran en producción, pues hace tiempo atrás se realizaron procesos de migración y virtualización para ciertos servicios. Un componente activo de esta DMZ es el servidor Radius, mismo que se emplea para el acceso y autenticación a la red inalámbrica EDUROAM de la universidad.

Si bien esta DMZ aún puede ser empleada, pues la conexión y la dirección del tráfico aún se consideran dentro del firewall, se ha optado por nuevas soluciones que permiten mejorar la respuesta de los servidores en una red altamente concurrida. Por lo tanto, este factor es importante al momento de pensar en la migración de este segmento de red, puesto que ciertos equipos localizados se encuentran desactualizados para la incorporación de nuevas tecnologías.

Figura 7.

Zona desmilitarizada del datacenter



Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

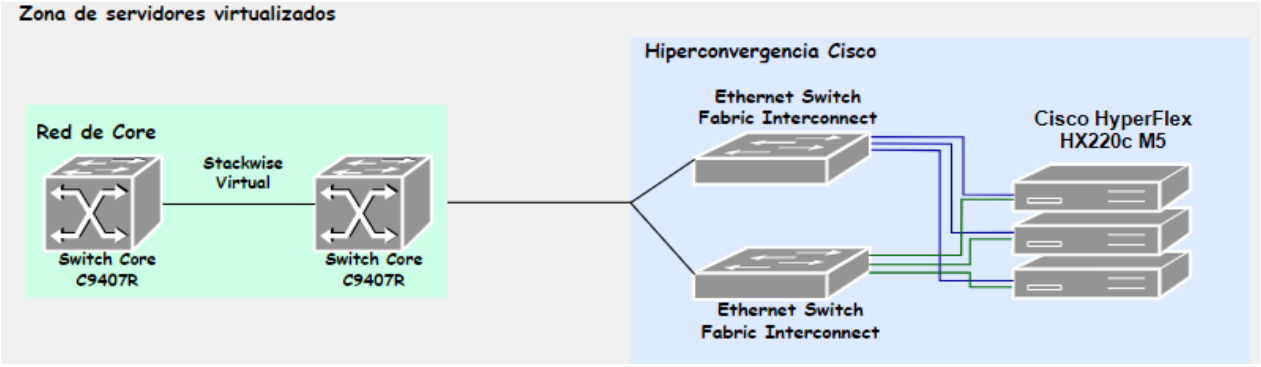
El segmento de la LAN de la UTN, como se mencionó anteriormente, se compone del núcleo o Core de la red formado por dos switches C9407R enlazados mediante una tecnología propietaria. Este core de red se conecta de forma directa a la llamada Hiperconvergencia, una tecnología que hace referencia a una solución de integración de infraestructura física para combinar características, como la capacidad de cómputo, el almacenamiento y gestión de redes en una plataforma integrada.

La Hiperconvergencia de Cisco, implementada en el datacenter de la UTN y representada en la **Figura 8**, actualmente consta de tres nodos Cisco HyperFlex HX220c M5. Estos nodos incorporan servidores con todos sus recursos a través del software de HyperFlex, lo que incluye el control y la gestión de los equipos mediante la formación de un clúster entre todos los nodos involucrados. La conexión se realiza mediante el llamado Fabric Interconnect, enlaces que se establecen con switches tradicionales para permitir la interconexión de red entre los equipos de HyperFlex y el Core de la red.

El objetivo de usar esta tecnología es mejorar el rendimiento y los tiempos de respuesta en los servidores del datacenter, pues con la integración de tres nodos se consigue un almacenamiento de 18 TB, y al usar dos Fabric Interconnect se ofrece mayor escalabilidad y resiliencia para los servidores alojados en el cloud de HyperFlex, este se basa en la virtualización de los servidores empleando VMWare y múltiples máquinas virtuales, reemplazando a los servidores tradicionales que se alojaban anteriormente en la DMZ.

Figura 8.

Hiperconvergencia Cisco para servidores en el datacenter



Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

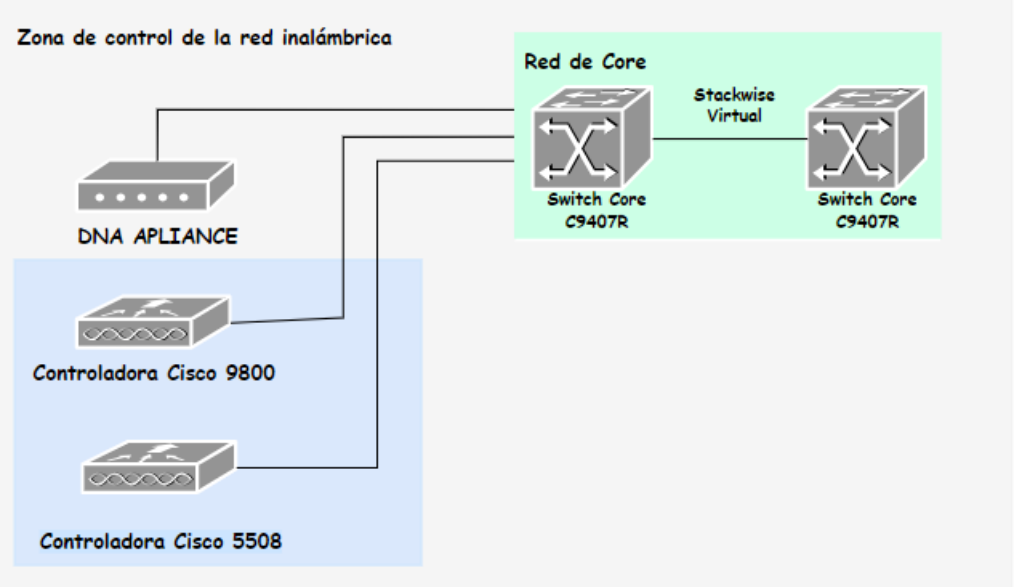
Por otro lado, dentro del datacenter se cuenta también con la conexión de las controladoras de la red inalámbrica, mismas que a su vez se conectan a los puntos de acceso inalámbricos de todo el campus. Como se muestra en la Figura 9, estas son de dos tipos, una Cisco 9800 empleada para los APs más nuevos, y la controladora Cisco 5508 que se usa para modelos pasados, estas se conectan directamente al core de la red para que este redireccione el tráfico según corresponda.

Además se encuentra también el DNA Appliance , un componente empleado principalmente para el monitoreo de los dispositivos que se encuentran asociados al mismo, guardando un registro de lo que sucede en cada uno de estos para que de este modo el personal de DDTI pueda detectar fallas que afecten al rendimiento de la red, sin embargo, esta plataforma de monitoreo no es la única que se emplea, ya que dentro de la Hiperconvergencia mostrada en la Figura 8, se

ha incluido un servidor de monitoreo que envía alertas de los eventos importantes en toda la red.

Figura 9.

Control de la red inalámbrica de la UTN en el datacenter



Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

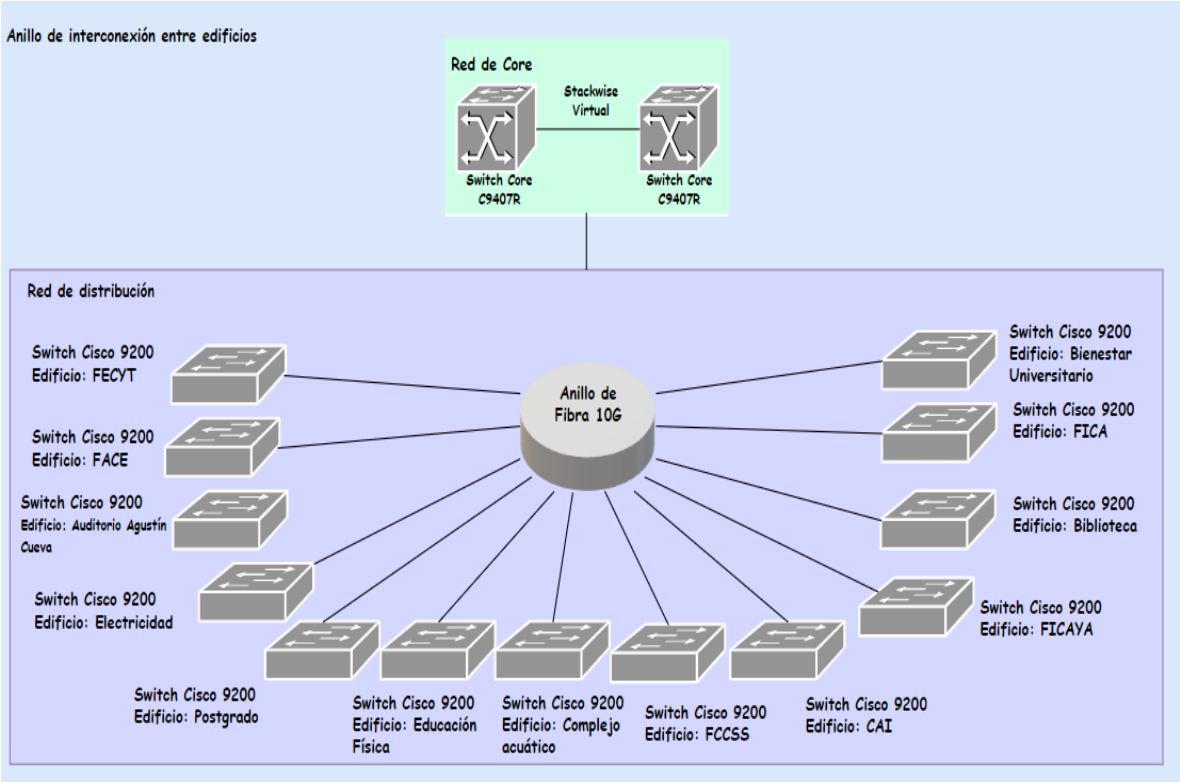
A continuación, la red de distribución planteada se conecta de forma directa al core de la red también, esto se logra mediante una construcción de una red en anillo para todos los edificios del campus principal, estos equipos son varios Cisco Catalyst en diferentes modelos como C9200, C3850 y C2960 , y se encuentran en cada uno de los edificios como se indica en la *Figura 10.*

Para asegurar una conectividad robusta, se implementa un anillo de fibra óptica con una capacidad de 10 Gbps, capaz de manejar las elevadas demandas de tráfico en todo el campus. Este anillo cuenta con redundancia, con enlaces en ambos sentidos, para garantizar la continuidad del servicio ante posibles fallas.

El tráfico generado en las redes de acceso que se conectan a cada uno de estos equipos pasa al core de la red para dirigirse al destino correspondiente en base a las políticas establecidas.

Figura 10.

Conexión de las redes de cada edificio de la UTN hacia el datacenter



Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

Finalmente, la red de acceso de cada uno de los edificios de la UTN se anexa a la red de distribución mostrada en la **Figura 10**, esta red no es idéntica en todas las facultades debido a que se acopla a las necesidades y utilización de dispositivos de las mismas, por lo que la distribución y los equipos empleados varían considerada la red de acceso mostrada en la **Figura 4** como una representación breve y generalizada, este escenario se encuentra fuera del cuarto de equipos del datacenter del DDTI.

3.2.2 Distribución lógica de la red

La distribución lógica de la red dentro del datacenter de la UTN se fundamenta en la asignación de VLANs (Redes de Área Local Virtuales), las cuales desempeñan un papel crucial en la gestión de redes de alta demanda. Estas VLANs permiten la división lógica de una red física, mejorando el rendimiento, la seguridad y la administración de los recursos. El uso de VLANS permite cumplir con necesidades particulares de los diferentes departamentos que forman parte de la institución. El DDTI se encarga actualmente de la gestión de 45 VLANs, mismas que permiten aislar determinados segmentos de red para incrementar la seguridad, y disminuir el impacto de eventos como fallos de dispositivos o ataques maliciosos que podrían afectar a la fiabilidad y disponibilidad de la red del campus.

En la **Tabla 3** se detallan las VLANs gestionadas por el DDTI, donde se especifica el número asignado a cada VLAN, así como una descripción que indica el departamento o área a la que pertenece el tráfico de esa VLAN, se considera también la información de la dirección IPv4 empleada, así como su máscara de red correspondiente y el gateway o puerta de enlace predeterminada para cada una de las VLANs. Esta información proporciona una visión clara del tráfico que circula por la red del campus y permite una administración más eficiente de los recursos, adaptándose a las demandas de los usuarios de la institución.

Tabla 3.

Distribución de VLANs en IPv4

N°	DESCRIPCIÓN	VLAN	DIRECCIÓN IP	MASCARA DE SUBRED	GATEWAY
1	EQUIPOS-ACTIVOS	1	172.16.1.0	255.255.255.0	172.16.1.1
2	DMZ	2	10.24.8.0	255.255.255.0	10.24.8.1
3	CORE-FIREWALL	3	172.16.3.0	255.255.255.0	172.16.3.1
4	EQUIPOS-WIRELESS	4	172.16.4.0	255.255.255.0	172.16.4.1
5	CCTV	6	172.16.6.0	255.255.255.0	172.16.6.1
6	RELOJES-BIOMETRICOS	7	172.16.7.0	255.255.255.0	172.16.7.1
7	TELEFONIA-IP-ELASTIX	8	172.16.8.0	255.255.252.0	172.16.8.1
8	AUTORIDADES	12	172.16.12.0	255.255.255.0	172.16.12.1
9	DDTI	14	172.16.14.0	255.255.255.0	172.16.14.1
10	FINANCIERO	16	172.16.16.0	255.255.255.0	172.16.16.1
11	COMUNICACION-ORGANIZACIONAL	18	172.16.18.0	255.255.255.0	172.16.18.1
12	ADMINISTRATIVOS	20	172.16.20.0	255.255.255.0	172.16.20.1
13	ADQUISICIONES	22	172.16.22.0	255.255.255.0	172.16.22.1
14	U-EMPRENDE	24	172.16.24.0	255.255.254.0	172.16.24.1
15	AGUSTIN-CUEVA	26	172.16.26.0	255.255.255.0	172.16.26.1

16	BIENESTAR-DOCENTES	28	172.16.28.0	255.255.255.0	172.16.28.1
17	BIENESTAR-ADMINISTRATIVOS	30	172.16.30.0	255.255.255.0	172.16.30.1
18	CLUBES-UTN	32	172.16.32.0	255.255.255.0	172.16.32.1
19	NATIVA	39	----------	----------	----------
20	FICA-LABORATORIOS	40	172.17.40.0	255.255.254.0	172.17.40.1
21	FICA-WIRELESS	42	172.17.42.0	255.255.255.0	172.17.42.1
22	FICA-ADMINISTRATIVOS	44	172.16.44.0	255.255.255.0	172.16.44.1
23	FICAYA-LABORATORIOS	48	172.17.48.0	255.255.254.0	172.17.48.1
24	FICAYA-ADMINISTRATIVOS	52	172.16.52.0	255.255.255.0	172.16.52.1
25	FECYT-LABORATORIOS	56	172.17.56.0	255.255.254.0	172.17.56.1
26	FECYT-ADMINISTRATIVOS	60	172.16.60.0	255.255.255.0	172.16.60.1
27	FACAE-LABORATORIOS	64	172.17.64.0	255.255.254.0	172.17.64.1
28	FACAE-ADMINISTRATIVOS	68	172.16.68.0	255.255.255.0	172.16.68.1
29	FCCSS-LABORATORIOS	72	172.17.72.0	255.255.254.0	172.17.72.1

30	FCCSS-ADMINISTRATIVOS	76	172.16.76.0	255.255.255.0	172.16.76.1
31	POSGRADO-LABORATORIOS	80	172.17.80.0	255.255.254.0	172.17.80.1
32	POSGRADO-ADMINISTRATIVOS	84	172.16.84.0	255.255.255.0	172.16.84.1
33	CAI-LABORATORIOS	88	172.17.88.0	255.255.254.0	172.17.88.1
34	CAI-ADMINISTRATIVOS	92	172.16.92.0	255.255.255.0	172.16.92.1
35	BIBLIOTECA-LABORATORIOS	96	172.17.96.0	255.255.254.0	172.17.96.1
36	BIBLIOTECA-DOCENTES	98	172.16.98.0	255.255.255.0	172.16.98.1
37	BIBLIOTECA-ADMINISTRATIVOS	100	172.16.100.0	255.255.255.0	172.16.100.1
38	EDUROAM	128	172.20.128.0	255.255.224.0	172.20.128.1
39	WIRELESS-EVENTOS	160	172.21.160.0	255.255.248.0	172.21.160.1
40	SERVICE-DNA	192	172.23.192.0	255.255.224.0	172.23.192.1
41	COPIADORA	201	172.24.201.0	255.255.255.0	172.24.201.1
42	HX-INBAND-MGMT	202	172.25.202.0	255.255.255.0	172.25.202.1
43	HX-STORAGE-DATA	203	172.25.203.0	255.255.255.0	172.25.203.1
44	HX-VMOTION	204	172.25.204.0	255.255.255.0	172.25.204.1

45	HX-VM-NETWORK	205	172.25.205.0	255.255.255.0	172.25.205.1

Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

Considerando el despliegue y el crecimiento de las redes, el protocolo IPv4 se ha vuelto insuficiente con el tiempo. Por tanto, años atrás, el DDTI realizó el proceso de transición a IPv6. En la actualidad, dentro del campus, se implementa la solución de transición dual stack[6], la cual permite manejar ambos protocolos de manera simultánea en todos los equipos de red de la institución. En este contexto, se considera que la distribución de VLANs también se debe realizar para IPv6, la **Tabla 4** presentada a continuación resume la distribución realizada para esta versión del protocolo IP.

Tabla 4.

Distribución de VLANs en IPv6

N°	DESCRIPCIÓN	VLAN	DIRECCIÓN IP	PREFIJO	GATEWAY
1	EQUIPOS-ACTIVOS	1	2801:10:6800:1::	/64	2801:10:6800:1::1
2	DMZ	2	2801:10:6800:1024::	/64	2801:10:6800:1024::1
3	CORE-FIREWALL	3	2801:10:6800:3::	/64	2801:10:6800:3::1
4	EQUIPOS-WIRELESS	4	2801:10:6800:4::	/64	2801:10:6800:4::1
5	CCTV	6	2801:10:6800:6::	/64	2801:10:6800:6::1
6	RELOJES-BIOMETRICOS	7	2801:10:6800:7::	/64	2801:10:6800:7::1

[6] Dual Stack: solución de migración de IPv4 a IPv6 llamada comúnmente como red de doble pila, es una red en la que todos los nodos admiten ambos protocolos permitiendo una transición fluida al permitir que el sistema elija la versión de IP a emplear para el reenvío de tráfico (Rouse, 2012).

7	TELEFONIA-IP-ELASTIX	8	2801:10:6800:8::	/64	2801:10:6800:8::1
8	AUTORIDADES	12	2801:10:6800:12::	/64	2801:10:6800:12::1
9	DDTI	14	2801:10:6800:14::	/64	2801:10:6800:14::1
10	FINANCIERO	16	2801:10:6800:16::	/64	2801:10:6800:16::1
11	COMUNICACION-ORGANIZACIONAL	18	2801:10:6800:18::	/64	2801:10:6800:18::1
12	ADMINISTRATIVOS	20	2801:10:6800:20::	/64	2801:10:6800:20::1
13	ADQUISICIONES	22	2801:10:6800:22::	/64	2801:10:6800:22::1
14	U-EMPRENDE	24	2801:10:6800:24::	/64	2801:10:6800:24::1
15	AGUSTIN-CUEVA	26	2801:10:6800:26::	/64	2801:10:6800:26::1
16	BIENESTAR-DOCENTES	28	2801:10:6800:28::	/64	2801:10:6800:28::1
17	BIENESTAR-ADMINISTRATIVOS	30	2801:10:6800:30::	/64	2801:10:6800:30::1
18	CLUBES-UTN	32	2801:10:6800:32::	/64	2801:10:6800:32::1
19	NATIVA	39	-----	----	-----
20	FICA-LABORATORIOS	40	2801:10:6800:40::	/64	2801:10:6800:40::1
21	FICA-WIRELESS	42	2801:10:6800:42::	/64	2801:10:6800:42::1
22	FICA-ADMINISTRATIVOS	44	2801:10:6800:44::	/64	2801:10:6800:44::1
23	FICAYA-LABORATORIOS	48	2801:10:6800:48::	/64	2801:10:6800:48::1

24	FICAYA-ADMINISTRATIVOS	52	2801:10:6800:52::	/64	2801:10:6800:52::1
25	FECYT-LABORATORIOS	56	2801:10:6800:56::	/64	2801:10:6800:56::1
26	FECYT-ADMINISTRATIVOS	60	2801:10:6800:60::	/64	2801:10:6800:60::1
27	FACAE-LABORATORIOS	64	2801:10:6800:64::	/64	2801:10:6800:64::1
28	FACAE-ADMINISTRATIVOS	68	2801:10:6800:68::	/64	2801:10:6800:68::1
29	FCCSS-LABORATORIOS	72	2801:10:6800:72::	/64	2801:10:6800:72::1
30	FCCSS-ADMINISTRATIVOS	76	2801:10:6800:76::	/64	2801:10:6800:76::1
31	POSTGRADO-LABORATORIOS	80	2801:10:6800:80::	/64	2801:10:6800:80::1
32	POSTGRADO-ADMINISTRATIVOS	84	2801:10:6800:84::	/64	2801:10:6800:84::1
33	CAI-LABORATORIOS	88	2801:10:6800:88::	/64	2801:10:6800:88::1
34	CAI-ADMINISTRATIVOS	92	2801:10:6800:92::	/64	2801:10:6800:92::1
35	BIBLIOTECA-LABORATORIOS	96	2801:10:6800:96::	/64	2801:10:6800:96::1

36	BIBLIOTECA-DOCENTES	98	2801:10:6800:98::	/64	2801:10:6800:98::1
37	BIBLIOTECA-ADMINISTRATIVOS	100	2801:10:6800:100::	/64	2801:10:6800:100::1
38	EDUROAM	128	2801:10:6800:128::	/64	2801:10:6800:128::1
39	WIRELESS-EVENTOS	160	2801:10:6800:160::	/64	2801:10:6800:160::1
40	SERVICE-DNA	192	2801:10:6800:192::	/64	2801:10:6800:192::1
41	COPIADORA	201	2801:10:6800:201::	/64	2801:10:6800:201::1
42	HX-INBAND-MGMT	202	2801:10:6800:202::	/64	2801:10:6800:202::1
43	HX-STORAGE-DATA	203	2801:10:6800:203::	/64	2801:10:6800:203::1
44	HX-VMOTION	204	2801:10:6800:204::	/64	2801:10:6800:204::1
45	HX-VM-NETWORK	205	2801:10:6800:205::	/64	2801:10:6800:205::1

Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

3.3 Descripción de los elementos de red del datacenter

Dentro del cuarto de equipos de DDTI hay varios equipos de la red de la Universidad Técnica del Norte, estos incluyen equipos administrables para la institución, como los que controla directamente el proveedor de servicios de telecomunicaciones, permitiendo que la red interna de la UTN pueda comunicarse con la extranet, varios equipos se mencionan brevemente en los diagramas presentados en un apartado anterior, por lo que en este apartado se abordará de manera detallada a cada uno de ellos.

3.3.1 Router Nokia 7705 SAR-8

El router Nokia 7705 SAR-8 mostrado en la **Figura 10**, es un equipo de alto rendimiento que ofrece una amplia gama de interfaces para diferentes servicios como: voz, datos, TDM, teleprotección, entre otras. Admite protocolos de enrutamiento aplicados a servicios remotos permitiendo formar redes robustas, lo cual lo vuelve un equipo ideal para trabajar en entornos de alta demanda. Dentro de la topología de red mostrada en la **Figura 11**, este router corresponde a los equipos gestionados por el proveedor de servicios.

Figura 11.

Router Nokia 7705 SAR-8

Fuente: Obtenido de (Nokia, 2023)

Para definir de mejor manera este equipo se consideran las características presentadas en la **Tabla 5**, mismas que describen aspectos tanto de software como hardware del router antes mencionado.

Tabla 5.

Características del router Nokia 7705 SAR-8

Característica	Descripción
Procesador	Cuenta con un procesador integrado Cavium OCTEON II CN6335 de seis núcleos. Soporta frecuencias de hasta 800 MHz.
Memoria	- Memoria RAM: memoria volátil empleada para almacenar datos y programas en ejecución, los valores típicos se encuentran entre 4 GB y 16 GB.

	- Memoria Flash: memoria no volátil empleada para el almacenamiento del sistema operativo y otros archivos de importancia para el dispositivo, comúnmente se establece en valores entre 8 GB y 32 GB. - Memoria externa: incluye tarjetas SD de tipo extraíbles que pueden ser colocadas en las ranuras añadidas al dispositivo para incrementar el almacenamiento hasta en 64 GB.
Sistema operativo	Los dispositivos de Nokia comparten un sistema operativo para todos sus SAR (Service Agreggation Router), el cual se denomina SAR OS (Service Agreggation Router OS).
Ranuras	- 2 ranuras de 10 Gbps - 4 ranuras de 2,5 Gbps
Interfaces de control	- Consola - Gestión
Interfaces	Soporta interfaces enfocadas a distintos protocolos como es el caso de: Ethernet, POS (Packets over SONET/SDH), ATM, ATM-IMA (Inverse Multipexing for ATM), Frame Relay, HDLC, PPP (Point to Point Protocol), MCPPP (Muti-class PPP), MLPPP (Muti- Link PPP), TDM.
Protocolos de red compatibles	Tiene compatibilidad con distintos protocolos de red como: enrutamiento por segmentos (SR) con protocolos como ISIS (Intermediate System-to Intermediate System) y OSPF (Open Shortest Path First), ingeniería de tráfico, MPLS, RIP, BGP, IPv6 sobre túneles IPv4, NG-MVPN (Next-Generation multicast VPNs) sobre MPLS.
Seguridad	Permite configurar determinadas medidas y políticas de seguridad basadas en Secure Shell (SSH) con la integración de HMAC y algoritmos de hash seguros como SHA2, así como el uso de Diffie-Helman para el intercambio de contraseñas. Se integran también funciones de tunelización con Dot1x, cifrado de seguridad IP (IPsec) sobre MPLS, seguridad para capa transporte mediante TLS1.2 y TLS 1.3, traducción de direcciones de red (NAT) y firewall de estado con soporte multicanal.
Calidad de servicio	Admite el uso de calidad de servicio jerárquica (H-QoS), clasificación inteligente de paquetes, vigilancia y programación.

Fuente: Adaptado de (Nokia, 2023) , (Nokia, n.d.) y (Alcatel-Lucent, 2009)

3.3.2 Switch Cisco C3750-E

El Switch Cisco C3750-E es un equipo de alto rendimiento empleado en soluciones de interconexión a nivel empresarial, como se aprecia en la **Figura 12**, es un equipo que cuenta con un número alto de puertos siendo ideal para redes de distribución. Es un switch multicapa, por lo que puede trabajar con conmutación de tramas convencional en la capa 2 del modelo de referencia OSI, o puede emplear protocolos de enrutamiento IP de cada 3 del mismo modelo referencial, los protocolos de enrutamiento compatibles con este equipo son: RIP v1, RIP v2, EIGRP y enrutamiento estático.

Figura 12.

Switch Cisco C3750-E

Fuente: Obtenido de (OceanTech, 2023)

Tomando como referencia lo mencionado con anterioridad, se han considerado las características y especificaciones más relevantes del Switch Cisco C3750-E, y se han resumido en la **Tabla 6** presentada a continuación.

Tabla 6.

Características del switch Cisco C3750-E

Característica	Descripción
Procesador	Como parte de la familia Catalyst 3750 cuenta con un procesador de un solo núcleo PowePc 405, con una frecuencia de 266 MHz.
Memoria	- Memoria RAM: cuenta con una memoria RAM de 256 MB.

	- Memoria Flash: memoria no volátil integrada en el dispositivo de 64 MB para almacenamiento del sistema operativo y archivos de configuración.
Sistema operativo	Trabaja con el sistema operativo integrado de la familia Catalyst 3750 que es Cisco IOS (Internetwork Operating System) y puede añadir softwares como: Standard Multilayer software Image (SMI) y Cisco CMS Software.
Puertos	48 puertos
Tamaño de tabla de direcciones MAC	12000 entradas en la tabla MAC.
Interfaces	- 48 interfaces x10Base-T/100Base-TX/1000Base-T RJ-45 PoE - 1interfaz de consola RJ-45 - 1 interfaz 10Base-t/100Base-TX administrativa RJ-45
Tiempo medio entre fallos (MTBF)	177975 horas
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.3, IEEE 802.3u, IEEE 802.3z, IEEE 802.1D, IEEE 802.1Q, IEEE 802.3ab, IEEE 802.1p, IEEE 802.3af, IEEE 802.3x, IEEE 802.3ad (LACP), IEEE 802.1w, IEEE 802.1x, IEEE 802.1s
Especificaciones	Equipo de alto rendimiento que opera en capa 3 del modelo de referencia OSI, por lo que cuenta con capacidad de ruteo IP, creación de listas de control de acceso y seguridad aplicable a puertos estáticos. Opera con tecnología de tipo alámbrica, admitiendo anchos de banda de 64 Gbps, tasa de transferencia máxima de 1 Gbps y velocidades de transferencia de paquetes de 13.1 Mpps[7], y es de tipo apilable.
Seguridad	Cuenta con diferentes formas de autenticación como: Kerberos, Secure Shell (SSH), RADIUS y TACACS+.
Protocolos de gestión	Existen varios protocolos compatibles con el acceso a la gestión de este equipo independientemente del grado de seguridad de estos: SNMP 1, RMON 1, RMON 2, RMON 3, RMON 9, Telnet, SNMP 3, SNMP 2c, SSH
Plataformas de gestión	Integra diferentes plataformas de gestión, mismas que se vinculan o reservan a licencias o elementos

[7] Mpps: Millones de Paquetes por Segundo o Mpps es empleado para la medición de rendimiento de un switch o conmutador (Sanchez, 2022).

	externos, estos son: Cisco IOS CLI, Cisco Network Assistant, Switching Database Manager templates, Cisco AVVID, CMS.

Fuente: Adaptado de (Cisco, 2009), (Tonitrus, n.d.) y (RenewTech, n.d.)

3.3.3 Firewall Fortigate 1800F

El control de acceso y políticas de seguridad de la red interna de la UTN se encuentran controladas por un firewall primario, este equipo es el Fortigate 1800F mostrado en la **Figura 13**, este es un firewall de siguiente generación o por sus siglas NGFW, que proporciona protección contra amenazas y demás funciones integradas como interacciones SD-WAN, conmutación inalámbrica y compatibilidad con tecnología 5G.

Figura 13.

Firewall Fortigate 1800F

Fuente: Obtenido de (Technologies Inc, 2021)

Al pertenecer la categoría de NGFW cuenta con un alto rendimiento para soluciones empresariales o proveedores de servicios, esto gracias a sus características mencionadas en la **Tabla 7**, donde se resumen de mejor manera las especificaciones del firewall Fortigate 1800F.

Tabla 7.

Características del Firewall Fortigate 1800F

Característica	Descripción
Procesador	Hace uso de procesador SPU NP7 Hyperscale diseñado para un alto rendimiento de hardware admitiendo flujos de tráfico de hasta 100 Gbps, y un

	procesador SPU CP9 para trabajar con el flujo directo de tráfico de la red aportando en procesos de inspección, cifrado y descifrado. Además, incorpora un procesador Intel® Xeon® W-3223 con frecuencia de 3.50GHz y 16 núcleos.
Memoria	- Memoria RAM: cuenta con una memoria volátil de 24102 MB. - Memoria Flash: integra una memoria no volátil de 28738 MB.
Sistema operativo	Emplea el Sistema operative propietario de Fortinet denominado FortiOS, mismo que es la base del software de gestión Fortinet Security Fabric para control de todas las capacidades de seguridad y red del firewall.
Interfaces	Cuentan con 6 tipos diferentes de interfaces a nivel de hardware. - 2 interfaces GE RJ-45 (MGMT Ports) - 2 interfaces 10 GE SFP+ / GE SFP HA slots - 16 interfaces GE RJ-45 ports - 8 interfaces GE SFP slots - 12 interfaces 25 SFP28 / 10 GE SFP+ / GE SFP slots - 4 interfaces 40 GE QSFP+ slots
Puertos	Los puertos disponibles a nivel de hardware se pueden clasificar en base a la función específica que cumplen las interfaces mencionadas previamente. - 16 puertos de alta velocidad RJ-45 - 2 puertos de administración GE RJ-45 - 1 puerto USB 3.0 - 1 puerto de consola RJ-45
Throughput	Los valores de Throughput del firewall varían en base a la función de seguridad a la que corresponde, por lo que se considera de la siguiente forma: - IPS Throughput: 13 Gbps - NGFW Throughput: 11 Gbps - Throughput de protección contra amenazas: 9.1 Gbps - SSL-VPN Throughput: 11 Gbps IPSec VPN Throughput: 55 Gbps - Throughput IPv4 para 1518/ 512/ 64 bytes en UDP: 198 / 197 / 140 Gbps - Throughput IPv6 para 1518/ 512/ 64 bytes en UDP: 198 / 197 / 140 Gbps
Latencia	3.22 μs
Sesiones	Al ser un firewall de próxima generación admite un total de 12 millones de sesiones simultáneas de tipo

	TCP, y hasta 750000 nuevas sesiones TCP al mismo tiempo.
Túneles	La tunelización admitida por el firewall considera relevantes dos tipos de túneles en concreto soportando los siguientes valores de túneles simultáneos: - Gateway-to-Gateway IPsec VPN Tunnels: 20000 túneles simultáneos. - Client-to-Gateway IPsec VPN Tunnels: 100000 túneles simultáneos.
Certificaciones	ICSA Labs: Firewall, IPsec, IPS, Antivirus, SSL-VPN; USGv6/IPv6

Fuente: Adaptado de (Technologies Inc, 2021), (JMTelcom, n.d.)

3.3.4 Switch Cisco C9407R

El Switch Cisco Catalyst 9407R presentado en la **Figura 14**, es un equipo diseñado con la finalidad de soportar grandes cantidades de tráfico. Cuenta con varios módulos que permiten que este equipo sea empleado en redes de core, distribución y núcleo en base a las necesidades de la empresa, cuenta con funcionalidades orientadas a la resiliencia y escalabilidad de las redes al tener compatibilidad con tecnologías como Virtual StackWise, una tecnología de virtualización de sistemas de red, de la cual se hace uso en la infraestructura de red existente en la UTN para formar el core de la red como se ha mencionado anteriormente.

Figura 14.

Switch Cisco C9407R

Fuente: Obtenido de (Router-Switch.com, n.d.-b)

Los equipos de alto rendimiento como el Switch Cisco C9407R cuentan con diferentes características destacables, estas se han resumido en la ***Tabla 8*** para especificar de mejor manera las funcionalidades que ofrece este equipo.

Tabla 8.

Características del switch Cisco C9407R

Característica	Descripción
Procesador	Como parte de la familia de switches Cisco Catalyst 9400 Series, este equipo cuenta con un procesador denominado Cisco Unified Access Data Plane (UADP) Application- Specific Integrated Circuit (ASIC), así como un CPU de 4 núcleos que opera a 2.4 GHz.
Memoria	- Memoria RAM: cuenta con una memoria RAM de 16 GB tipo DDR4 con velocidad de transferencia de daros de 240 MT/s. - Memoria Flash: memoria no volátil integrada en el dispositivo de 16 GB para almacenamiento del sistema operativo y archivos de configuración.

	Memoria externa: cuenta con compatibilidad con unidades de estado sólido (SSD) interfaz Sata M2 con valores de 240 GB, 480 GB o 960 GB.
Sistema operativo	Al igual que todos los equipos de la familia Cisco Catalyst 9400 Series emplea el sistema operativo Cisco IOS XE.
Slots	7 slots o ranuras.
Ranuras	-Ranuras para tarjetas de línea: cuenta con 5 ranuras para inserción de tarjetas de línea, estas corresponden a la numeración 1, 2, 5, 6 y 7. -Ranuras de módulos supervisores: considerando un total de siete ranuras, los módulos de supervisión corresponden a las ranuras 3 y 4.
Tarjetas de línea	Considerando el máximo de 5 ranuras para la inserción de tarjetas de línea, el switch es compatible con los siguientes módulos: - Módulo UPOE 10/100/1000 de 48 puertos - Módulo 10/100/1000 de 48 puertos - Módulo SFP/SFP+ de 24 - Módulo Multigigabit UPOE de 48 puertos - Módulo SFP de 48 puertos - Módulo SFP de 24 puertos - Módulo POE/POE+ Gigabit Ethernet de 48 puertos - Módulo UPOE+ 10/100/1000 Ethernet Gigabit de 48 puertos - Módulo Multigigabit UPOE+ de 100 Mbps/1 G/2,5 G/5 G de 48 - Módulo Multigigabit UPOE+ de 100 Mbps/1 G/2,5 G/5 G/10 G de 48 puertos - Módulo SFP/SFP+ de 48 puertos
Densidad máxima de puertos	La densidad máxima de puertos admitida por el dispositivo depende del tipo de tarjeta de línea insertada en las diferentes ranuras, de este modo, se determina que los módulos de 48 puertos admiten una densidad máxima de 240 puertos y los módulos de 24 puertos alcanzan una densidad máxima de 240 puertos.
Tiempo medio entre fallos (MTBF)	1571010 horas.
Escalabilidad máxima de ancho de banda	Cada una de las tarjetas de línea insertadas en el switch tiene una escalabilidad máxima de su ando de banda de 480 Gbps en todas las ranuras.

Plataformas de supervisión compatibles	Cuenta con 3 motores de supervisión compatibles con la versión de software correspondiente: C9400-SUP-1 C9400-SUP-1XL, C9400-SUP-1XL-Y.

Fuente: Adaptado de (Cisco, 2019), (Cisco, 2020d) y (Router-Switch.com, n.d.-b)

3.3.5 Switch Cisco C9200

El Switch Cisco Catalyst 9200 mostrado en la **Figura 15**, es un equipo altamente empleado en redes de distribución y acceso en las que se requiere de una cantidad considerable de puertos para interconexión de equipos, por lo que es posible encontrar distintas versiones del conmutador desde 24 puertos, además cuenta con la característica de ser apilable permitiendo que estos se adapten de mejor forma la escalabilidad que se presente en la red.

Figura 15.

Switch Cisco C9200

Fuente: Obtenido de (Cisco, 2020a)

Una de las principales características de la familia de Switches Cisco Catalyst es la compatibilidad existente con varios protocolos y equipos de la propia serie, por lo que en la **Tabla 9**, se resumen algunas de sus características más relevantes.

Tabla 9.

Características del switch Cisco C9200

Característica	Descripción
Procesador	Dentro de la familia Catalyst 9000, se emplean distintos procesadores, para el caso del Cisco C9200 se emplea un procesador Broadcom BCM56150 con núcleos de procesamiento ARM-Cortex-A57 que alcanza una frecuencia de hasta 1.5 GHz.
Memoria	- Memoria RAM: cuenta con una memoria RAM de 2 GB. - Memoria Flash: memoria no volátil integrada en el dispositivo de 4 GB para almacenamiento del sistema operativo y archivos de configuración.
Sistema operativo	Trabaja con el sistema operativo integrado de la familia Catalyst 9000, el sistema Cisco IOS XE, es un sistema Operativo que se basa se en licencias propietarias de cisco con soporte para programabilidad.
Puertos	48 puertos
Tamaño de tabla de direcciones MAC	16000 entradas en la tabla MAC.
Interfaces	- 48 puertos Ethernet de alta velocidad (10GE, 25 GE o 40 GE) RJ-45 - 1 puerto de consola RJ-45
Número de VLANs admitidas	4096 VLANs
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.1D,IEEE 802.1Q,IEEE 802.1p,IEEE 802.1s,IEEE 802.1w,IEEE 802.1x,IEEE 802.3,IEEE 802.3ab,IEEE 802.3ad,IEEE 802.3af,IEEE 802.3at,IEEE 802.3bz,IEEE 802.3u,IEEE 802.3x,IEEE 802.3z.
Especificaciones	Equipo de alto rendimiento que opera en capa 3 del modelo de referencia OSI, por lo que cuenta con capacidad de ruteo IP, creación de redes de área local virtuales (VLANs) y listas de acceso. Opera con tecnología de tipo alámbrica, admitiendo anchos de banda de 336 Gbps, una capacidad de switching de 176 Gbps, tasas de reenvío de 130.95 Mpps y la implementación de acceso definido por software.

Seguridad	Emplea el algoritmo de cifrado AES-128 MACsec, así como segmentación basada en políticas.
Protocolos de gestión	Maneja varias versiones del protocolo de administración simple de red, estas son: SNMPv1/v2c/v3

Fuente: Adaptado de (Cisco, 2020a) y (BRAINCORP TECNOLOGIA E INFORMATICA S.A, n.d.)

3.3.6 Switch Cisco C9300

El switch Cisco Catalyst 9300 mostrado en la **Figura 16** es un equipo empleado en redes acceso y distribución, debido a la cantidad de puertos que posee y a las características de alto rendimiento con las que cuenta. Este switch perteneciente a la familia de Cisco Catalyst permite la integración de plataformas de gestión que facilitan el proceso de control y gestión de redes con gran densidad de clientes.

Figura 16.

Switch Cisco C9300

Fuente: Obtenido de (Router-Switch.com, n.d.-a)

El Switch Cisco C9300 cuenta con muchas consideraciones importantes al momento de gestionar una red de acceso o distribución, sus principales características se han resumido en la **Tabla 10** con la finalidad de mantener un panorama claro de las facultades que tiene el equipo y se realice un análisis correcto de su incorporación en dichas redes.

Tabla 10.

Características del switch Cisco C9300

Característica	Descripción
Procesador	Hace uso del procesador UADP (Cisco Unified Access Data Plane) 2.0 ASIC (Application- Specific Integrated Circuit), este proporciona un alto rendimiento y se emplea en equipos de la familia Catalyst 9000.
Memoria	- Memoria RAM: cuenta con una memoria RAM de 8 GB. - Memoria Flash: memoria no volátil integrada en el dispositivo de 16 GB para almacenamiento del sistema operativo y archivos de configuración.
Sistema operativo	Trabaja con el sistema operativo integrado de la familia Catalyst 9000, el sistema Cisco IOS XE, es un sistema operativo que se basa se en licencias propietarias de cisco con soporte para programabilidad.
Puertos	48 puertos
Tamaño de tabla de direcciones MAC	32000 entradas en la tabla MAC.
Interfaces	- 48 puertos Ethernet de alta velocidad (10GE, 100 GE o 1000 GE) RJ-45 - 1 puerto de consola RJ-45 - 1 puerto USB 2.0
Número de VLANs admitidas	4094 VLANs
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.1D,IEEE 802.1Q,IEEE 802.1p,IEEE 802.1s,IEEE 802.1w,IEEE 802.1x,IEEE 802.3ad,IEEE 802.3af,IEEE 802.3ap,IEEE 802.3bz,IEEE 802.3u,IEEE 802.3z.
Especificaciones	Equipo de alto rendimiento que opera en capa 3 del modelo de referencia OSI, por lo que cuenta con capacidad de ruteo IP, creación de redes de área local virtuales (VLANs) y listas de acceso. Opera con tecnología de tipo alámbrica, admitiendo anchos de banda de 480 Gbps, una capacidad de switching de 128 Gbps, tasas de reenvío de 190.48 Mpps, y un tiempo medio entre fallas de 277,70 horas y la implementación de acceso definido por software.

Seguridad	Emplea el algoritmo de cifrado AES-256 MACsec, así como segmentación basada en políticas.
Protocolos de gestión	Maneja distintos protocolos de gestión remota para la administración del dispositivo: SNMP 1, RMON 1, RMON 2, SNMP 3, SNMP 2c, CLI, NETCONF, RESTCONF.

Fuente: Adaptado de (Cisco, 2024a), (Router-Switch.com, n.d.-a) y (almacen-informtico.com, n.d.)

3.3.7 Switch Cisco C2960

El switch Cisco C2960 es un equipo que entra en la categoría de swiches clásicos dentro de las redes de área local, se consideran equipos de buen rendimiento ideales para redes de acceso por su sencilla administración y alta densidad de puertos, el componente de hardware presentado en la **Figura 17** corresponde a una versión de 48 puertos de este equipo.

Figura 17.

Switch Cisco C2960

Fuente: Obtenido de (datacenter360, n.d.)

Al tratarse de una solución de red bastante usada cuenta con varias características importantes, estas se han resumido en la ***Tabla 11***, considerando los aspectos más relevantes destacados por el fabricante.

Tabla 11.

Características del switch Cisco C2960

Característica	Descripción
Procesador	Cuenta con un procesador PowerPC 405 de arquitectura RISC (Reduced Instruction Set Computing) que garantiza equilibrio ente rendimiento, eficiencias energética y velocidad de conmutación.
Memoria	- Memoria DRAM: cuenta con una memoria de tipo volátil de 64 MB. - Memoria Flash: memoria no volátil integrada en el dispositivo de 32 MB para almacenamiento del sistema operativo y archivos de configuración. - Memoria interna: cuenta con una memoria interna adicional de 64 MB.
Sistema operativo	Trabaja con el sistema operativo integrado Cisco IOS LAN, mismo que se basa en la prestación de funcionalidades de conmutación y gestión de red de forma limitada.
Puertos	48 puertos
Tamaño de tabla de direcciones MAC	8000 entradas en la tabla MAC.
Interfaces	- 48 puertos Ethernet (10FE o 100 FE) RJ-45 - 2 puertos SPF de doble propósito - 1 puerto de consola RJ-45
Número de VLANs admitidas	255 VLANs
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.1D,IEEE 802.1Q,IEEE 802.1p,IEEE 802.1s,IEEE 802.1w,IEEE 802.1x,IEEE 802.3,IEEE 802.3ab,IEEE 802.3ad,IEEE 802.3af,IEEE 802.3u,IEEE 802.3x,IEEE 802.3z
Especificaciones	Switch de acceso de alta capacidad que trabaja en la capa 2 del modelo de referencia OSI, cuenta con soporte para VLANs y admite hasta 9016 bytes en jumbo frames. Opera con tecnología de tipo alámbrica, admite ancho de banda de 16 Gbps, una capacidad de switching de 12.8 Gbps, tasas de reenvío de 9.52 Mpps, y un tiempo medio entre fallas de 336983 horas.

Seguridad	Es compatible con protocolos de seguridad 802.1X RADIUS para autenticación y SSH-2 para administración remota.
Protocolos de gestión	Maneja distintos protocolos de gestión remota para la administración del dispositivo SNMP 1, RMON 1, RMON 2, RMON 3, RMON 9, Telnet, SNMP 3, SNMP 2c, HTTP

Fuente: Adaptado de (Cisco, 2008), (intercompras.com, n.d.-b) y (ds3comunicaciones.com, n.d.)

3.3.8 Switch Cisco C3850

El Switch Cisco C3850 presentado en la **Figura 18** es una solución de red que cuenta con un alto rendimiento y una gestión destacable. Considera la integración de elementos que portan robustes a redes de alta demanda siendo considerado ampliamente en redes empresariales o de centros de datos.

Figura 18.

Switch Cisco C3850

Fuente: Obtenido de (Cisco, 2018)

Dentro de la **Tabla 12** se resumen de forma clara las características más relevantes del switch Cisco Catalyst 3850, que al igual que otros modelos de la familia Catalyst brindan a los administradores alternativas de distintos módulos de gestión para un mejor control de la red.

Tabla 12.

Características del switch Cisco C3850

Característica	Descripción
Procesador	Hace uso del procesador UADP (Cisco Unified Access Data Plane) ASIC (Application- Specific Integrated Circuit), este proporciona un alto rendimiento y se emplea en equipos de la familia Catalyst.
Memoria	- Memoria DRAM: cuenta con una memoria volátil de 4 GB. - Memoria Flash: memoria no volátil integrada en el dispositivo de 2048 MB para almacenamiento del sistema operativo y archivos de configuración. - Memoria interna: cuenta con una memoria integrada de 4096 MB para almacenamiento del equipo.
Sistema operativo	Trabaja con el sistema operativo integrado de tipo propietario diseñado por Cisco, este es el Cisco IOS Software mismo que cuenta con soporte para funciones como enriamiento, QoS, Flexible NetFlow, etc.
Puertos	48 puertos
Tamaño de tabla de direcciones MAC	32000 entradas en la tabla MAC.
Interfaces	- 48 puertos Ethernet de alta velocidad (10GE, 100 GE o 1000 GE) RJ-45 - 1 puerto de consola RJ-45
Número de VLANs admitidas	4000 VLANs
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.11ac,IEEE 802.1D,IEEE 802.1p,IEEE 802.1Q,IEEE 802.3,IEEE 802.3ab,IEEE 802.3ad,IEEE 802.3af,IEEE 802.3u,IEEE 802.3x,IEEE 802.3z.
Especificaciones	Equipo de alto rendimiento por lo que admite creación de redes de área local virtuales (VLANs). Opera con tecnología de tipo alámbrica, admitiendo anchos de banda de 480 Gbps, una capacidad de switching de 176 Gbps, tasas de reenvío de 190.48 Mpps, y un tiempo medio entre fallas de 303660 horas. Cuenta con hasta tres niveles de licencia de Cisco que son LAN BASE, IP BASE, IP SERVICES.

Seguridad	Es compatible con protocolos de seguridad 802.1X RADIUS para autenticación. Además guarda compatibilidad con AES-256 con MACSEC de 256 bits.
Protocolos de gestión	Maneja distintos protocolos de gestión remota para la administración del dispositivo: SNMP v1, SNMP v2c, SNMP v3.

Fuente: Adaptado de (Cisco, 2018), (intercompras.com, n.d.) y(MercadoIT, n.d.)

3.3.9 Controladora Cisco 9800

La controladora inalámbrica Cisco 9800 forma parte de los equipos de alto rendimiento del fabricante para redes no cableadas. El equipo representado en la **Figura 19**, cuenta con múltiples mejoras a los modelos vistos con anterioridad dentro de la misma marca y se considera aplicable en redes que requieren de una arquitectura robusta y de alta disponibilidad.

Figura 19.

Controladora Cisco 9800

Fuente: Obtenido de (IT Corporation, n.d.)

Tomando como base las especificaciones aportadas por el fabricante, la **Tabla 13** resume de forma puntual las consideraciones más importantes de la controladora inalámbrica Cisco 9800.

Tabla 13.

Características de la controladora Cisco 9800

Característica	Descripción
Procesador	Cuenta con un procesador Intel x86 como base para la arquitectura del equipo, el cual permite que cada

	núcleo del procesador maneje múltiples procesos en simultáneo ofreciendo un alto rendimiento.
Memoria	- Memoria RAM: cuenta con una memoria volátil de 16 GB. - Memoria interna: cuenta con una memoria integrada de 32 GB para almacenamiento del equipo.
Sistema operativo	Cuenta con el sistema operativo integrado Cisco IOS XE, propio de la familia Catalyst 9000. es un sistema operativo que se basa se en licencias propietarias de cisco con soporte para programabilidad.
Protocolos de interconexión	Gigabit Ethernet, 10 Gigabit Ethernet, IEEE 802.11ac Wave 2, IEEE 802.11ac Wave 1.
Compatibilidad	Es compatible con varios protocolos de capa transporte/red como: TCP/IP, UDP/IP, ICMP/IP, IPSec, ARP, BOOTP, DHCP, Bonjour.
Interfaces	- 1 puerto de consola RJ-45 - 1 puerto de consola MicroUSB - 1 puerto USB 3.0 - 1 puerto de gestión RJ-45 - 1 puerto de redundancia RJ-45 - 4 puertos RJ-45 (2.5 GE/1 GE) - 2 puertos de cobre 10 G (Multgigagbit)
Capacidad de usuarios	La capacidad del equipo puede considerarse de las siguientes formas: - APs integrados: 250 - Clientes: 5000 - LANs inalámbricas: 4096 -VLANs compatibles: 4096
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.3, IEEE 802.3u, IEEE 802.1Q, IEEE 802.11b, IEEE 802.11a, IEEE 802.11d, IEEE 802.11g, IEEE 802.11h, IEEE 802.11e, IEEE 802.11n, IEEE 802.11k, IEEE 802.1AX, IEEE 802.11w, IEEE 802.11r, IEEE 802.11u, IEEE 802.11ac Wave 2, IEEE 802.11ac Wave 1, IEEE 802.11ax
Especificaciones	Equipo de alto rendimiento ideado para espacios de alta demanda que combina las características más avanzadas de la familia Catalyst con puntos de acceso Wi-Fi 6. Es altamente flexible y cuanta con enlaces ascendentes de fibra o cobre para mejorar la interconexión, cuenta con un rendimiento de hasta 5Gbps, soporte para VLANs, Wi-Fi Multimedia, ICMP y compatibilidad con múltiples algoritmos de

	cifrado para aportar mayor seguridad a las conexiones existentes.
Seguridad	- Hace uso de algoritmos de cifrado como: Triple DES, AES, IKE, SSL, TLS, WPA, WPA2, PKI, ESP, CRL, CCM, CCMP. - Cuenta con dos modos de autenticación: RADIUS y EAP (Extensible Authentication Protocol)
Protocolos de gestión	Maneja distintos protocolos de gestión remota para la administración del dispositivo: SNMP 1, RMON, Telnet, SNMP 3, SNMP 2c, HTTP, HTTPS, TFTP, SSH.

Fuente: Adaptado de (Cisco, 2020b), (IT Corporation, n.d.) y (Switch-Wifi, n.d.)

3.3.10 Controladora Cisco 5508

La controladora Cisco 5508 mostrada en la **Figura 20** cuenta con un rendimiento consistente para su aplicación en redes de área local de gran escala gracias a su rendimiento y especificaciones propias de su sistema operativo.

Figura 20.

Controladora Cisco 5508

Fuente: Obtenido de (Cisco, 2017)

Dentro de la **Tabla 14**, se especifican las características clave de la controladora Cisco 5508, destacando en l esta las capacidades del dispositivo para la gestión de redes inalámbricas.

Tabla 14.

Características de la controladora Cisco 5508

Característica	Descripción
Procesador	Cuenta con un procesador Intel x86 como base para la arquitectura del equipo, el cual permite que cada núcleo del procesador maneje múltiples procesos en simultáneo ofreciendo un alto rendimiento.
Memoria	- Memoria RAM: cuenta con una memoria volátil de 8 GB. - Memoria interna: cuenta con una memoria integrada de 8 GB para almacenamiento del equipo.
Sistema operativo	Cuenta con el sistema operativo Cisco Wireless LAN Controller (WLC), este sistema operativo es específico a controladoras inalámbricas y ofrece una gestión confiable y centralizada para puntos de acceso, así como la facilidad de aplicar políticas de seguridad y gestión del rendimiento de las redes inalámbricas.
Interfaces	- 8 puertos Ethernet 10/100/1000 Mbps - 1 puerto de servicio RJ-45 - 1 puerto de consola RJ-45 - 1 puerto USB 2.0
Capacidad de usuarios	500 puntos de acceso en simultáneo
Velocidad	Transferencia de datos hasta 0.1 Gbps.
Estándares admitidos	Se rige y adapta a múltiples estándares del IEEE, como, por ejemplo: IEEE 802.11a, 802.11b, 802.11g, 802.11d, WMM/802.11e, 802.11h, 802.11n
Especificaciones	Controladora de tipo centralizada para redes inalámbricas, es compatible con equipos de la familia Cisco como los puntos de acceso Cisco Aironet. Aporta escalabilidad y flexibilidad a redes medianas y grandes debido a sus facultades de configuración y administración. Cuenta con detección y protección contra interferencias de radiofrecuencias aportando una pronta resolución de problemas en las redes inalámbricas.
Seguridad	Cuenta con distintos modos de autenticación: IEEE 802.1X, RADIUS, TACACS.
Protocolos de gestión	Maneja distintos protocolos de gestión remota para la administración del dispositivo: SNMP v1, v2c, v3, Telnet, TFTP, SNTP, HTTP

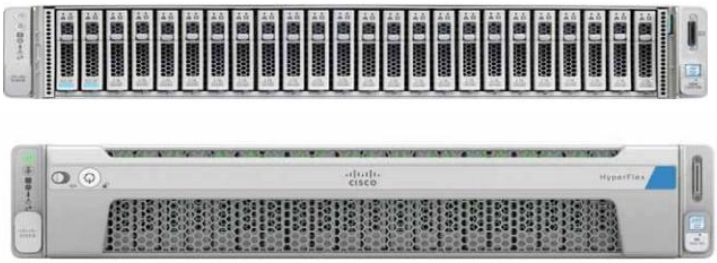
Fuente: Adaptado de (Cisco, 2017), (intercompras.com, n.d.-a) y (IT Planet, n.d.)

3.3.11 Cisco Hyperflex HX220C-M5

Los sistemas de Hiperconvergencia de Cisco se caracterizan por ser soluciones innovadoras para la optimización de estructuras tecnológicas en redes de gran escala al integrar cómputo en la nube y elementos de virtualización que mejoran las características de los componentes de hardware planteado. La **Figura 21** representa un nodo de la familia de equipos HyperFlex, mismo que forma parte del sistema total que se encuentra integrado por más de un elemento.

Figura 21.

Cisco Hyperflex HX220C-M5



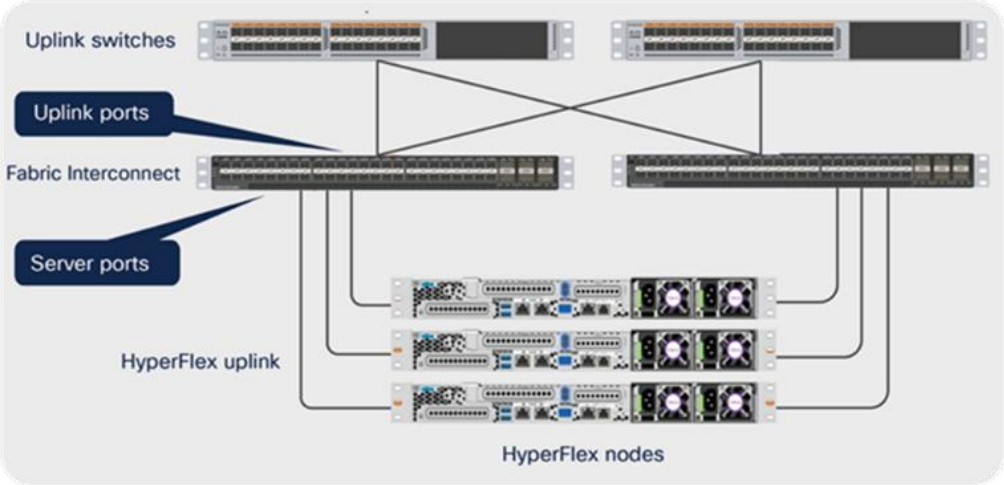
Fuente: Obtenido de (mosupervs.live, n.d.)

Como se ha mencionado anteriormente, la Hiperconvergencia de Cisco se compone de varios equipos, para este caso en particular, tomando referencia la **Figura 22**, se cuenta con tres nodos Cisco HyperFlex y dos Fabric Interconncet. La interconexión entre los nodos se realiza mediante una red de alta velocidad para mantener la comunicación eficiente entre los nodos, pues para formar el clúster de HyperFlex se requiere que cada uno de los nodos integrados contribuya con sus recursos de cómputo, almacenamiento y red. Para lograr la interconexión, se requiere que los nodos guarden compatibilidad entre sí, y que se haga uso de equipos de cada dos como los Cisco Fabrics Interconnects, mismos que son componentes de la infraestructura Cisco UCS (Unified Computing System) para

actuar como puntos de conexión entre los nodos y la red garantizando de este modo la conectividad y gestión centralizada (Cisco, 2020c).

Figura 22.

Diagrama de conexión de nodos Cisco Hyperflex



Fuente: Obtenido de (Cisco, 2020c)

Dentro de la **Tabla 15** se resumen las características claves del sistema de Hyperconvergencia de Cisco HyperFlex. Se han considerado detalles sobre la capacidad en los nodos de esta serie para así contar con un mejor panorama del rendimiento general de estos sistemas.

Tabla 15.

Características de hiperconvergencia Cisco Hyperflex HX220C-M5

Característica	Descripción
Procesador	Cuenta con un procesador Intel Xeon Scalable de 14 nanómetros de alto rendimiento con soporte para añadir Intel Automated Vector Extensions 2 (AVX2) con la finalidad de mejorar el rendimiento de la infraestructura.
Memoria	Ofrece capacidad de hasta 3 TB de memoria con tecnología DDR4 para mantener un perfil de consumo bajo y ofertar altas velocidades. Las configuraciones de memoria son de tipo híbrida contando con memoria

	flash o totalmente NVMe (Non-Volatile Memory Express) que permiten mejorar el rendimiento al incorporar también la virtualización para expandir la capacidad final.
Sistema operativo	Se basa en el software incluido Cisco HyperFlex HX Data Platform, el cual admite capacidades avanzadas de almacenamiento, así como sus características más relevantes que son la administración y automatización para la infraestructura y la escalabilidad y flexibilidad al admitir la integración en horizontal de nodos compatibles con el sistema base integrado.
Interfaces	- 1 puerto de gestión Ethernet Gigabit Ethernet RJ-45 - 2 puertos 10 Gigabit Ethernet RJ-45 - 1 puerto serie RS-232 - 1 puerto de vídeo VGA - 2 puertos USB 3.0
Compatibilidad	Es compatible con conexiones Ethernet de 40 Gigabits de baja latencia, así como con entornos de virtualización de servidores como VMware vSphere, computación en la nube, infraestructura de escritorio virtual (VDI), bases de datos SQL, Oracle y SAP. Además, proporciona soporte para protocolos de red estándar para garantizar conectividad confiable.
Administración	La administración de HyperFlex es compatible con la plataforma Cisco Intersight™ misma que permite la automatización de flujos de trabajo, además ofrece soporte para complementos de VMware vSphere y una interfaz en HTML 5 para administración web.
Modo de operación	Se considera que se despliegue en forma de nodos que se agrupa para formar clústers en los que se consideran como una infraestructura única que admite altas cargas de trabajo con una gestión centralizada mediante la plataforma de gestión propietaria de Cisco. Cuenta con una capacidad de adaptación a entornos altamente congestionados, permitiendo gestionar de manera eficiente las necesidades de almacenamiento y velocidad de las empresas en aplicaciones críticas.

Fuente: Adaptado de (Cisco, 2021a)

3.3.12 Chasis Blade HP BladeSystem C7000

El Chasis Blade representado en la **Figura 23** es a una solución de infraestructura diseñada especialmente para centros de datos. Este chasis permite

alojar varios servidores en un mismo bastidor permitiendo así optimizar el espacio dentro de los datacenters y proporcionar flexibilidad en la distribución de equipos en un espacio compacto.

Figura 23.

Chasis blade HP BladeSystem C7000

Fuente: Obtenido de (Montez, 2006)

Las características más relevantes respecto a la infraestructura del Chasis Blade de HP se encuentran resumidas en la **Tabla 16**. Esta tabla especifica consideraciones claves para el uso adecuado del equipo.

Tabla 16.

Características del Chasis Blade HP BladeSystem C7000

Característica	Descripción
Espacio de Rack	Es un chasis de 10U (10 unidades de rack) que admite dos distribuciones: - 16 servidores de altura media - 8 servidores de atura completa
Compatibilidad	El BladeSystem es compatible con equipos: Blades HP ProLiant, Integrity y Storage

Fuentes de alimentación	Cuenta con 6 unidades de alimentación de 240 V "hot-swap" o en caliente, es decir no se requiere que se apague la infraestructura para realizar el intercambio.
Plataforma de administración	Cuenta con una plataforma de administración de la infraestructura completa, 456204-B21 HP BLc7000 DDR2 Enclosure Management Option para gestionar todo el chasis y los servidores.
Refrigeración	10 ventiladores incorporados

Fuente: Adaptado de (Give 1 Life, n.d.) y (DC Parts, n.d.)

3.3.13 Servidor HP ProLiant DL360 Gen9

El servidor HP ProLiant DL360 Gen9 pertenece a una familia de servidores que conforman soluciones de infraestructura diseñadas para ofrecer un alto rendimiento y altas tasas de disponibilidad en entornos empresariales al ser fácilmente adaptables a distintos entornos dentro centros de datos o redes de gran volumen, la **Figura 24** muestra el componente en hardware del equipo previa a ser montado en un Rack o en un chasis apilable.

Figura 24.

Servidor HP ProLiant DL360 Gen9

Fuente: Obtenido de (PST Colombia, n.d.)

Las características más relevantes del servidor se encuentran resumidas en la **Tabla 17**, equipo está diseñado para soportar cargas de trabajo intensivas y adaptarse sin dificultad a redes de gran escala.

Tabla 17.

Características del Servidor HP ProLiant DL360 Gen9

Característica	Descripción
Procesador	Cuenta con procesador Intel® Intel Xeon E5 v3 de 8 núcleos de 16 hilos con una frecuencia de 3.2 GHz y un caché de 20 MB. A nivel de procesador cuenta con una velocidad de transferencia de datos del bus del sistema de 8 GT/s.
Memoria	Cuenta con una memoria interna de 32 GB de tipo DDR4-SDRAM distribuida en 2 ranuras de 16 GB cada una, el máximo de memoria interna corresponde a 384 GB con una frecuencia de reloj en memoria de 2133 MHz. Su memoria RAM es de 16 GB.
Sistema operativo	No cuenta con un sistema operativo integrado por defecto, sin embargo, guarda compatibilidad con distintos SO como: Microsoft Windows Server, Canonical Ubuntu, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Oracle Solaris, VMware, Citrix XenServer.
Red	Compatible con tecnología de cableado 10/100/1000Base-T(X) para interfaces Gigabit Ethernet, cuenta con un controlador LAN Broadcom 5720.
Interfaces	- 4 puertos Ethernet LAN RJ-45 - 1 puerto serial - 1 puerto BGA (D-sub) - 3 puertos USB 3.0
Ranuras de expansión	Cuenta con 2 ranuras PCI-x en versión 3.0
Plataformas de administración	Cuenta con diferentes plataformas de administración de servidores: Administración de infraestructura iLO Management (estándar), Intelligent Provisioning (estándar), iLO Advanced (estándar), HP OneView Advanced (opcional), HP Insight Control (opcional).
Almacenaje	Es compatible con RAID (Redundant Array of Independent Disks) en niveles 0, 1, 5 y 10 , tiene una capacidad de 8 discos duros con una capacidad máxima de almacenaje de 16 TB.
Adaptador gráfico	Cuenta con un control de gráficos dividido en dos componente, un adaptador gráfico G200 y un chipset gráfico Matrox.

Fuente: Adaptado de (DIGITALIFE, n.d.) y (HP, 2014)

3.4 Definición de soluciones SDN

La evaluación de soluciones para redes definidas por software es crucial en el proceso de selección de estrategias de implementación para el proceso de migración dentro de una red. Estas soluciones se diseñan con la finalidad de gestionar el tráfico de las redes de forma dinámica y automatizada considerando distintas tecnologías acorde a los distintos tipos de SDN (Plesant, 2023). Se pueden considerar diferentes enfoques, como las soluciones Open SDN (SDN abiertas), que permiten implementaciones basadas en código abierto y estrategias no propietarias. Por otro lado, existen soluciones propietarias, en las que la participación de los fabricantes de dispositivos es fundamental para aportar con soluciones precisas para ofrecer el mejor servicio posible. Por esta razón se deben conocer las implicaciones de cada una de estas alternativas para seleccionar aquella que se adapte mejor a las necesidades de la red actual del datacenter.

3.4.1 Soluciones no propietarias

Las soluciones no propietarias y de código abierto se enfocan en dar al administrador de red la libertad de programar las reglas de flujo de tráfico en la red SDN con mayor libertad al no sujetarse a norma establecidas por los fabricantes directos de los dispositivos de red. Este tipo de soluciones se basa en la compatibilidad de los conmutadores de red definida por software compatibles con el protocolo OpenFlow (Raza et al., 2014). Si bien, las soluciones no propietarias pueden aportar mayor flexibilidad y libertad a la red, traen consigo dificultades enfocadas a la falta de soporte y documentación propia para la resolución de problemas, más esto no aparta a estas soluciones del mercado ya que distintos productos como Big Switch Networks trabajan bajo esta estructura.

Las soluciones abiertas pueden ofertar un controlador OpenFlow para la programación de los conmutadores de la red, estos controladores deben ofrecer algunas funcionalidades como la gestión de estado de red por lo que deben contar con un récord o registro de los elementos que se están gestionando. Se considera también que muchos de estos controladores cuentan con un conjunto de APIs que facilitan la interacción del controlador con elementos de programación que incrementan el nivel de complejidad final de la red (García Centeno et al., 2014).

Con una Open SDN se pueden aplicar estructuras de controladores centralizados que admiten un control total de los dispositivos de red gestionados, teniendo de este modo una visión completa de la red que facilite al administrador la toma de decisiones respecto al flujo de tráfico existente de la red. Sin embargo, se debe considerar que, estas soluciones incorporan a su vez características de las implementaciones de tipo superpuesta (SDN overlay) y SDN API (Goransson et al., 2017).

Se debe recalcar que, los tipos de soluciones SDN ya sean de tipo superpuesta, o basadas en API, pueden formar parte de una solución de código abierto, pues, ciertas características de SDN overlay permiten solucionar limitaciones en el manejo de tráfico con una cantidad significativa de VLANs haciendo uso de tecnologías de tunelización como VXLAN, NVGRE o STT (Goransson et al., 2017). Si bien las soluciones Open SDN pueden estar sujetas a limitaciones de red, como la capacidad de las tablas MAC de los conmutadores de red, número máximo de VLANs soportados o las limitaciones propias de la red física, es posible incorporar aspectos de programación de red enfocados a la modificación de recursos para obtener un mejor resultado en el rendimiento final de las redes que adoptan este tipo de soluciones.

Otro aspecto importante que tomar en cuenta es que el uso de soluciones Open SDN propone emplear controladores de tipo centralizado que permiten tener una visión completa de la red, así como tomar decisiones de enrutamiento e ingeniería de tráfico coherentes, facilitando la gestión y la recuperación de fallos.

3.4.2 Soluciones propietarias

Una solución propietaria hace referencia a un conjunto de equipos y protocolos propios de un fabricante de dispositivos de red orientadas a brindar las características de una red definida por software, sin embrago en muchos casos se considera que este tipo de soluciones impone limitaciones a la programabilidad de la red que es una de las principales características de la red definida por software, sin embargo, este hecho no es acertado del todo, pues se considera que los fabricantes no cierran totalmente sus esquemas y permiten incorporar políticas que se acoplen a las necesidades de los administradores de red con la peculiaridad de contar con mayor documentación y soporte para sus componentes.

Al tomar como referencia la evolución de las redes de datos se puede evidenciar la tendencia a la programabilidad de las redes. La programabilidad antes mencionada se enfoca en acuerdos de nivel de servicios y no bajo protocolos de red tradicionales haciendo que, los fabricantes deban planificar soluciones enfocadas al trabajo conjunto de políticas de flujo para el tráfico de red y de conexiones dinámicas y flexibles entre los equipos de una misma red (Raza et al., 2014).

Varios fabricantes de dispositivos han incursionado en el mercado con controladores propios y aplicaciones de gestión y administración como parte de sistemas de red programables ofertados en el mercado. Proveedores con alta demanda de equipos de red como Cisco ofrecen enfoques híbridos enfocados a la

migración progresiva a una estructura de red definida por software, además con la finalidad de lograr la separación del plano de control tradicional del plano de datos , para así integrar un controlador externo que vuelva a la red programable, ha integrado en sus equipos la compatibilidad con el protocolo de código abierto OpenFlow (Raza et al., 2014).

Muchas de las soluciones propietarias de redes definidas por software se caracterizan por su fácil implementación en entornos nuevos, esto debido a que plantean a los usuarios finales un modelo adaptable a los beneficios que ofrecen y que admite fácilmente la incorporación de nuevos dispositivos que ya se encuentran pensados para funcionar dentro de estos entornos, sin embargo incurren en altos costes relacionados a la adquisición de equipamiento y licencias disponibles para la gestión y administración de la red, por lo que, a nivel empresarial requieren de complejos procesos de planificación para aceptar el levantamiento de una nueva red.

En el mercado existen varios tipos de soluciones propietarias que pueden incorporarse a redes actuales mediante la actualización de firmware[8] en el hardware existente para de esta forma permitir incorporar elementos programables a la red de forma gradual.,Con este panorama, no solo Cisco se ha planteado la incorporación de mayor compatibilidad, pues en el mercado se han logrado posicionar otros fabricantes como: Juniper Netoworks, VMWare, Dell, IBM, entre otros que plantean sus propias estrategias de integración de programación a las redes tradicionales (BasuMallick, 2022)

[8] Firmware: software integrado en dispositivos de hardware para el control de las funciones de estos, es actualizable y se almacena en la memoria no volátil (Malwarebytes, 2024).

3.4.3 Soluciones SDN overlay

Una red definida por software sobrepuesta o SDN overlay se basa en la virtualización y en la superposición de la SDN sobre la estructura física de la red actual. Este tipo de soluciones se centran en brindar una solución de la limitación de tablas MAC y del manejo de VLANs dentro de los conmutadores SDN incluidos en la red, pues estas hacen uso de nuevas técnicas de tunelización altamente escalables como VXLAN, NVGRE o STT (Goransson et al., 2017), aportando así mayor flexibilidad a la red, al admitir un mayor número de segmentos de red sin saturar las tablas MAC de los conmutadores de red.

Las soluciones relacionadas a la superposición pueden ser incorporadas a las estrategias planteadas por determinados fabricantes con la finalidad de ofrecer soluciones adaptables a las estructuras de red existentes, permitiendo de esta forma acoplar las nuevas tecnologías sin comprometer la infraestructura existente.

Sin embargo, estas soluciones pueden acoplarse a entornos no propietarios mediante el uso de tecnologías de código abierto, mismas que se integran de forma directa con OpenFlow para admitir flujos de tráfico programables y de esta manera usar de mejor forma los recursos de las redes tradicionales que se empleaban anteriormente.

3.4.4 Soluciones SDN API

Una red definida por softwrae basada en APIs se caracteriza por enfocarse en la programabilidad y automatización de operaciones en la red mediante programación de aplicaciones facilitando la implementación de políticas de red y gestión de los recursos. Las soluciones basadas en APIs, proporcionan interfaces de programación estandarizadas con un alto grado de flexibilidad y adaptabilidad a las necesidades de las redes (Goransson et al., 2017).

Para las soluciones de SDN basadas en APIs, el proceso de integración en sistemas existentes en los escenarios de aplicación es bastante sencillo y se puede realizar de forma gradual permitiendo así maximizar la utilización de recursos dentro de infraestructuras previamente construidas (Goransson et al., 2017). Las APIs se incorporan a la plataforma de soluciones propietarias en muchos casos con la finalidad de aprovechar y lograr integraciones de componentes de hardware y protocolos propietarios para obtener mayor fluidez en la comunicación entre componentes, más estas no se limitan a este modo de operación, pues es posible emplear estas interfaces en soluciones no propietarias basadas en protocolos como NETCONF, MPLS y XMPP que se adapten a las redes planteadas mediante la programación directa de la comunicación de la red.

CAPÍTULO IV

Propuesta Metodológica

4.1 Descripción de requerimientos

Al realizar una transición de una arquitectura tradicional a la de una red definida por software, es necesario tomar a consideración ciertos requerimientos que pueden varía en función de las necesidades operativas de la red. Una de las principales características de las SDN es la alta programabilidad, debido a que es lo que aporta la gestión dinámica y eficiente de los recursos de red.

Los requerimientos para la implementación de una SDN se centran generalmente en la virtualización de redes y de componentes de hardware. Esta virtualización es crucial para obtener un rendimiento óptimo, para de este modo permitir un manejo eficaz del tráfico en la red. Además, es esencial que los dispositivos que formen la red soporten protocolos abiertos y se acoplen a entornos de redes definidos por software sin limitaciones asociadas a políticas de fabricación (Alberti, 2012).

Sin dejar de lado la virtualización, factores como la adaptabilidad y la escalabilidad son de suma importancia (Alberti, 2012). Se considera que la red debe ser capaz de adaptarse a cambios y escalar conforme a las demandas establecidas con una mínima interrupción, esto se considera de suma importancia para mantener la continuidad y la calidad del servicio dentro de la institución.

Estos requerimientos implican que una red definida por software debe contar con una interacción bien coordinada entre software para formar una arquitectura robusta, que sea capaz de soportar de forma eficiente la alta demanda establecida por los propietarios y administradores de la red. Al considerar estos aspectos, se

considera que, la implementación realizada sea efectiva y que a su vez sea sostenible a largo plazo.

4.1.1 Contextualización de requerimientos de software

Una red definida por software cuenta con ciertos requisitos a cumplir para el campo del software, mismos que consolidan las grandes ventajas que ofrecen estas redes a comparación de las redes tradicionales y que a su vez demuestran la factibilidad de esta red en el campo a considerar, por lo que se considera que, a nivel de software una SDN debe cumplir con:

- Simplicidad: las SDN deben emplear modelos semánticos genéricos que simplifiquen la gestión de la red mediante esquemas regulares y comprensibles. Esto implica utilizar modelos de información y comportamiento que faciliten la transición y reorganización de una red tradicional a una definida por software. Esta simplificación se basa en la interacción coherente entre los planos de control y de datos, permitiendo una administración más intuitiva y menos propensa a errores.(Open Networking Foundation, 2015).
- Aplicabilidad: a nivel de software, se espera que una SDN cuente con un nivel de aplicabilidad que pueda evolucionar favorablemente con el tiempo, es decir que pueda dar un soporte completo al escenario actual y que a su vez se incorpore exitosamente a versiones futuras del mismo. Otra consideración de la aplicabilidad se encuentra estrechamente ligada a los tipos de redes en las que pueda coexistir, ya que se busca que, redes de transporte de área extensa, redes de servicio enfocadas, arquitecturas de centros de datos, entre otras, puedan realmente consolidarse como red definida por software (Open Networking Foundation, 2015).

- Interfuncionamiento e interoperabilidad: estas hacen referencia a la compatibilidad que guarden las SDN con las redes existentes tratando de que estas sean lo menos invasivas posibles al no incorporarse del todo en redes de tipo híbridas, y que, por otro lado, en aquellas que si lo son se incorporen de la mejor forma creando normas y políticas para integrar elementos de SDN (controladores, APIs, elementos de tunelización), sin interrumpir el funcionamiento de la red tradicional y que promuevan mejoras en el modo de operación de la misma (Open Networking Foundation, 2015).
- Escalabilidad: este requisito es considerado en distintos niveles dentro de las SDN, sin embargo, se le da una mayor relevancia a nivel de software a la escalabilidad del plano e control, pues si se considera una red lógicamente centralizada en un único punto (controlador), se deben considerar factores de diseño mediante sistemas distribuidos que, permitan aportar con escalabilidad ante el potencial crecimiento de la red, y que aporten disponibilidad y una mejor tolerancia a fallos mediante las funciones definidas por software que se encuentran presentes en este factor clave, dejando que gran parte del factor de escalabilidad resida de forma directa en la parte no tangible de la red en cuestión (Open Networking Foundation, 2015).
- Seguridad: la seguridad a nivel físico pasa a segundo plano al hablar de una red definida por software, pues no guarda mucha diferencia con la seguridad que se debe tener para una red tradicional, sin embargo, a nivel de software, se deben integrar mecanismos enfocados por ejemplo a: autenticación de entidades haciendo referencia a controladores y elementos, control de

acceso de APIs e interfaces, cifrado y tunelización que garanticen la integridad, confidencialidad y disponibilidad de la red en todo momento (Open Networking Foundation, 2015).

Estos requisitos generales, son según (Open Networking Foundation, 2015), fundamentales para cualquier red definida por software que se busque emplear en cualquier ámbito.

Además de los requisitos primordiales que definen y caracterizan una red definida por software, existen requerimientos técnicos específicos que son cruciales al momento de realizar una implementación de SDN dentro de una red existente o una nueva red. Los requerimientos técnicos se encuentran estrechamente ligados a la asignación de recursos para la red general y en la eficiencia del núcleo de esta, el controlador SDN, como se indica según (Hata, 2013). Sin embargo, el controlador SDN no opera de forma aislada y no es el único componente de software crítico, pues, la estructuración de la red y las demandas específicas que hacen que se deba considerar elementos técnicos esenciales que se describen a continuación:

a) Controlador SDN

Para todo controlador de una red definida por software, se considera principalmente la existencia de uno o varios controladores que interactúan con las aplicaciones (APIs) y dispositivos de red presentes en toda la arquitectura, los componentes de un controlador se basan bajo el principio de emplear southbound interfaces hacia los dispositivos para realizar funciones de control y gestión. En los dispositivos SDN el controlador actúa como el núcleo o cerebro de los mismos ya que indica las configuraciones de los dispositivos y los modos de operación que estos deben tomar en la red. Además, dentro de la red SDN, el controlador es

pieza fundamental en todos los proceso de gestión de recursos y la asignación de funcionalidades a toda la red (Hoang & Pham, 2015).

En base a su funcionalidad, un controlador SDN puede componerse de varias partes, que caracterizan su alcance dentro de la red y las funciones que debe cumplir dentro de la misma, estos componentes son:

- Controlador y unidad de gestión de conmutación: gestiona todos los switches conectados al controlador y se admite la aplicación de cambios y envío de mensajes para el modelado del flujo de tráfico dentro de los mismos (Hoang & Pham, 2015).
- Unidad de procesamiento de paquetes: hace referencia al elemento que procesa las cargas útiles en base a sus protocolos de red, cada uno de estos se modela en función de sus componentes: dirección origen y destino, tipo de mensaje enviado y la carga útil, esto permite que el controlador determine las reglas de flujo a emplear en cada uno de estos casos (Hoang & Pham, 2015).
- Gestor de dispositivos: se encarga de mantener un registro de todos los dispositivos del controlador, realizando un modelado que permita identificara cada uno de los dispositivos en base a su dirección de capa enlace de datos, direcciones asignadas y puertos de conmutación definidos (Hoang & Pham, 2015).
- Gestor de topología: es un módulo funcional orientado principalmente a la gestión de la red al identificar los cambios existentes en la topología de red y enviar actualizaciones a los elementos de esta para indicar posibles cambios asociados a las

variaciones presentadas. Este bloque funcional permite mantener la automatización de la red en cuestión (Hoang & Pham, 2015).

- Enrutamiento: es el bloque funcional dentro del controlador que se encarga de gestionar los protocolos de enrutamiento en base al direccionamiento origen y destino que se haga definido encada uno de los equipos dentro o fuera de la red, se considera que los protocolos de enrutamiento soportados por cada controlador dependerán de su versión y compatibilidad (Hoang & Pham, 2015).
- Módulo OpenFlow: es el módulo principal de los controladores que admite las funciones relacionadas a mensajes OpenFlow, acciones de entrada relacionadas a tablas y reglas de flujo presentes en toda la red, colas de mensajes y generación de variables estadísticas (Hoang & Pham, 2015).
- Servicios y plugnins: permite añadir protocolos nuevos necesarios para operaciones propias del controlador asociada a versiones de protocolos ya empelados, o a nuevos protocolos necesarios de incorporar dentro de la red (Hoang & Pham, 2015).
- Interfaz de gestión: admite conexiones de forma sencilla para acceder a distintas funciones dentro del controlador, puede darse a modo de interfaz web u otras APIs para el acceso directo al controlador (Hoang & Pham, 2015).

En base a estos componentes, es posible agrupar a los controladores de red conforme sus características. Teniendo en cuenta la comparativa evidenciada en la **Tabla 2**, se pueden determinar las posibles aplicaciones

que tiene cada controlador, por lo que, considerando las necesidades de un Datacenter, se propone el uso de ONOS u OpenDaylight, estos controladores, ambos de código abierto pueden incorporarse a redes tradicionales considerando los recursos y compatibilidad que exista en esta.

b) Virtualización

Dentro de las redes definidas por software el componente de virtualización influye en el uso compartido de recursos de red y computación entre los distintos componentes de la red, se considera en este marco las instancias de red virtuales con sus requisitos de conectividad, seguridad y direccionamiento. Para la virtualización hay que considerar los siguientes aspectos, la creación de instancias virtuales, las interacciones entre sistemas de control de tráfico asociadas al sincronismo de la red, el aprovisionamiento de túneles empleado en cifrado y negociación de protocolos de enrutamiento o transporte, comportamiento de túneles el cual se basa en los mecanismos que emplea la red para generar tunelización entre hosts o elementos de red, y la clasificación y procesamiento de tráfico (Open Networking Foundation, 2015).

c) Compatibilidad con estándares y protocolos

Uno de los principios de las redes definidas por software se centra en la compatibilidad que tengan los dispositivos para incorporar estándares y protocolos propios de una SDN que habilitan la comunicación directa de estos dispositivos con elementos de control (controladores) incluidos a la red tradicional para cambiar la arquitectura anterior.

Dentro de las SDN existe un factor de estandarización que comprende la manera en que los dispositivos de red establecen comunicación entre sí, y a su vez, con el controlador que gestiona y controla las políticas de red. Existen distintos estándares para redes definidas por software como es el caso de NETCONF y OpenFlow, siendo este último el más reconocido y sobre el cual se sustentan muchas de las soluciones SDN existentes (Juniper, n.d.).

Si bien en un indicio, los fabricantes de dispositivos de red definieron parámetros de comunicación propios para sus soluciones particulares, el crecimiento de las redes y la incorporación de nuevos elementos gestionables, incentivó a estos mismos fabricantes a acoplarse a los estándares y protocolos abiertos con la finalidad de permitir a sus clientes conformar estructuras híbridas, y que a su vez aporten mayor versatilidad a sus equipos, por lo que incorporaron en las actualizaciones de sus sistemas operativos la posibilidad de incorporar principalmente al protocolo OpenFlow mediante plugins, o instalaciones directas.

Cisco, uno de los mayores fabricantes de dispositivos de red, ha propuesto la incorporación del protocolo OpenFlow en varios de sus dispositivos, principalmente en la serie 9000, esto considerando elementos claves de las redes definidas por software como la escalabilidad y la facilidad de control para los dispositivos. En la mayoría de los dispositivos Cisco en los que se ha añadido la compatibilidad con el protocolo, se define a la versión 1.3 del mismo en switches de capa 3 e incluso capa 2 sin limitar funcionalidades propias como el uso de ACLs, VLANs, routing de IPv4 (Cisco, 2021c).

Tomando como referencia el estudio realizado en el CAPÍTULO II, los elementos de red existentes en el centro de datos UTN son en su mayoría del fabricante cisco, por lo que, en base a su documentación oficial, se puede definir que uno de los requisitos básicos para el uso de OpenFlow, es que el dispositivo físico, y su sistema operativo soporten la instalación del protocolo, para lo cual se debe realizar una consulta dentro de los sitios oficiales, y en base a la misma se puede determinar que conforme a la versión soportada se cuenta con el soporte para controladores evidenciado en la **Tabla 18**.

Tabla 18.

Soporte para controladores ofertados por Cisco

Versión de OpenFlow	**Controladores Compatibles**
OpenFlow 1.0	XNC (Extensible Network Controller)
	1.0
	POX
	Cisco Open SDN Controller
	Ixia Controllers
OpenFlow 1.3	Ixia Controllers
	Cisco Open SDN Controller
	OpenDaylight

Fuente: Obtenido de (Cisco, 2016)

Con la finalidad de dar soporte a este tipo de controladores dentro de los dispositivos de hardware existentes se han planteado distintas actualizaciones a los sistemas operativos IOS de Cisco, en muchos casos OpenFlow actúa como un agente que controla el plano de datos y permite definir, modificar o eliminar flujos de tráfico mediante interfaces estandarizadas para este fin. Con esta incorporación, es posible realizar

operaciones como descarte de paquetes, designación de tráfico a interfaces específicas, cambios en valores de TTL de la cabecera IP, modificaciones en campos de las trama ethernet, entre otras demostrando la alta flexibilidad que aporta OpenFlow a la gestión del tráfico a nivel de software (Cisco, 2016).

d) Inclusión de políticas

Dentro de una red definida por software, la gestión de las políticas es de vital importancia para el tratamiento y control de tráfico dentro de la red. Las políticas son las encargadas de definir como se manejan los flujos de datos en la red siendo determinantes en la escalabilidad de los sistemas. Según (H. Zhang et al., 2019), las políticas presentes en una SDN se implementan mediante la programación de flujos.

Con protocolos como OpenFlow, es posible establecer reglas en el plano de datos. Estas reglas permiten que los dispositivos gestionen los flujos de tráfico. Cuando un dispositivo recibe un paquete, este busca dentro de sus tablas de flujos la entrada que corresponda, de forma convencional, al encontrar coincidencias, procesa el paquete en base a las configuraciones que hayan sido especificadas por el controlador, estableciendo un control sobre el comportamiento del tráfico dentro de la red (H. Zhang et al., 2019).

En caso, de no encontrar ninguna coincidencia para el flujo entrante, se procederá a encapsular y reenviar el paquete hacia el controlador para que este devuelva una nueva entrada a las tablas de flujo de los conmutadores y devuelva el paquete para que los dispositivos instalen estas actualizaciones en sus tablas de flujo y procedan a reenviar

el paquete por la salida que corresponda en base a las instrucciones dadas (H. Zhang et al., 2019).

En este contexto, se destaca la importancia de la aplicación de políticas para determinar el modo de operación de una SCN, así como facilitar la gestión centralizada del tráfico mediante el controlador SDN, por lo que, aplicar políticas que garanticen una respuesta adecuada de la red a las distintas demandas y condiciones de tráfico.

4.1.2 Contextualización de requerimientos de hardware

Para una red definida por software, el componente de hardware es el esqueleto de soporte y ejecución de funciones determinadas por el software en cuestión, este debe cumplir con especificaciones mínimas del estándar relacionadas al rendimiento o a la confiabilidad que se pueda aportar, además debe de ser capaz de integrarse con las tecnologías relacionadas a la SDN como la gestión y control dinámico.

El hardware existente en una red tradicional puede en muchas ocasiones, ser incorporado dentro de una SDN gracias a las capacidades de soporte existentes por varios fabricantes. Mediante actualizaciones de sistema operativo, o de licencias de funcionamiento es posible que muchos equipos se logren integrar a la SDN sin la necesidad de adquirir nuevos dispositivos, sin embargo, estas condiciones son puestas directamente por el fabricante, pues al diseñar las nuevas versiones de sistemas operativos establecen un rango de procesadores y componentes físicos propios de la infraestructura de hardware que son requeridos para poder hacer uso de las nuevas funcionalidades establecidas, esto se hace con la finalidad de no comprometer el rendimiento de los equipos ni sobreexigir sus capacidades.

La elección de hardware adecuado es crucial para el adecuado funcionamiento de la SDN, pues de este depende el rendimiento máximo de la red, la respuesta y la resistencia a fallos, por lo que la robustez que ofrezcan los equipos en conjunto asegura que la infraestructura de red realmente va a tener un acople óptimo al sistema planteado.

Con el análisis realizado en la sección 3.3 Evaluación y diagnóstico**¡Error! No se encuentra el origen de la referencia.** se determina que los equipos que conforman la mayor parte de la infraestructura de red actual son del fabricante Cisco, y según lo establecido en (Cisco, n.d.) tienen compatibilidad con el protocolo OpenFlow, adicionalmente se tiene que es posible integrar soluciones SDN propietarias para una centro de datos. Sin embargo, las limitaciones de Cisco se hallan presentes en el procesador con el que cuentan sus equipos, permitiendo así que los dispositivos de la serie Catalyst 9XXX en su gran mayoría admitan la versión de sistema operativo que habilita funciones de SDN relacionadas a la versión 1.3 de OpenFlow, mientras que, la serie de Catalyst 2960 cuentan con cierto grado de soporte incorporado en sus últimos releases, admitiendo que la serie 2960X y 2960XR puedan actualizar el sistema operativo e integrar estas funcionalidades (Cisco, 2016). Estas especificaciones pueden ser verificadas para cada software dentro del sitio: https://cfnng.cisco.com/compare , que ofrece una comparativa de los equipos que admiten protocolos o configuraciones específicas.

Tomando como punto de partida el recurso de comparativa de Cisco y las especificaciones de los equipos (modelos y sistemas operativos) que se encuentran en existencia dentro del data center se puede realizar una verificación de esta compatibilidad, la **Figura 25** evidencia la consulta realizada para el switch

Cisco Catalyst 9300 con sistema operativo IOS XE 17.3.4, en la cual se indica que si admite OpenFlow en su versión 1.3

Figura 25.

Protocolos compatibles con el switch Cisco Catalyst 9300

All Features Search	CAT9300 IOS XE - 17.3.4
NSF Support for IPv6	✓
NTP MIB	✓
NTP Server for Access Point	✓
NTPv4 Orphan Mode support, Range for trusted key configuration	✓
Object Tracking: IPv6 Route Tracking	✓
Onboard Failure Logging	✓
Open Authentication + MAB	✓
openconfig-LLDP	✓
Openflow1.3 Multi table	✓
OSPF	✓
OSPF Area Transit Capability	✓
OSPF Flooding Reduction	✓
OSPF Forwarding Address Suppression in Translated Type-5 LSAs	✓
OSPF Inbound Filtering using Route Maps with a Distribute List	✓

Fuente: Obtenido de https://cfnng.cisco.com/compare

De igual forma, se consulta por la compatibilidad de los equipos que forman el core de la red, los equipos Cisco Catalyst 9407R pertenecientes a la familia 9400 y con un sistema operativo específico 17.3.4. En la **Figura 26**, se puede apreciar que el igual que el modelo anterior el protocolo OpenFlow si se encuentra presente entre los protocolos admitidos por el equipo en esa versión de sistema operativo.

Figura 26.

Protocolos compatibles con el switch Cisco Catalyst 9400

All Features Search	CAT9400 IOS XE - 17.3.4
NTPv4 Orphan Mode support, Range for trusted key configuration	✓
Object Tracking: IPv6 Route Tracking	✓
Onboard Failure Logging	✓
openconfig-LLDP	✓
Openflow1.3 Multi table	✓
OSPF	✓
OSPF Area Transit Capability	✓
OSPF Flooding Reduction	✓
OSPF Forwarding Address Suppression in Translated Type-5 LSAs	✓
OSPF Inbound Filtering using Route Maps with a Distribute List	✓
OSPF Limit on Number of Redistributed Routes	✓
OSPF Link State Database Overload Protection	✓
OSPF Link-local Signaling (LLS) Per Interface Basis	✓
OSPF MIB Support of RFC 1850 and Latest Extensions	✓

Fuente: Obtenido de: https://cfnng.cisco.com/compare

Dentro de la red de distribución se cuenta con equipos de la serie Catalyst 9300 que como se aprecia en la ***Figura 27.*** *Protocolos compatibles con el switch Cisco Catalyst 9200*, si cuentan con el soporte para OpenFlow. Sin embargo, al realizar el mismo análisis para los equipos C9200L con el sistema operativo IOS XE 17.3.4 carecen de este soporte como se puede apreciar en la Figura 27 que se presenta a continuación. Esto implica que con la versión establecida del sistema operativo seleccionado no se pueden integrar a este tipo de soluciones, y en base a consultas realizadas en la herramienta Cisco Feature Navigator dispobinle en : https://cfnng.cisco.com/compare, se puede establecer que, la última versión disponible de sistemas operativos para este equipo sigue sin tener compatibilidad

con ninguna versión de OpenFlow, imponiendo un limitante para la inclusión de este equipo dentro de una arquitectura de red definida por software.

Figura 27.

Protocolos compatibles con el switch Cisco Catalyst 9200

All Features Search	CAT9200L IOS XE - 17.3.4
Object Tracking: IPv6 Route Tracking	✓
Onboard Failure Logging	✓
Open Authentication + MAB	×
openconfig-LLDP	×
Openflow1.3 Multi table	×
OSPF	✓
OSPF Area Transit Capability	✓
OSPF Flooding Reduction	✓
OSPF Forwarding Address Suppression in Translated Type-5 LSAs	✓
OSPF Inbound Filtering using Route Maps with a Distribute List	✓
OSPF Limit on Number of Redistributed Routes	✓
OSPF Link State Database Overload Protection	✓
OSPF Link-local Signaling (LLS) Per Interface Basis	✓
OSPF MIB Support of RFC 1850 and Latest Extensions	✓

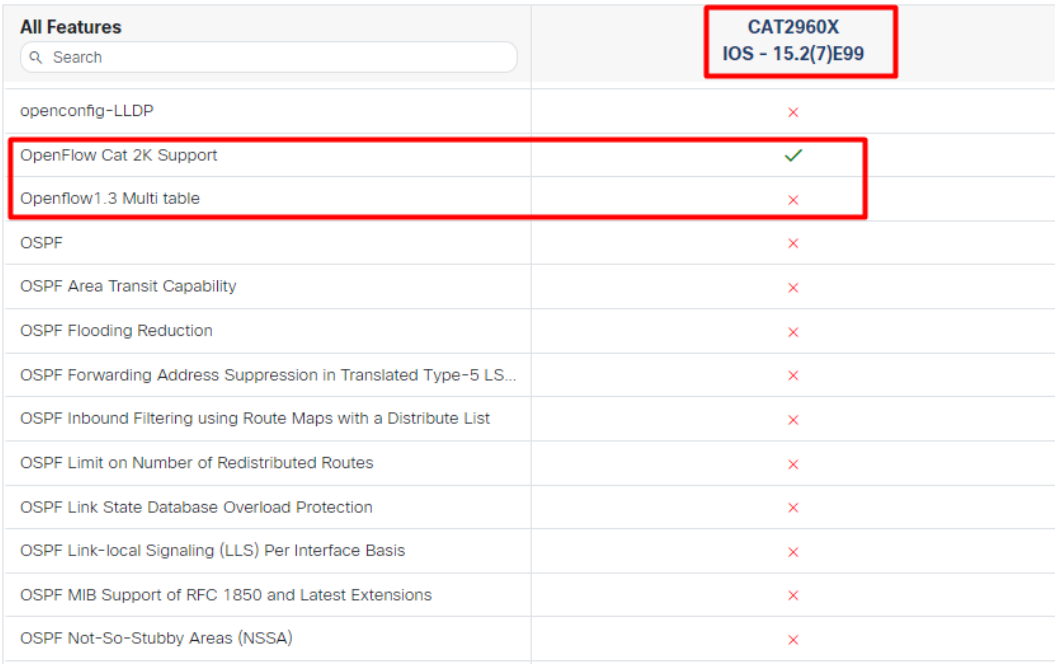
Fuente: Obtenido de: https://cfnng.cisco.com/compare

Según (Cisco, 2016) los requisitos para la compatibilidad de OpenFlow con los dispositivos de red se relacionan directamente con la capacidad del hardware para soportar nuevas versiones del sistema operativo. En el caso de la serie Cisco 2960X/XR, el último lanzamiento admite su integración en entornos SDN gracias al soporte de Cisco OpenFlow Cat 2K Support incluido en los dispositivos con el sistema operativo más reciente. La

Figura **28** presenta la esta opción marcada como compatible para los dispositivos de esta serie.

Figura 28.

Protocolos compatibles con el switch Cisco Catalyst 2960X



All Features Search	CAT2960X IOS - 15.2(7)E99
openconfig-LLDP	×
OpenFlow Cat 2K Support	✓
Openflow1.3 Multi table	×
OSPF	×
OSPF Area Transit Capability	×
OSPF Flooding Reduction	×
OSPF Forwarding Address Suppression in Translated Type-5 LS...	×
OSPF Inbound Filtering using Route Maps with a Distribute List	×
OSPF Limit on Number of Redistributed Routes	×
OSPF Link State Database Overload Protection	×
OSPF Link-local Signaling (LLS) Per Interface Basis	×
OSPF MIB Support of RFC 1850 and Latest Extensions	×
OSPF Not-So-Stubby Areas (NSSA)	×

Fuente: Obtenido de: https://cfnng.cisco.com/compare

4.2 Selección de solución SDN

El proceso de selección de una solución de redes definidas por software para la migración de un centro de datos debe considerar las características que mejor se acoplen a la demanda y al estado actual de la red de datos. Considerando que, se deben evaluar distintos parámetros se considera hacer uso del estándar de calidad de software ISO 25010, con la finalidad de determinar las características más importantes a considerar para evaluar la solución óptima para este caso.

El estándar ISO 25010 considera nueve características de calidad primordiales para evaluar los productos de software, entre estas se considera: idoneidad funcional, fiabilidad, eficiencia de desempeño, usabilidad, seguridad, protección, compatibilidad, mantenibilidad y portabilidad. Estas características pueden

subdividirse en más aspectos a considerar acorde al tipo de producto de software como tal (ISO/IEC, 2023).

En base a las características principales, es posible derivar sub características que se relacionan con cada una de estas, y que son las que establecen que tan bien cumple el producto de software con cada una de estas. Para determinar las características que se seleccionarán es necesario definir cada una de estas con sus respectivas sub características:

- **Idoneidad funcional:** hace referencia a la capacidad del producto de proporcionar las funciones implícitas del usuario y considera como sub características a la integridad funcional que responde si el producto cumple con las tareas y objetivos establecidos, la corrección funcional que indica si se obtienen los resultados correctos y a la adecuación funcional que responde a que tan bien las funciones cumplen con tareas y objetivos propios del producto de software (Ormeño Rojas, 2019).
- **Fiabilidad:** indica si el sistema o producto de software tiene la capacidad de cumplir con sus funciones en entornos y periodos de tiempo de terminados sin interrupciones. Para ello considera cuatro sub características que son: ausencia de fallos que como su nombre lo indica se refiere a la capacidad de cumplir funciones sin fallos en condiciones normales, disponibilidad que hace referencia a la capacidad del sistema de mantenerse operativo y accesible, tolerancia a fallos que comprende la capacidad del sistema de operar en presencia de situaciones de fallo, y finalmente, la capacidad de recuperación que es la capacidad del producto de software de recuperar su estado de operabilidad y los datos afectados tras un evento de interrupción o fallo (Ormeño Rojas, 2019).

- **Eficiencia de desempeño:** hace referencia al desempeño del producto o sistema en relación con los recursos asignados para el mismo. Se consideran 3 sub características que permiten evaluar al sistema o producto que son: comportamiento temporal que hace referencia a los tiempos de respuesta o procesamiento del sistema mientras se están haciendo uso de sus funciones determinadas, utilización de recursos hace referencia a la cantidad y tipo de recursos que se emplea para cumplir con las funciones determinadas en el entorno habitual para el que se ha diseñado, y la capacidad que indica el límite o el máximo de uno de sus parámetros (Portal ISO 25000, n.d.).
- **Compatibilidad:** es la capacidad de un producto de interactuar con otros productos o sistemas sin interferir en sus funciones al compartir el mismo escenario o recurso. Para esto define dos sub características que con: la coexistencia es decir que puede compartir entorno y tiempo de funcionalidad con otro sistema de software independiente sin disminuir su rendimiento, y el segundo es la interoperabilidad que hace referencia a la capacidad de uno o varios sistemas de intercomunicarse e intercambiar información útil para su funcionamiento cotidiano (Portal ISO 25000, n.d.).
- **Usabilidad:** conocida también como la capacidad de interacción hace referencia justamente a la capacidad para que el usuario interactúe directamente con el sistema, por lo que comprende varias sub características a considerar que son: reconocimiento de la adecuación que refiere a la capacidad del producto de dar a entender al usuario si el software se acopla a sus necesidades, curva de aprendizaje que indica

que tan fácilmente el usuario puede acoplarse y aprender del funcionamiento del software o producto, operabilidad indica la capacidad del producto de ser operado y controlado por el usuario final, protección contra errores como su nombre o indica es la capacidad del sistema de prevenir errores de operación generados por el usuario, inclusividad indica la capacidad del producto de ser usado por usuarios de distintas características, auto – descriptividad hace referencia a la información que presenta el software para definir sus funciones dentro de la aplicación de usuario ya sea mediante manuales, cuadros de diálogo, soporte con otros usuarios o desarrolladores, entre otros (Ormeño Rojas, 2019).

- **Seguridad:** se encarga de revisar la protección de los datos y de la información que maneja el sistema o producto de software a evaluar, así como también medir la capacidad de respuesta ante incidencias de seguridad. Las sub características se centran en dos elementos del triángulo CIA [9] , por lo que se tiene: confidencialidad que es la capacidad del sistema de mantener los datos confidenciales y solo al alcance de usuarios autorizados, integridad que refiere a la capacidad de mantener la información sin cambios o alteraciones indeseadas, no repudio que refiere al registro de acciones o eventos junto con el usuario responsable de cada uno de ellos para evitar conflictos de seguridad, responsabilidad ligado directamente con el criterio anterior hace referencia al rastreo de acciones y eventos, resistencia que hace

[9] Triángulo CIA: llamado también como triada CIA, hace referencia a la confidencialidad, integridad y disponibilidad como un modelo establecido para el diseño de políticas de seguridad dentro de organizaciones centrándose en la seguridad de la información y servicios informáticos (Hashemi-Pou, 2023).

referencia a la capacidad de mantener el funcionamiento del sistema o producto en condiciones de ataques o vulneraciones al sistema (Ormeño Rojas, 2019).

- **Protección:** relacionada con la característica de seguridad e integrada a la misma en ocasiones , pero enfocada más a la protección del recurso humano o del usuario final, pues considera sub características como: restricción operativa que son los límites de funcionamiento establecidos para evitar daños materiales, identificación de riesgos que es la capacidad de identificar situaciones que pueden ocasionar daños al recurso humano de forma directa o indirecta, también se incluyen daos a la propiedad del organismo o daños al ambiente (Portal ISO 25000, n.d.).
- **Mantenibilidad:** se relaciona con el grado de efectividad y eficiencia en el que el producto pueda modificarse, corregirse o adaptarse a los cambios en el entorno. Se basa en las sub características como: modularidad que implica el nivel en que el sistema o producto se conforma de componentes discretos, reutilizabilidad que implica que el sistema pueda acoplarse a funcionar como parte de otros sistemas, modificabilidad hace referencia a la capacidad de realizar modificaciones de forma efectiva y eficiente sin disminuir la calidad del producto (Portal ISO 25000, n.d.).
- **Portabilidad:** relacionado directamente con la flexibilidad y conforma el grado en el que el producto o sistema se adapta a cambios dentro de los distintos contexto de uso que se le puedan presentar. Se fundamente en las siguientes su características: adaptabilidad que consiste en la

capacidad del sistema de adaptarse de forma eficiente a recursos de hardware o entornos operativos de mayor relevancia, la escalabilidad hace referencia a la facilidad que tiene el sistema para integrar nuevos elementos o manear cargas de trabajo cambiantes, reemplazabilidad hace referencia al grado en que un producto puede legara reemplazar a otro producto de software, y la instalabilidad que indica la eficiencia con la que el sistema se puede instalar o desinstalar en un entorno determinado (Portal ISO 25000, n.d.).

Para la selección de una solución de red definida por software es necesario evaluar diferentes alternativas que puedan acoplarse con la realidad de la red existente, por lo que se consideran tres alternativas. En base a los tipos de soluciones planteadas en el apartado 3.4 Definición de soluciones SDN, se considera la aplicación de soluciones propietarias y no propietarias, las cuales son:

- **Solución Cisco Meraki :** es una solución propietaria pensada para administración en la nube en entornos de centros de datos con recursos más limitados, no maneja controladores locales y ofrece una larga lista de dispositivos que se anexan a la gestión centralizada, ofrece altos niveles de seguridad y se acopla con la gestión y configuración propia de Cisco. Se considera esta solución como una propuesta de análisis debido a la garantía de cisco al ofertar seguridad y rendimiento en sus dispositivos.
- **Solución Cisco ACI:** solución de tipo propietaria diseñada para ofrecer automatización y mayor simplicidad a las operaciones de una red garantizando el rendimiento y la escalabilidad mediante un sistema de control centralizado basado en el uso de políticas que

facilitan la resolución de problemas volviéndolo altamente recomendado para redes concurridas.

- **Solución basada en OpenFlow:** consta de una solución no propietaria basada en el protocolo OpenFlow para integrar SDN a una red, se considera el uso de un controlador centralizado que permita administrar todos los dispositivos vinculados, esta solución considera una alta compatibilidad y la escalabilidad que integra a la arquitectura de red.

Como parte del proceso de selección, se considera que, en base a las características de análisis propuestas en la normativa ISO 25010, se realice una comparación de las soluciones existentes para de este modo determinar la que mejor se acople a la situación actual de la red del data center de la UTN. Con esta finalidad, se ha considerado la revisión de documentación de apoyo provista por los desarrolladores de dichas soluciones para tener un mejor panorama de las características y puntos destacables en cada una. Con la finalidad de representar de mejor manera estas características, se considera realizar varias tablas en las que se puntúa en la escala de 1 a 10 los distintos criterios asociados a las características descritas en la normativa 25010.

En primer lugar, se procede a evaluar la idoneidad funcional, por su definición esta característica hace referencia al cumplimiento de funciones que pueda tener la solución de red definida por software que se ha propuesto, por lo que, en la Tabla 19 se evalúan distintos criterios que permiten comparar los resultados de cada solución.

Tabla 19.

Evaluación de idoneidad funcional

Criterio de evaluación de idoneidad funcional	Cisco Meraki	Cisco ACI	Solución basada en OpenFlow
Cumplimiento de requisitos	7	8	9
Flexibilidad de configuración	6	8	9
Escalabilidad funcional	6	9	9
Total	19	25	27

La evaluación realizada de demuestra que, la solución basada en OpenFlow puede aportar un mejor desempeño en cuanto a la funcionalidad de la solución, ya que, sobre todo en términos de flexibilidad y escalabilidad es superior a las soluciones propietarias ya que da mayor libertad al operador de red para gestionar los recursos de la red. Además, se considera que, una solución de código abierto tiene una mayor tendencia a acoplarse a los requisitos y demandas de un entorno ya existente, mientras que en este caso las dos soluciones de cisco tiene como principales limitantes criterios de licenciamiento que establecen ya configuraciones y capacidades máximas para la gestión de la red, y en caso de que estas no sean suficientes a futuro se debe pensar una reestructuración completa de la red que haga que esta vuelva a acoplarse con la solución estableciendo una relación en la que la red se adapte a la solución y no viceversa como se espera se opere en este tipo de entornos.

La siguiente característica es la eficiencia de desempeño, un factor crucial ya que principal objetivo de la migración a una red definida por software es mejorar el desempeño de la red, por lo cual evaluar el rendimiento, el consumo de

recursos y otros factores mencionados en la Tabla 20 favorecerán a tener una mejor perspectiva del desempeño de las soluciones que se están evaluando.

Tabla 20.

Evaluación de eficiencia de desempeño

Criterio de evaluación de eficiencia de desempeño	**Cisco Meraki**	**Cisco ACI**	**Solución basada en OpenFlow**
Rendimiento general	7	9	9
Latencia	7	9	9
Uso de recursos	6	8	9
Capacidad de procesamiento	7	9	9
Total	27	35	36

En términos de eficiencia, se considera que, las soluciones cisco difieren mucho entre sí, esto por el enfoque específico plateado para cada una, más se sostiene que la solución basada en código abierto tiene un mejor puntuaje debido a que, gracias a la integración de configuraciones propias y de integración con políticas y operaciones el rendimiento general, la latencia y el consumo de recursos es mucho mejor. Además, se considera que, el procesamiento implicado en el uso de APIs desarrolladas en código abierto el mucho más ligero y menos demandante que el de las APIs de soluciones que ofrece cisco que requieren de más recursos para integrarse con el entorno de red en cuestión. La Tabla 20 indica

puntuaciones asignadas a criterio en base a una calificación perceptiva[10] construida a partir de información obtenida de sitios de opinión relevantes como es el caso de (Gartner Peer Insights, n.d.).

Otra característica de suma relevancia es la compatibilidad y al mismo tiempo la portabilidad , ya que se busca que la solución se acople lo mejor posible a la infraestructura de red existente, ya que considerando que, los dispositivos de red existentes se encuentran operativos lo ideal es poder trabajar con la mayor cantidad de estos para de este modo disminuir el impacto financiero que conlleva levantar una nueva red con todos sus componentes desde cero, por lo que en la Tabla 21 se considera la evaluación de este parámetro.

Tabla 21.

Evaluación de compatibilidad y portabilidad

Criterio de evaluación de compatibilidad y portabilidad	Cisco Meraki	Cisco ACI	Solución basada en OpenFlow
Interoperabilidad con otros sistemas	7	8	10
Estándares y protocolos soportados	8	9	10
Integración con sistemas de terceros	7	8	10
Soporte para múltiples plataformas	7	8	9
Compatibilidad con hardware existente	7	8	9

[10] Calificación perceptiva: Sistema de calificación basado en opiniones o percepciones de individuos o comunidades que se emplean en evaluaciones de desempeño, revisiones de productos entre otras (Operé, 2016).

Facilidad de implementación en la red actual	7	8	8
Facilidad de migración	7	8	9
Total	50	57	65

Considerando los criterios especificados, se puede determinar que el grado de compatibilidad que se pueda tener con los dispositivos de red son de crucial importancia. Dentro del análisis se ha podido determinar que, para una solución basada en OpenFlow la compatibilidad con recursos de software y hardware es mucho mayor a la que se alcanza con soluciones propietarias, pues para el caso de Meraki y ACI las versiones específicas del hardware se vinculan directamente con la solución, es decir los dispositivos recomendados para integrarse deben pertenecer a la línea completa dela solución, este factor hace que la compatibilidad con el hardware existente sea mucho menor que en el caso de la solución de código abierto, este criterio también ocasiona que, la facilidad de implementación y de la migración se consideren con valores más altos, estos valores se asignan siguiendo el mismo esquema de puntuación en el que, tomando como partida opiniones de sitios como (TrustRadius, n.d.) es posible determinar de mejor forma los puntos fuertes de soluciones SDN que se encuentran en el mercado. En base a este análisis, el tema de la portabilidad también define a la solución de código abierto basada en OpenFlow como la mejor alternativa para trabajar en una red que ya existe y que, al contar con los componentes de hardware, se debe optar por la solución que permita hacer uso de la mayor cantidad de recursos existentes posibles.

Dentro de un datacenter, otro factor crucial es la fiabilidad, pues se requiere que la red cumpla con sus funciones con la menor cantidad de interrupciones y fallos posibles, por lo que, en la tabla 22 se muestran los criterios de evaluación a considerar para determinar la solución que mejor cumpla con esta característica.

Tabla 22.

Evaluación de fiabilidad

Criterio de evaluación de fiabilidad	**Cisco Meraki**	**Cisco ACI**	**Solución basada en OpenFlow**
Tolerancia a fallos	8	9	9
Disponibilidad	8	8	8
Robustez	9	9	9
Recuperabilidad	8	9	8
Total	33	35	34

En términos de fiabilidad, los resultados obtenidos permiten determinar, que la fiabilidad de las diferentes soluciones es similar, siendo la solución Cisco ACI la que mejores resultados obtuvo, esto debido a al integración que realiza con los equipos y a la robustes que puede ofertar en entornos de alta demanda, además debido a las integraciones ACI con la nube es posible contar con una mayor tolerancia a fallos garantizando así la disponibilidad y la recuperabilidad de servicios y sistemas ante posibles fallos, en comparación a las soluciones de código abierto es superior debido a que, la capacidad de recuperación y la tolerancia a fallos se ligan directamente con el desempeño de la configuración que

se realice dentro del controlador y de como de defina la red propiamente, en cambio la solución ACI cuenta con respaldo propio del proveedor lo cual la cataloga como la mejor en este campo mencionando sus beneficios en (Cisco, 2024b) y ganando mayor credibilidad gracias a opinpiones expuestas en (Gartner Peer Insights, n.d.) siendo estos factores de suma relevancia en la asignación de puntajes dentro de la Tabla 22.

Con la intención de garantizar la seguridad de la red, se deben evaluar distintos criterios dentro de las soluciones planteadas, estos se centran en la protección de la información y en la privacidad como tal que se oferta con estas a la red en cuestión. Se debe recalcar que se pueden considerar aún más criterios de análisis, pero de forma más enfocada a una solución de seguridad avanzada, por lo que en la Tabla 23 se considera únicamente una implementación básica de seguridad de cada una de las soluciones propuestas

Tabla 23.

Evaluación de seguridad

Criterio de evaluación de seguridad	**Cisco Meraki**	**Cisco ACI**	**Solución basada en OpenFlow**
Autenticación y autorización	9	9	9
Encriptación	8	9	9
Protección contra amenazas	8	9	9
Gestión de vulnerabilidades	8	9	10
Total	33	36	37

Al igual que en apartados anteriores, se toma como punto de partida elementos de la documentación de cada solución, así como los comentarios relevantes que refieran a los criterios evaluados para de este modo determinar con lo expuesto en la Tabla 23 la solución que cuente con mejores prestaciones de seguridad, misma que en base a la calificación perceptiva aplicada señala a la solución basada en OpenFlow como la mejor en el ámbito evaluado. Este resultado se ve influenciado principalmente por la apertura que tienen las soluciones código abierto en la implementación de políticas y medidas de seguridad que se moldeen a las necesidades de la red incluyendo altos niveles de protección contra amenazas y la gestión de vulnerabilidades.

Finalmente, considerando la usabilidad y la mantenibilidad como características de integración con el usuario y el entorno, se evalúan dentro de la Tabla 24 criterios específicos orientados a determinar la solución que tenga mejores prestaciones para los dos componentes. Se considera sobre todo que, para el factor humano, la solución sea comprensible y que pueda perdurar en el entorno por un periodo de tiempo largo considerando que deba someterse a cambios para lograr este objetivo.

Tabla 24.

Evaluación de mantenibilidad y usabilidad

Criterio de evaluación de mantenibilidad y usabilidad	Cisco Meraki	Cisco ACI	Solución basada en OpenFlow
Facilidad de mantenimiento	10	8	10
Modularidad	5	9	9

Facilidad para aplicar actualizaciones	9	9	8
Curva de aprendizaje	9	7	8
Disponibilidad de documentación	8	8	10
Total	41	41	45

En este último aspecto se recalca sobre todo la capacidad de la solución de mantenerse dentro de esquema que se está diseñando, para ello se tiene siempre en consideración que, la solución de código abierto puede contar con mayores facilidades al cotar con cierta independencia, ya que, dentro de soluciones propietarias, en muchos casos se debe considerar, que el mantenimiento está ligado a niveles de soporte otorgados por el proveedor, de igual forma, se encuentra la limitación de que la documentación no está liberada por completo como es el caso de las soluciones de código abierto por lo que esta impone una mayor ventaja en la calificación que se ha realizado a cada una de las soluciones planteadas.

Con la evaluación de cada característica y sus respectivos criterios se procede a realizar una valoración final que resuma los valores asignados para cada solución, esto con la finalidad de determinar de forma numérica una solución óptima para el plan de migración. Los valores obtenidos se representan dentro de la Tabla 25, estos son una sumatoria de los puntajes obtenidos en cada uno de los criterios desplegados con anterioridad, los valores finales de cada solución implican que, con 244 puntos, la solución basada en OpenFlow cuenta con mejores resultados en base a la calificación perceptiva realizada, se considera a la

vez que esta se realizó tomando como referencia la arquitectura actual y la disponibilidad de recursos para la nueva solución que se plantea.

Tabla 25.

Tabla de evaluación final de los criterios

Criterio	Cisco Meraki	Cisco ACI	Solución basada en OpenFlow
Adecuación funcional	19	25	27
Eficiencia de desempeño	27	35	36
Compatibilidad y portabilidad	50	57	65
Fiabilidad	33	35	34
Seguridad	33	36	37
Mantenibilidad y usabilidad	41	41	45
Total	203	229	244

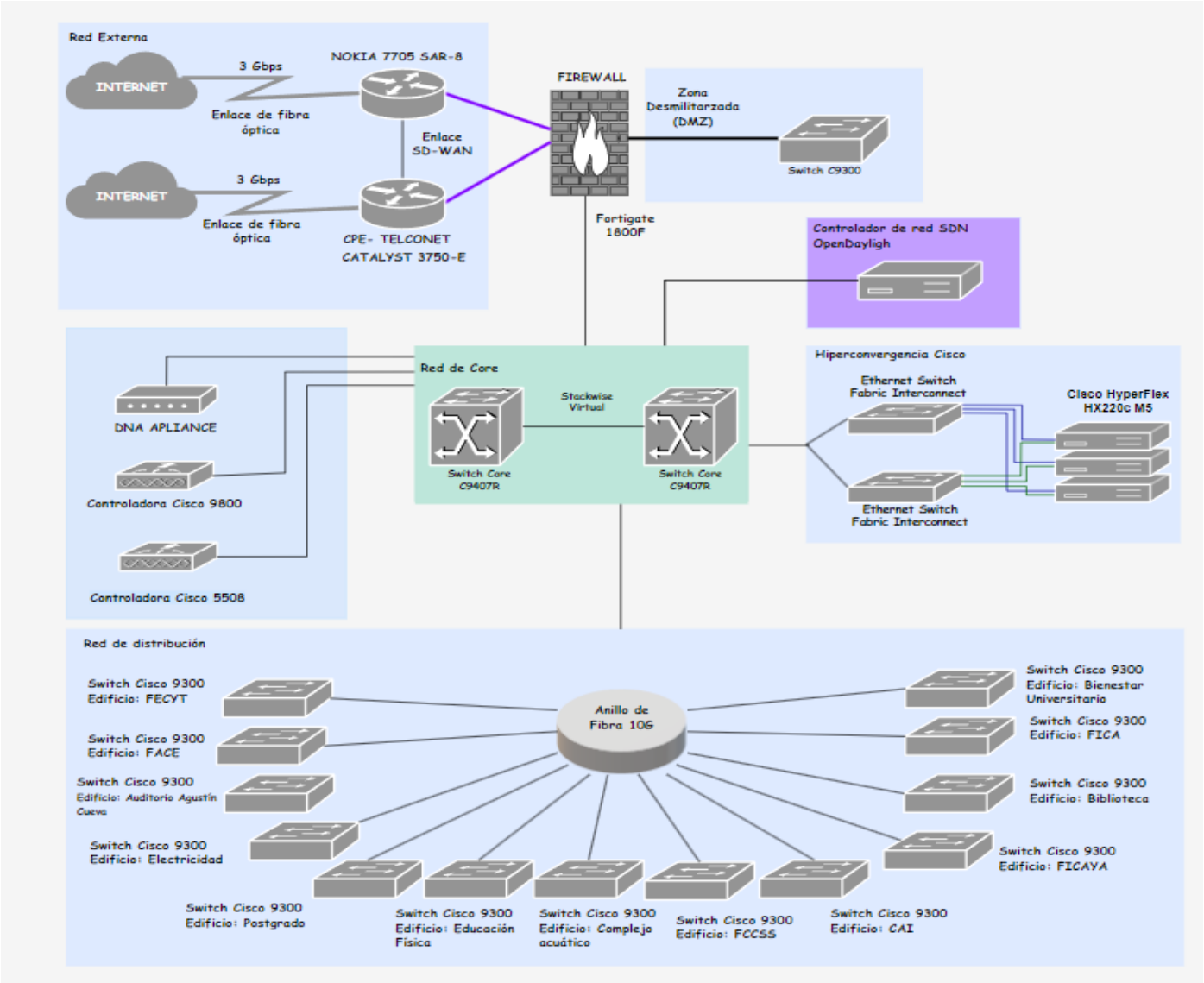
4.2.1 Descripción de la solución seleccionada

En base al análisis realizado anteriormente, se propone la implementación de una red definida por software utilizando un controlador OpenDaylight con soporte para OpenFlow 1.3. Se plantea esta arquitectura para llevar una gestión centralizada y dinámica de la red, mejorando la eficiencia y flexibilidad en la administración de recursos. La topología de la red se mantiene en gran medida igual a la infraestructura existente, con algunos ajustes clave para integrar el controlador SDN. Los switches Cisco Catalyst 9300, configurados como conmutadores SDN, se utilizarán en la red de distribución para garantizar la compatibilidad con OpenFlow. El controlador SDN estará instalado en un servidor

dedicado, conectado directamente a la red de core a través de los switches compatibles con OpenFlow. Esto permite que el controlador gestione y controle el tráfico de red, aplicando políticas y reglas de manera dinámica. La Figura 29 presenta la arquitectura base para la implementación de la solución, con esta se define una nueva conexión para el controlador desde el core de la red con la finalidad de agregar a través de este los equipos presentes en la red de distribución de la arquitectura.

Figura 29.

Topología de red propuesta para la solución seleccionada



La selección del controlador de código abierto se realizó, en base a la información obtenida de (Cisco, 2016) y expuesta en la Tabla 18, donde se especifican los controladores compatibles con los dispositivos de las series de Cisco Catalyst. Se seleccionó OpenDaylight gracias al amplio soporte y adopción en la industria, así como su capacidad de integración con diversos dispositivos de red a través del protocolo OpenFlow 1.3 que permite una administración dinámica y eficiente. OpenDayligth favorece a la automatización de la red mediante la separación del plano de control y el plano de datos. Funciona sobre una arquitectura modular basada en OSGi, lo que permite añadir o quitar módulos según las necesidades de la red, incrementando la flexibilidad y personalización (Eftimie & Borcoci, 2020).

Para garantizar el rendimiento óptimo del controlador, se recomienda su instalación en un servidor dedicado con un procesador de alto rendimiento y al menos 16 GB de RAM. Este servidor se conectará directamente a la red de núcleo mediante switches compatibles con OpenFlow, como los Cisco Catalyst 9300 configurados con OpenFlow. Esta configuración permitirá que el controlador gestione el tráfico de red y aplique políticas en tiempo real, mejorando la eficiencia operativa y optimizando la distribución del tráfico (Eftimie & Borcoci, 2020).

Además, para fortalecer la seguridad de la red, se sugiere realizar considerar la futura integración del Fortinet FortiGate Connector para realizar un nexo con la SDN. Este conector facilita la comunicación de políticas de seguridad desde el controlador OpenDaylight al firewall Fortinet FortiGate, logrando así tener un sincronismo y un buen manejo de la seguridad dentro de la red. Aunque en esta fase metodológica no es posible llevar a cabo esta integración debido a la

falta de acceso a licencias específicas. Esta solución proporciona una visibilidad completa y un control granular sobre el tráfico de red, asegurando que las políticas de seguridad se apliquen de manera consistente en toda la infraestructura definida por software.

4.3 Guía metodológica

Para llevar a cabo el proceso de migración de un centro de datos es necesario contar con una guía que permita determinar de forma estructurada los pasos a seguir para realizar cambios dentro de la arquitectura de red. Para ello se deben considerar aspectos analizados anteriormente como la evaluación del estado actual, la evaluación de compatibilidad, entre otros. En este apartado se hace especial énfasis en solución a desarrollar, esta solución debe ser la que mejor se adepte al entorno actual y que permita hacer el mejor uso posible de todos los elementos que existen actualmente dentro de la red.

4.3.1. Descripción de recursos de hardware existentes

Con el estudio inicial de la estructura de red actual del data center de la Universidad Técnica del Norte ha sido posible reconocer a nivel de hardware cada uno de los dispositivos de red de interés para el proceso de migración hacia la tecnología de redes definidas por software. La revisión de la topología de red proporcionada por el DDTI permite establecer como punto de partida al core de la red, el cual se encuentra integrado por dos switches de la marca Cisco en su serie Catalyst 9400; estos se interconectan de tal forma que actúan como uno solo mediante la tecnología Virtual Stackwise, misma que permite usar de forma combinada los recursos de los equipos para darle mayor robustes al núcleo de la red. El núcleo de red a su vez se asocia directamente a la red de distribución, la cual interconecta todas las facultades del campus y que se

conforma de switches de la marca Cisco en diferentes modelos y versiones. Al evaluar el estado actual de la red es necesario determinar de manera puntual los dispositivos que formarán parte del plan de migración a la estructura de redes definidas por software, considerando las especificaciones de cada uno de ellos más allá de las generalidades de hardware que comparten muchos de estos dispositivos para de este modo establecer procesos clave dentro de la migración como actualizaciones y configuraciones.

Actualmente dentro del data center de la Universidad Técnica del Norte, se encuentran equipos que conforman una red tradicional para brindar conectividad a la institución. Los dispositivos considerados para la migración de esta infraestructura a una nueva arquitectura de redes definidas por software, involucran al core de la red y a la red de distribución, estas se conforman en su totalidad por switches Cisco en diferentes modelos. La **Tabla 26** resume las características necesarias de estos equipos para el proceso de migración a proponerse.

Tabla 26.

Listado de componentes de hardware dentro de la red actual de la UTN

ID Dispositivo	Modelo	Versión de Software	Ubicación física	Rol dentro de la red
CENTRAL.DC.CORE	Catalyst 9400 (C9407R)	17.3.4	Data Center	Core
SW.DMZ	Catalyst 9300 (C9300-48P)	17.3.4	Data Center	DMZ

AUDIAC.ACC	Catalyst 29xx (WS-C2960X-48TS-L)	15.2(7)E4	Auditorio Agustín Cueva	Acceso
FACAE.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	FACE	Distribución
CENTRAL.TERRAZA.DIS	Catalyst 3850 (WS-C3850-48T)	03.06.03E	Edificio Central	Distribución
FICAYA.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	FICAYA	Distribución
DBU.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Departamento de Bienestar Estudiantil	Distribución
CAI.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	CAI	Distribución
SW01.BIBLIO.DIS	Catalyst 29xx (WS-C2960X-48TS-L)	15.2(7)E5	Biblioteca	Distribución
SW02.BIBLIO.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Biblioteca	Distribución
FCCSS.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	FCCSS	Distribución
SW01.COMPAC.DIS	Catalyst 29xx (WS-C2960X-48TS-L)	15.0(2a)Ex5	Complejo Acuático	Distribución

SW02.COMPAC.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Complejo Acuático	Distribución
POSG.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Posgrado	Distribución
ELECT.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	Departamento de Electricidad	Distribución
FECYT.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	FECYT	Distribución
SW01.FICA.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	FICA	Distribución
SW02.FICA.DIS	Catalyst 4510R (WS-C4510R+E)	03.11.04E	FICA	Distribución
IEF.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	Polideportivo	Distribución
Firewall	Fortigate 1800F	FortigateOS	Data Center	WAN

Fuente: Dirección de Desarrollo Tecnológico e Informático de la UTN

4.3.2. Descripción de actividades a realizar

El proceso de migración de la red actual hacia una arquitectura definida por software presenta un plan estructurado que contempla actividades específicas orientadas a asegurar un desarrollo ordenado, eficiente y con el menor impacto posible en la continuidad del servicio dentro del data center de la Universidad Técnica del Norte. Las actividades detalladas en este apartado,

sustentadas en el análisis previo, incluyen la verificación de compatibilidad, la actualización de firmware, la adquisición de nuevos recursos de hardware, el diseño de la nueva arquitectura de red y la configuración secuencial de los dispositivos.

a) Verificación de compatibilidad.

Para evaluar la compatibilidad de los dispositivos de red existentes dentro de la red actual con la arquitectura de redes definidas por software, se examinaron las características específicas de cada equipo, enfocándose en el soporte para protocolos SDN como OpenFlow; según (De Peña Núñez, 2023) determinar la compatibilidad de los elementos que forman parte de la red es crucial para mantener el funcionamiento adecuado de cualquier solución SDN y que esta cuente con el soporte adecuado para dar continuidad al servicio que preste la infraestructura de datos. Al considerar que la mayoría de los equipos son de la marca Cisco, se opta por utilizar una herramienta proporcionada por el fabricante.

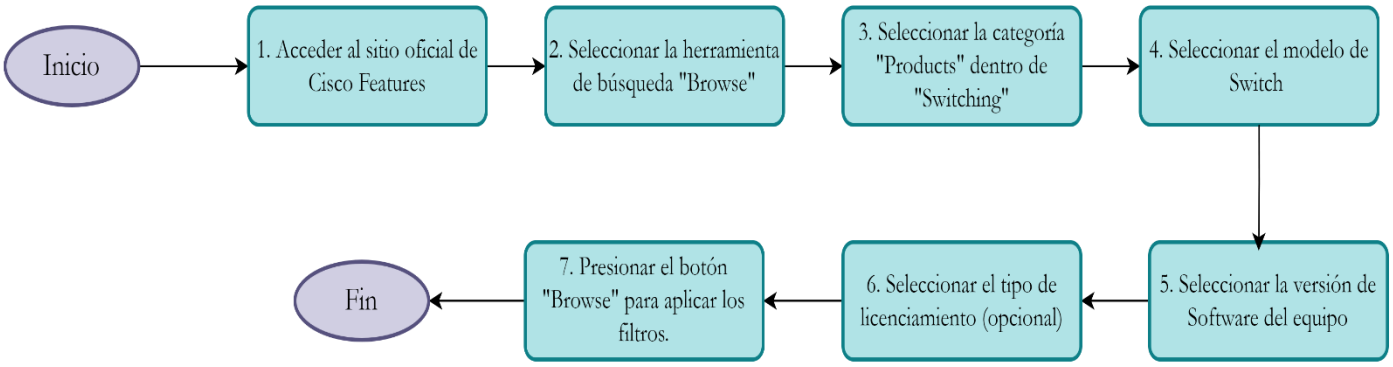
La herramienta seleccionada **Cisco Feature Navigator**, permite explorar en las características de cada uno de los equipos según su modelo, y versión del software ; esta herramienta se encuentra en el siguiente enlace: https://cfnng.cisco.com/ y ofrece la opción de ingresar con una cuenta de forma directa o realizar la consulta en modo de invitado.

La navegación dentro del sitio indicado se realiza se realizó a través de la opción **"Browse"** seleccionando la categoría de **"Switching"**, para buscar cada uno de los modelos indicados en la Tabla

26. Posteriormente se selecciona en las pestañas siguientes las especificaciones de versión de software y licenciamiento que es un campo opcional ya que realiza una recopilación general de todas las licencias existentes para los campos seleccionados anteriormente. La Figura 30 describe este procedimiento paso a paso.

Figura 30.

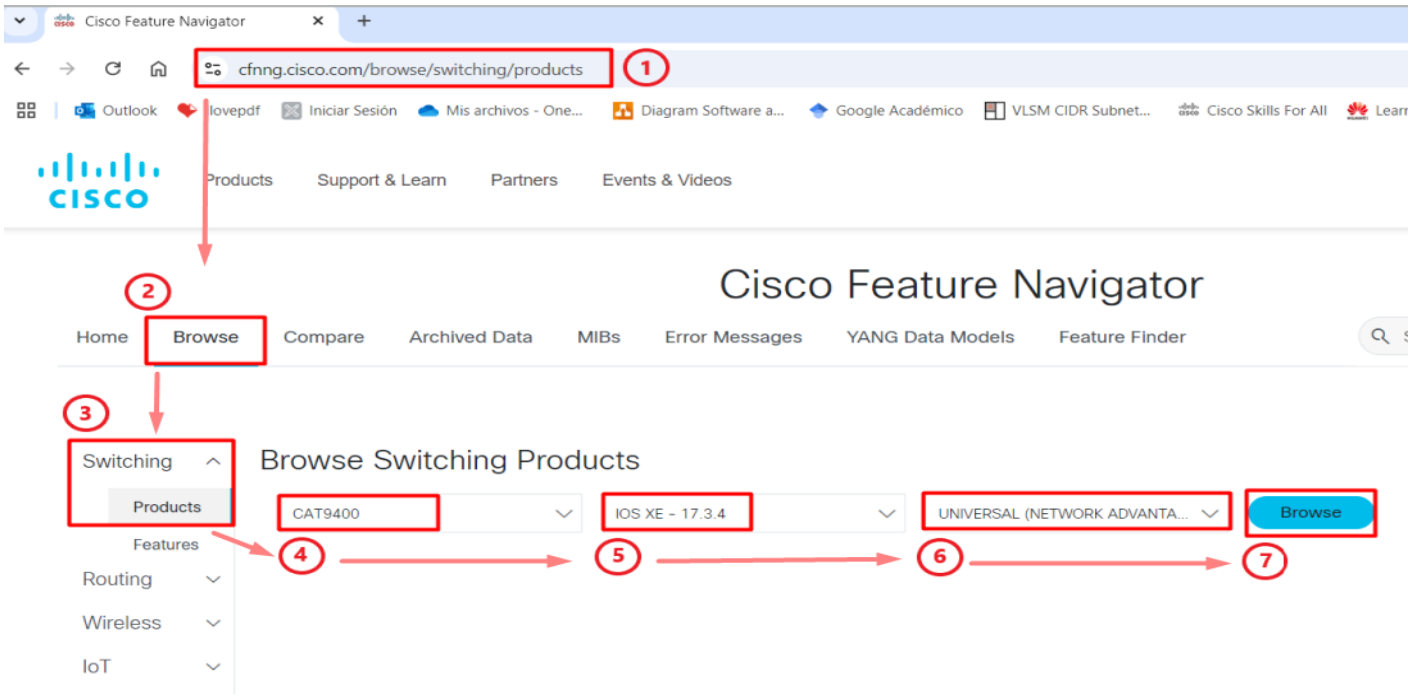
Procedimiento para la consulta de compatibilidad de equipos dentro de Cisco Feature Navigator



En *Figura 31* se puede observar los pasos explicados anteriormente, y se realiza la consulta para un switch Cisco Catalyst 9407R, que pertenece a la red de core de la topología actual.

Figura 31.

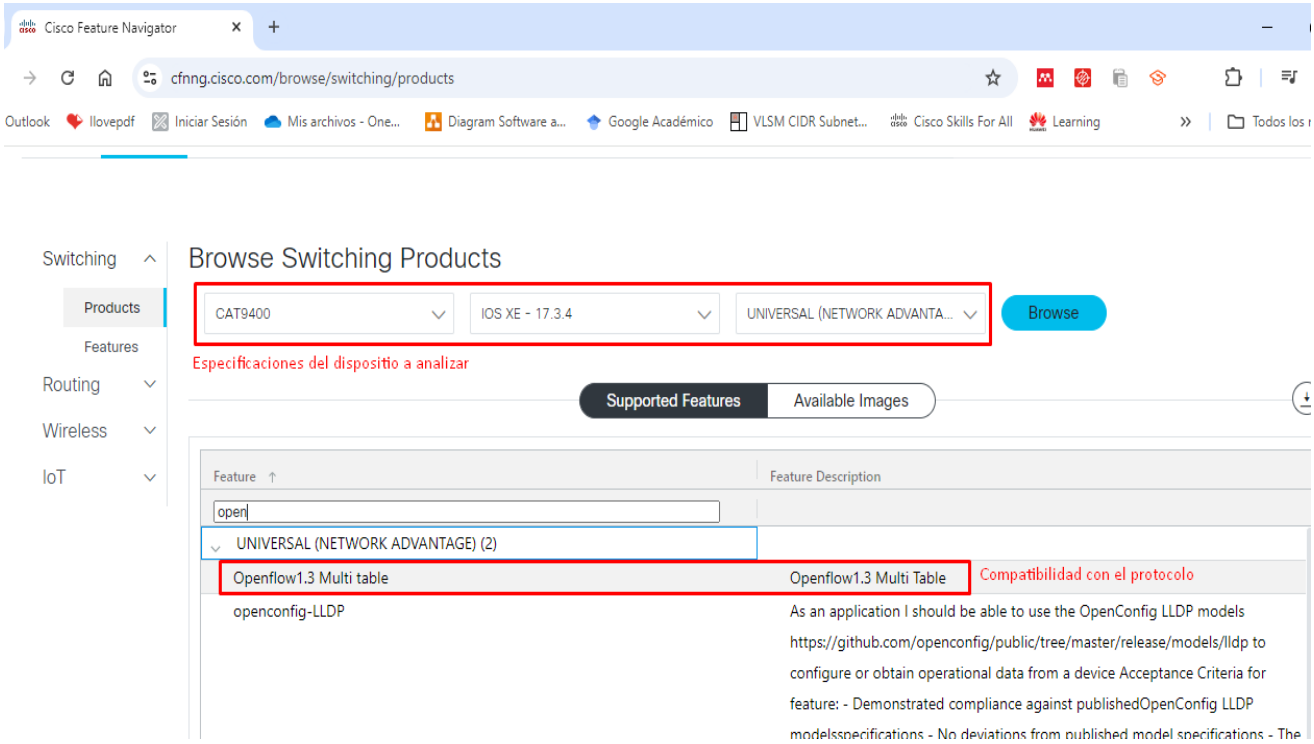
Ejemplo de consulta de compatibilidad de equipos Cisco dentro de Cisco Feature Navigator



Al aplicar los filtros de búsqueda, se enlistan todas las características disponibles para el dispositivo en ese modelo de sistema operativo, para ello se puede aplicar un nuevo filtro para verificar que el protocolo OpenFlow se encuentre en la lista se considera que el equipo o la versión de software no lo soportan. Tras realizar la consulta para el equipo mencionado, se confirma la compatibilidad con OpenFlow 1.3, como se muestra en la Figura 32.

Figura 32.

Resultado de la consulta de compatibilidad para el Switch de core Cisco Catalyst 9407R



Al realizar este proceso con cada uno de los equipos se puede determinar los dispositivos que admiten el protocolo OpenFlow, y que por ende pueden ser empleados en la nueva arquitectura de red definida por software, la **Tabla 27** muestra el resultado de esta evaluación para cada uno de los modelos descritos anteriormente.

Tabla 27.

Compatibilidad de dispositivos de red del data center con el protocolo OpenFlow.

ID Dispositivo	Modelo	Versión de Software	Compatibilidad con OpenFlow
CENTRAL.DC.CORE	Catalyst 9400 (C9407R)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
SW.DMZ	Catalyst 9300 (C9300-48P)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
AUDIAC.ACC	Catalyst 29xx (WS-C2960X-48TS-L)	15.2(7)E4	Compatibilidad limitada (OpenFlow Cat 2k Support)
FACAE.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
CENTRAL.TERRAZA.DIS	Catalyst 3850 (WS-C3850-48T)	03.06.03E	No compatible
FICAYA.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)

DBU.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
CAI.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	No compatible
SW01.BIBLIO.DIS	Catalyst 29xx (WS-C2960X-48TS-L)	15.2(7)E5	No compatible
SW02.BIBLIO.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
FCCSS.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
SW01.COMPAC.DIS	Catalyst 29xx (WS-C2960X-48TS-L)	15.0(2a)Ex5	No compatible
SW02.COMPAC.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
POSG.DIS	Catalyst 9300 (C9300-48P)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)

ELECT.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	No compatible
FECYT.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
SW01.FICA.DIS	Catalyst 9300 (C9300-48UXM)	17.3.4	Compatible (OpenFlow 1.3 Multi Table)
SW02.FICA.DIS	Catalyst 4510R (WS-C4510R+E)	03.11.04E	No compatible
IEF.DIS	Catalyst 9200L (C9200L-48P-4X)	17.3.4b	No compatible

b) Actualización de firmware en dispositivos de red.

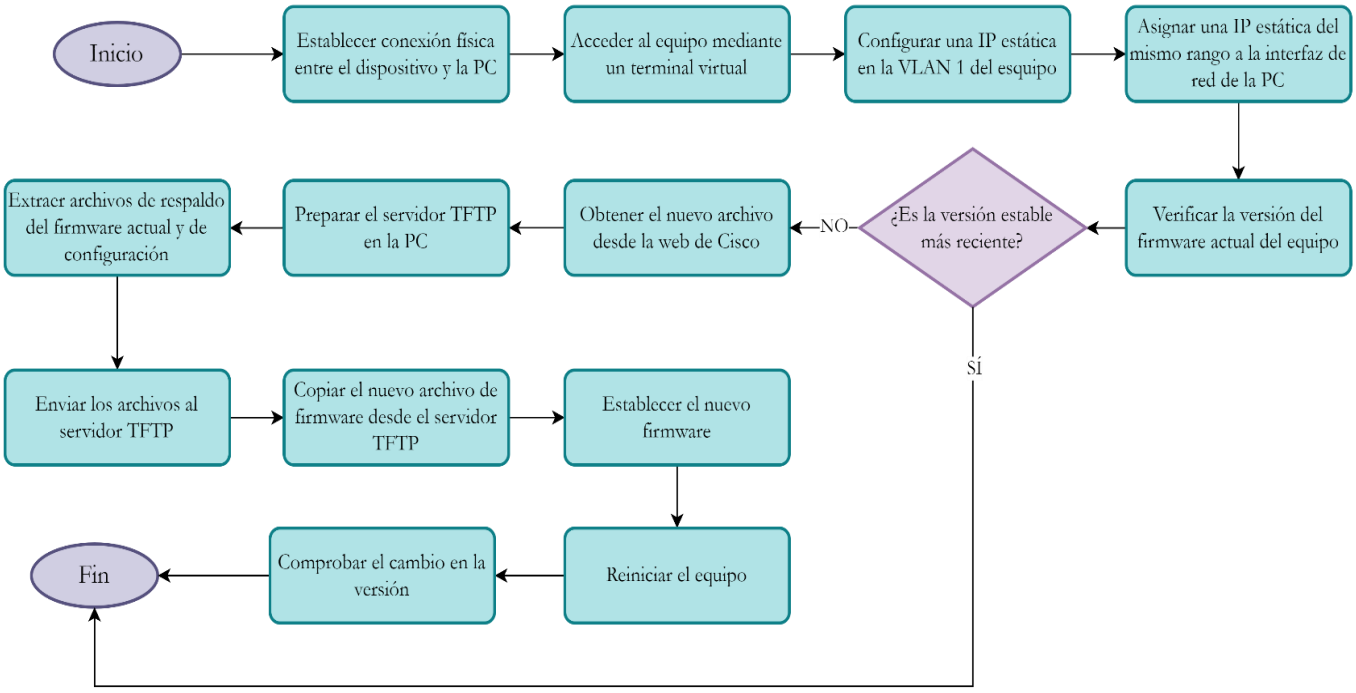
El firmware cumple un papel crucial al controlar las operaciones básicas de cualquier dispositivo actuando como intermediario entre el software y el hardware, razón por la cual el buen funcionamiento depende en gran medida de este. Los dispositivos de red cuentan con un firmware específico asociado a su sistema operativo, por lo que mantenerlo actualizado permite corregir errores, mejorar el rendimiento o integrara nuevas funcionalidades para estos. Dependiendo de cada fabricante el proceso de actualización de firmware puede variar, por lo que es necesario consultar con las guías establecidas por cada uno para llevar a cabo este proceso.

Las actualizaciones se deben realizar de forma constante acorde a las recomendaciones del fabricante del dispositivo de red, en el caso de Cisco, el propio fabricante ofrece un listado de las versiones recomendadas para cada uno de los modelos existentes, actualmente, según (Cisco, 2024c) los sistemas operativos cuentan con versiones nuevas compatibles con los equipos del data center, por lo que es posible realizar el proceso de actualización.

Existe documentación propia del fabricante que indica el proceso a seguir para realizar la actualización del firmware, para esto se debe considerar cuales son las versiones recomendadas y posteriormente, acceder al sitio oficial y realizar la búsqueda de la imagen correspondiente para poder iniciar el proceso de actualización, este proceso se muestra resumido en la Figura 33.

Figura 33.

Descripción del proceso de actualización de Firmware para un equipo Cisco



Tomando como referencia el proceso descrito en la Figura 33, para actualizar el firmware de los dispositivos Cisco, se pueden seguir los pasos descritos a continuación:

- **Establecimiento de conexión:**

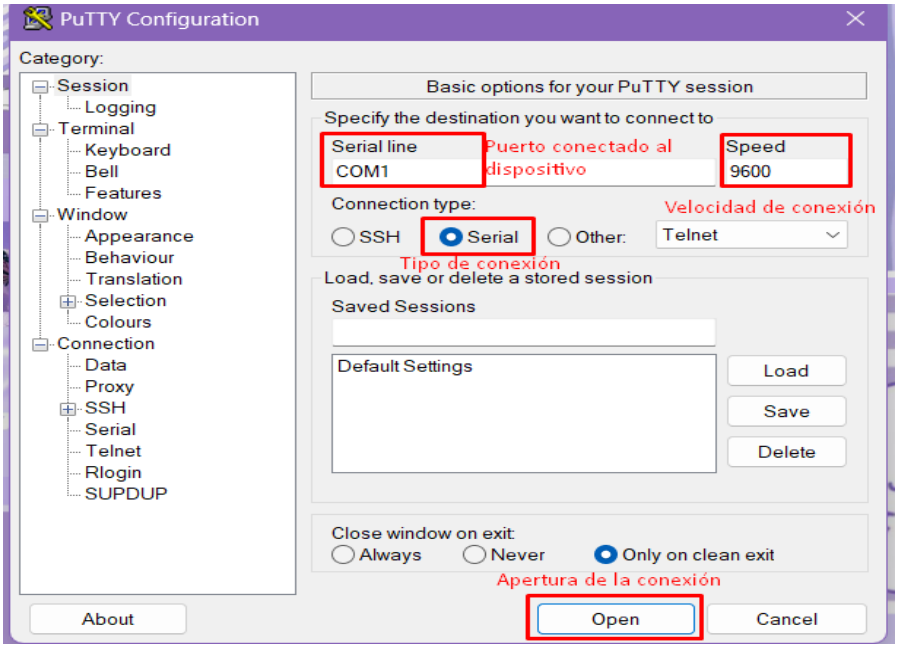
1. El primer paso implica establecer una conexión física entre el dispositivo de red y una PC haciendo uso de un cable de consola para conectarse directamente con la PC, el acceso al equipo se realiza mediante una consola serial que permita acceder a la interfaz de comandos del dispositivo. Es posible encontrar distintos softwares que se acoplan a esta necesidad, un ejemplo de ello es **PuTTY**, un emulador de terminal de licencia gratuita que cuenta con funcionalidades como: consola serial, cliente SSH, transferencia de archivos, entre otros. De forma previa a

realizar el proceso de actualización es posible obtener el software desde su sitio oficial: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html.

2. Una vez se ha realizado la conexión física, y se cuenta con el terminal para el acceso, se procede a acceder a su interfaz para solicitar la conexión, en este caso se debe seleccionar la opción de **Puerto COM** y setear la velocidad por defecto de **9600 bps**. La Figura 34 muestra la interfaz inicial del software PuTTY para el establecimiento de conexión con el dispositivo.

Figura 34.

Interfaz de PuTTY previa a la conexión con el dispositivo de red



3. Tras establecer la conexión física, es necesario contar con una conexión lógica entre la PC y el dispositivo de red, para ello se realizan configuraciones en ambos, para el dispositivo de red, en este caso un switch, se hace uso de la VLAN 1 para administrar el equipo, por lo que para establecer la conexión lógica se procede a configurar una dirección

IP del rango de direcciones privadas, esto se realiza a través de la consola serial y de la introducción de las líneas de comando especificadas en el Bloque de Comandos 1.

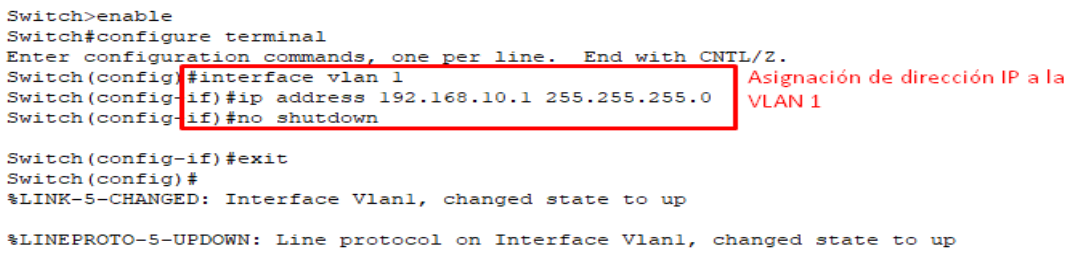
Bloque de Comandos 1.
Asignación de direcciones IP a una VLAN

```
enable
configure terminal
interface vlan 1
ip address 192.168.10.1 255.255.255.0
no shutdown
exit
```

La Figura 35 evidencia la aplicación de estas configuraciones dentro de un switch Cisco 2960 que se toma como ejemplo para el proceso de actualización de firmware.

Figura 35.

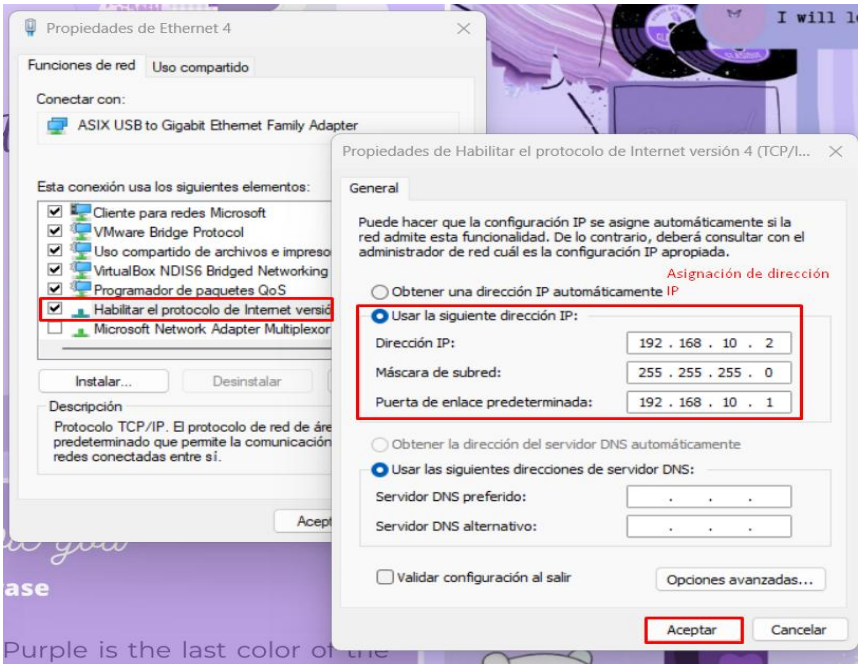
Asignación de dirección IP a la VLAN 1 en un switch Cisco



4. Para lograr el enlace entre el switch y la PC, es necesario asignar una dirección IP dentro de la misma subred de la que se ha configurado en la VLAN del switch, para acceder a las configuraciones del adaptador de red de la PC y realizar la configuración respectiva, la Figura 36 ejemplifica el proceso.

Figura 36.

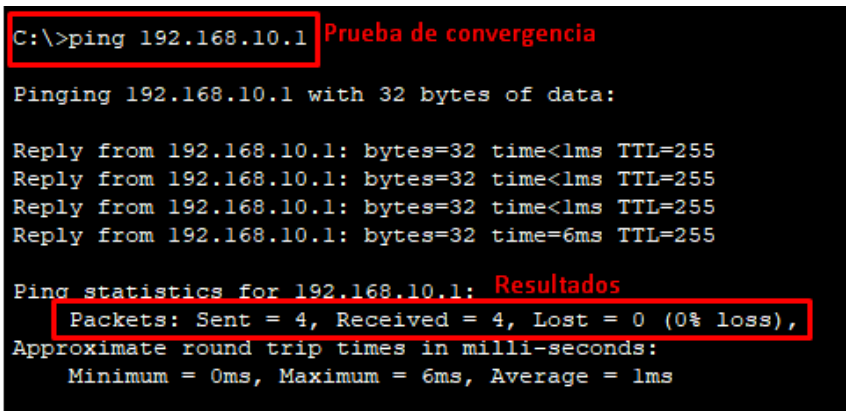
Configuración de dirección IP estática a la interfaz de red la PC



5. Finalmente, se procede a verificar la comunicación entre la PC y el dispositivo de red mediante una petición de ping hacia el contrario, en este caso se realiza hacia la VLAN 1 con la dirección IP 192.168.10.1 con un resultado exitoso como se muestra en la Figura 37.

Figura 37.

Prueba de convergencia mediante solicitud ICMP (Ping) desde la PC hacia el Switch

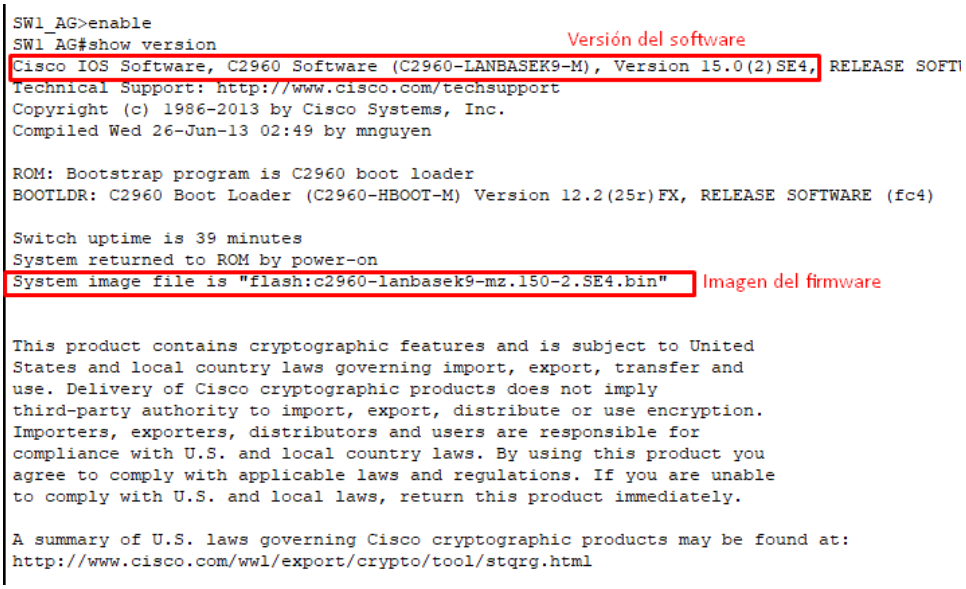


- **Verificación de versiones:**

Desde la consola serial se procede a verificar la versión de software actual del dispositivo de red, para ello se introducen los comandos `enable` para visualizar estadísticas y características del dispositivo, posteriormente se introduce el comando `show version` para visualizar la versión del software, detalles sobre el tiempo de actividad del dispositivo, y el nombre de la imagen (.bin) que se encuentra cargada actualmente en la memoria flash del dispositivo, esta información se puede apreciar en la Figura 38. Con esta información es posible determinar si el equipo se encuentra actualizado.

Figura 38.

Verificación de la versión actual del software del dispositivo de red



1. Al conocer la versión de firmware cargada en el dispositivo se procede a verificar que esta se encuentre en una versión estable y que preferiblemente esta encaje con la recomendación del fabricante especificada en su web oficial https://www.cisco.com/c/es_mx/support/docs/switches/catalyst-9300-

series-switches/214814-recommended-releases-for-catalyst-9200-9.html, esto permite que se mantengan las actualizaciones y parches de seguridad en los dispositivos y evitar fallos o conflictos relacionados al grado de soporte, gestión de actualizaciones, entre otras.

2. Las versiones de firmware pueden obtenerse del sitio oficial de descargas de Cisco, se debe acceder al sitio de descargas: https://software.cisco.com/download/home con la cuenta de usuario asociada a los licenciamiento o contratos de soporte de los equipos para buscar el modelo del dispositivo y seleccionar la versión de firmware que se desee obtener y descargarla.

La nomenclatura manejada por Cisco para el nombre de sus imágenes de sistema operativo consta de diferentes partes ordenadas a modo de trenes para brindar información respecto a las versiones establecidas para los sistemas operativos existentes. Por lo que de forma general el nombre de la imagen se integra por once componentes que incluyen: hardware, designación de la imagen, ubicación de la memoria, formato de compresión, indicador de firma digital (opcional en ciertos casos), versión principal, versión secundaria, versión con características nuevas, versión de mantenimiento extendido recopilación de mantenimiento y extensión del archivo(CCNA desde Cero, 2017). Tomando como referencia el nombre de imagen mostrado en la Figura 38 se puede señalar cada uno de estos componentes en la Tabla 28 presentada a continuación

Tabla 28.

Nomenclatura explicada del archivo c2960-lanbasek9-mz.150-2.SE4.bin

Componente del nombre de archivo	**Valor asociado**
Nombre completo	c2960-lanbasek9-mz.150-2.SE4.bin
Hardware	c2960
Designación de imagen	lanbasek9
Ubicación en memoria	m
Formato de compresión	z
Indicador de firma digital	No especificado
Versión principal	15
Versión secundaria	0
Versión con nuevas características	2
Versión de mantenimiento extendido	SE
Recopilación de mantenimiento	4
Extensión del archivo	bin

De este modo se determina que el archivo pertenece a la serie de imágenes para switches Cisco 2960 con características del licenciamiento LanBase, y se ejecuta en la memoria RAM en formato comprimido, no especifica la firma digital de Cisco, corresponde a la versión 15.0(2) SE4 y es un archivo binario ejecutable (.bin).

- **Preparación el servidor:**

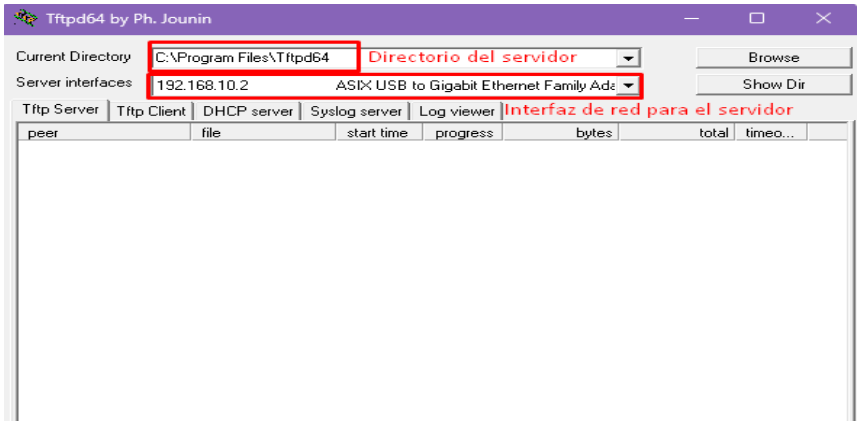
1. Una de las formas más comunes y sencillas de realizar el proceso de actualización del firmware de un dispositivo Cisco es mediante un servidor TFTP para la transferencia de archivos, para ello es necesario

contar con alguna versión de este servidor, en el caso de hacer uso de una PC con sistema operativo Windows, una alternativa simple es la obtención de TFTPD64, un servidor de licencia gratuita que permite realizar la transferencia de los archivos de respaldo así como la nueva imagen para el sistema del dispositivo de forma sencilla, el servidor se descarga desde su sitio oficial: https://pjo2.github.io/tftpd64/, y se instala en la PC de forma previa a la realización del proceso de actualización. Existen otros servidores que pueden ser empelados sin problema para este proceso.

2. Dentro de la configuración base del servidor es necesario seleccionar el directorio en el que se van a guardar los archivos, y de donde se extraerán los archivos a transferir. De igual forma, se debe seleccionar la interfaz de red por la que se va a comunicar con otros elementos, para ello se debe escoger la interfaz configurada con la dirección IP perteneciente a la misma red interna que el switch para poder establecer la conexión y de este modo realizar la trasferencia de archivos sin problemas, Figura 39 indica la configuración de ejemplo para este caso.

Figura 39.

Configuración del servidor TFTP para la transferencia de archivos con el switch

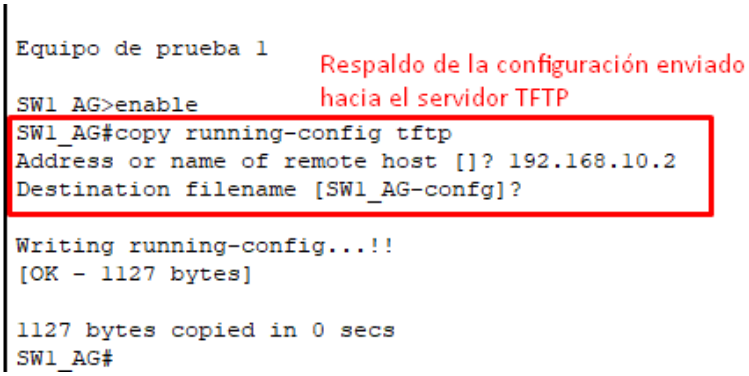


- **Generación de respaldos:**

1. Posterior a la configuración del servidor TFTP, es necesario obtener respaldos para garantizar que la información relevante del equipo no se pierda en el proceso, para ello se realiza una copia de la configuración del switch, esto para que en caso de existir alguna complicación o error que pueda perjudicar al equipo la configuración no se pierda y pueda ser cargada de nueva cuenta. Para ello, se realiza el proceso evidenciado en la Figura 40, en este caso de emplea el comando "`copy running-config tftp`", y a continuación se solicitarán datos como: dirección IP destino y nombre del archivo en el destino, una vez indicados estos aspectos se procede con el envío hacia el servidor.

Figura 40.

Envío de copia de la configuración actual del switch hacia el servidor TFTP



2. De forma similar se genera un respaldo de la imagen actual, esto porque posteriormente la imagen será eliminada y se recomienda guardar una copia de la imagen original en caso de que se requiera regresar a la versión previa del firmware se pueda hacer uso de la misma para garantizar la compatibilidad de forma directa, para ello se hace uso del comando

“`copy flash: tftp`”, y se procede a completar los siguientes datos: nombre del archivo que se va a copiar, dirección IP del servidor TFTP, nombre del archivo en el destino y se espera a que finalice el proceso de transferencia. Es importante considerar que el nombre del archivo debe ser escrito exactamente igual a como se registra para que pueda enviarse, caso contrario existirá un error y la transferencia no se concretará, por ello se debe tomar en cuenta el nombre que se imprimió en pantalla, o revisar el nombre de este con el comando “`show flash`”. La Figura 41 ilustra el proceso realizado en el switch de prueba para el envío del respaldo hacia el servidor TFTP.

Figura 41.

Envío de copia del firmware actual del switch hacia el servidor TFTP

Equipo de prueba 1 — Copia de la imagen actual enviada al servidor TFTP

```
SW1_AG>enable
SW1_AG#copy flash: tftp
Source filename []? 2960-lanbasek9-mz.150-2.SE4.bin
Address or name of remote host []? 192.168.10.2
Destination filename [2960-lanbasek9-mz.150-2.SE4.bin]?

Writing 2960-lanbasek9-mz.
150-2.SE4.bin...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
[OK - 4670455 bytes]

4670455 bytes copied in 0.06 secs (6258128 bytes/sec)
SW1_AG#
SW1_AG#
```

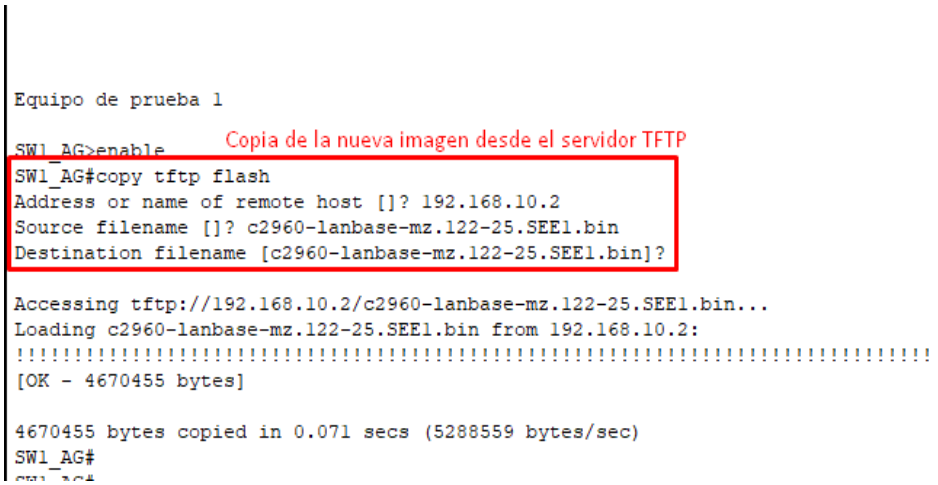
- **Obtención de la nueva imagen y configuración:**

1. A continuación, se procede a descargar la nueva imagen desde el servidor, para ello se debe haber descargado previamente la nueva imagen del sitio oficial y haberla colocado en el directorio especifico del servidor TFTP, una vez hecho esto, dentro del equipo se procede a ejecutar el comando “`copy tftp flash`”, mismo que da la orden de realizar la

transferencia de un archivo desde el servidor TFTP hacia la memoria flash del equipo, y se procede a completar los datos solicitados como: dirección IP del servidor, nombre del archivo en la fuente (tal como se haya guardado dentro del directorio) y nombre del archivo en el destino, una vez concretado se inicia el proceso de transferencia y al finalizar este con éxito se puede proseguir con el siguiente paso. La **Figura 42** muestra el proceso realizado con el switch de prueba para obtener la nueva imagen.

Figura 42.

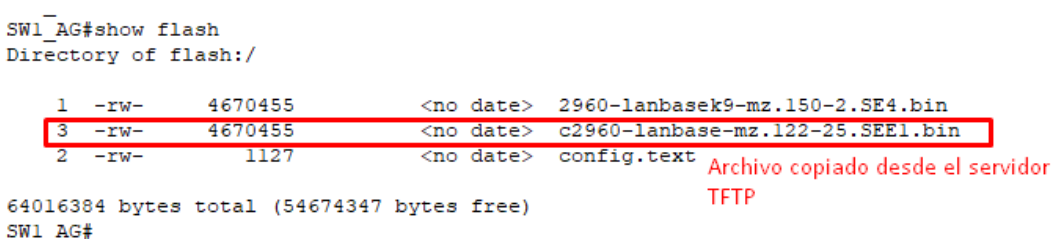
Transferencia del nueva firmware hacia el switch Cisco



2. Al finalizar el proceso de transferencia de archivos desde el servidor TFTP hacia el equipo, se debe verificar que se haya copiado el archivo correcto y que esté en la ubicación adecuada, para ello se hace uso del comando "`show flash`" que muestra un listado de todos los archivos existentes en la memoria flash del dispositivo. La Figura 43 evidencia este proceso tras la obtención de la nueva imagen para la actualización del firmware.

Figura 43.

Verificación de los archivos existentes en la memoria flash del dispositivo tras la copia de la nueva imagen



3. Para setear una nueva imagen como archivo de inicio de sistema en los dispositivos, es necesario contar con la nueva imagen e introducir la configuración indicada en el Bloque de Comandos 2, en este casos se requiere acceder al modo de configuración global del dispositivo (configure terminal) y proceder con la asignación de la nueva imagen.

Bloque de Comandos 2.

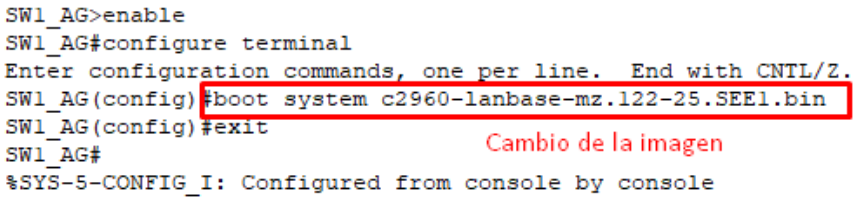
Configuración de la nueva imagen en el sistema

```
enable
configure terminal
boot system <nombre del archivo .bin>
exit
```

La aplicación de esta configuración se muestra en la Figura 44 que evidencia la selección del archivo .bin obtenido del servidor TFTP como nueva imagen de inicio para el switch de prueba.

Figura 44.

Configuración del archivo de inicio dentro de un dispositivo Cisco



- **Reinicio y verificación:**

1. Posterior a setear la nueva imagen pata el inicio del sistema es necesario solicitar un reinicio del dispositivo para finalizar el proceso de actualización, para ello se hace uso del comando "`reload`" y tras guardar las configuraciones y aceptar el reinicio, se debe esperar a que el nuevo firmware se instale y una vez el proceso finalice el dispositivo arrancará de nuevo con la nueva versión ya cargada, este proceso se evidencia en la Figura 45.

Figura 45.

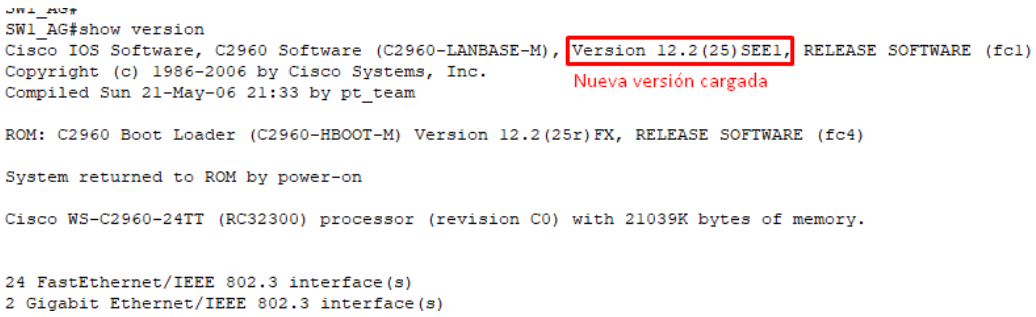
Proceso de reinicio de un dispositivo Cisco

2. Una vez reiniciado el equipo es necesario comprobar que la versión del software haya cambiado de forma exitosa, para ello se repite la

secuencia para verificar que la versión corresponda a la que se ha configurado con anterioridad, en la Figura 46 se puede visualizar este cambio determinando así que el proceso de actualización se ha logrado concretar sin inconvenientes.

Figura 46.

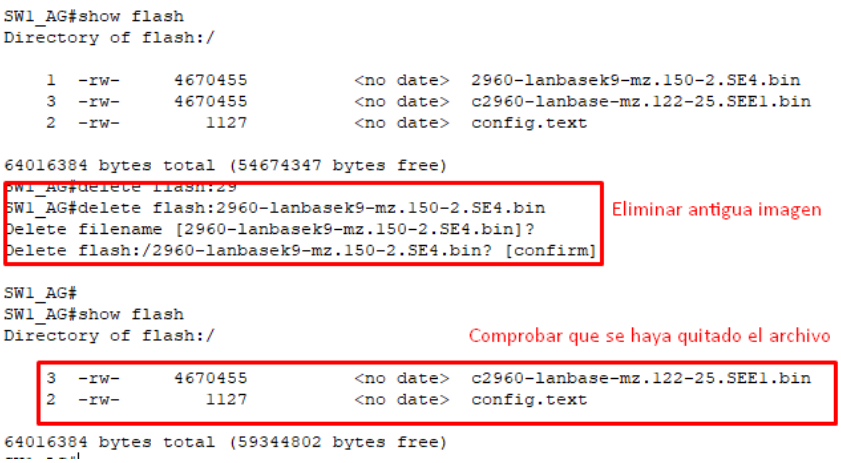
Verificación de la actualización de firmware dentro del dispositivo



3. Tras verificar que la versión se cambió de forma exitosa, es posible eliminar la imagen del firmware anterior, esto debido a que la memoria flash de los dispositivos es poca y no es recomendable mantener archivos que no están siendo usados, por lo que con el comando "`delete flash: < nombre del archivo >` " se puede proceder a eliminar los archivos que no están siendo usados, la **Figura 47** demuestra este proceso, mostrando en primera instancia todos los archivos existentes para posteriormente eliminar la imagen anterior y finalmente verificar que el archivo ya no se liste con el comando "`show flash`".

Figura 47.

Proceso para eliminar la imagen anterior de la memoria flash



Una vez realizado el proceso de actualización dentro de los equipos es posible visualizar nuevas funcionalidades o correcciones a fallos existentes en la versión previa, esto aporta mayor estabilidad a la arquitectura de red y debe ser considerado como primer paso dentro de las configuraciones a realizar en los equipos previo a la migración hacia la arquitectura de redes definidas por software.

c) Descripción de recursos de hardware por adquirir

Al realizarse la verificación de compatibilidad, se determinó que existen equipos que no pueden integrarse de forma adecuada a la arquitectura de redes definidas por software debido a limitaciones en el soporte del protocolo OpenFlow o incompatibilidad con la tecnología indicada. Con la finalidad de contar con un rendimiento eficiente y garantizar la comunicación adecuada, es necesario reemplazar estos equipos por dispositivos que cumplan con los parámetros de compatibilidad requeridos.

Los dispositivos marcados como no compatibles en la Tabla 27 deben ser reemplazados por una versión más reciente o que admita la actualización a los sistemas operativos estables compatibles con el protocolo OpenFlow. Los equipos de la serie 2960X, aunque compatibles de manera limitada, no son óptimos para SDN debido a sus restricciones. Se recomienda reemplazarlos por modelos con soporte completo para OpenFlow 1.3, que garanticen escalabilidad y rendimiento a largo plazo.

En base al análisis realizado se determina que existen ocho dispositivos que tendrán que reemplazarse dentro de la red de distribución, por lo que entre las alternativas a tomar en cuenta se puede contar con equipos dentro de la misma serie de Cisco Catalyst, algunos de los modelos sugeridos son:

- Cisco Catalyst 9300 o 9300X: son modelos estables y ofrecen capacidades de conexión óptimas para redes de distribución debido a su capacidad administrativa y su compatibilidad con OpenFlow 1.3, lo cual facilita la integración de estos equipos a una SDN.
- Cisco Catalyst 9400 o 9500 Series: son versiones superiores a los dispositivos existentes en la red, debido a la integración de versiones actuales del sistema operativo pueden integrarse a soluciones SDN sin problema con soporte para el protocolo OpenFlow 1.3, además de ser equipos diseñados para brindar alto rendimiento dentro de entornos empresariales de alta demanda.

Para albergar al controlador que actuará como componente central de la nueva arquitectura de red, se requiere de un servidor de alto rendimiento, con la robustez necesaria para manejar grandes volúmenes de tráfico

correspondientes a una red institucional. Los modelos sugeridos cumplen con estos requerimientos y admiten distintos sistemas operativos que permiten la instalación de OpenDaylight y otros servicios necesarios.

- Cisco UCS C220 M6: es un servidor de alto rendimiento del fabricante Cisco, cuenta con un procesador Intel Xeon de tercera generación de 40 núcleos, capacidad para memoria RAM máxima de hasta 10 TB, y hasta 76 TB de almacenamiento en SSD.
- Dell PowerEdge R740: es un servidor de alto rendimiento que cuenta con un procesador Intel Xeon de segunda generación de 28 núcleos, capacidad para memoria RAM de hasta 3TB y almacenamiento máximo de 128 TB en SSD ofreciendo capacidad suficiente para cumplir con múltiples funciones dentro de una red.
- HPE ProLiant DL380 Gen10: es un servidor robusto con capacidad de expansión, cuenta con un procesador Intel Xeon de segunda generación de 28 núcleos, hasta 3TB de memoria RAM , y distintas configuraciones de almacenamiento SSD que permiten ir desde los 92 TB hasta los 184 TB.

Para la alternativa de implementar una red híbrida en la arquitectura propuesta, se pueden considerar diversas opciones de routers que actúen como dispositivos dedicados al enrutamiento de tráfico IPv4 e IPv6. Estos equipos deben ser capaces de manejar las demandas actuales de la red institucional, complementando las capacidades del controlador SDN al gestionar tablas de enrutamiento y resolver direcciones de capa 3, mientras proporcionan la información necesaria para generar reglas de flujo específicas. Las opciones recomendadas son:

- **Cisco ISR 4451-X**: es router modular destaca por su capacidad de procesamiento de hasta 2 Gbps, lo que lo hace adecuado para redes institucionales que requieren alta disponibilidad y soporte para dual-stack. Además, su diseño modular permite agregar interfaces adicionales según las necesidades de la red, además cuenta con compatibilidad con protocolos avanzados como BGP, OSPF y EIGRP.
- **Juniper MX204**: está diseñado para entornos de alta densidad, este equipo ofrece un rendimiento de hasta 400 Gbps, lo que lo hace ideal para redes que manejan grandes volúmenes de tráfico. Cuenta con soporte nativo para IPv6 y es adecuado para redes institucionales proporcionando alto rendimiento y flexibilidad para manejar configuraciones complejas.
- **MikroTik CCR2004-16G-2S+:** cuenta con un procesador ARM de 4 núcleos con una frecuencia de hasta 1.7 GHz, soporta IPv6 de manera nativa y cuenta con 16 interfaces Gigabit Ethernet y 2 interfaces SFP+ de 10 Gbps, siendo ideal para conexiones de alta velocidad.

d) Planteamiento de la nueva arquitectura de red.

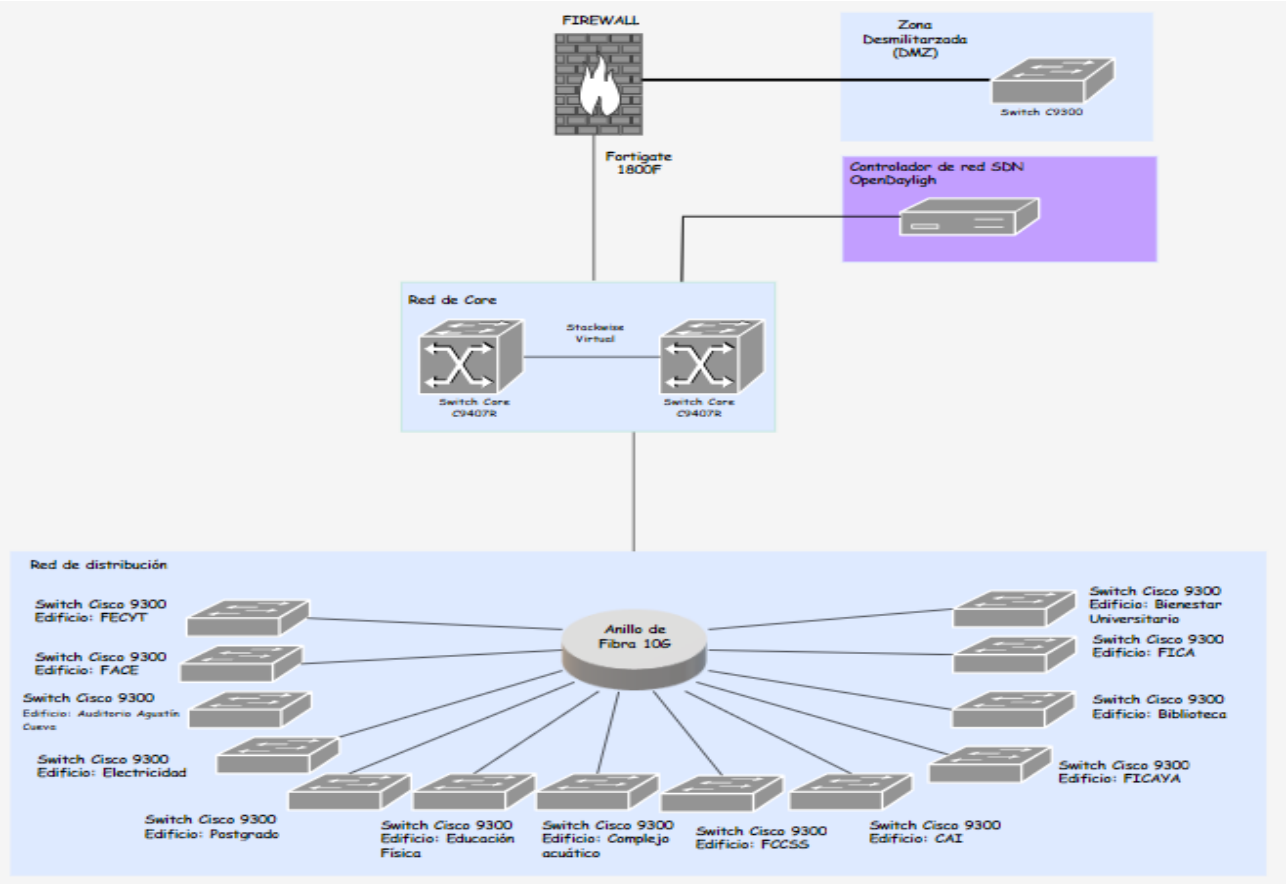
La migración hacia una red definida por software en la UTN tiene como objetivo mejorar la gestión del tráfico y facilitar la toma de decisiones dentro de la infraestructura de red. Este cambio requiere ajustes en la arquitectura existente, para establecer una gestión centralizada que permita mayor control y flexibilidad.

El primer paso clave es la integración del controlador SDN en la red. Los dispositivos que se conectarán al controlador serán aquellos pertenecientes a la red de core y distribución, ya que estas capas son responsables de las decisiones críticas de la red, como el enrutamiento y la gestión del tráfico. No se consideran cambios en los equipos de la capa de acceso ya que, esta capa se especializa en la conectividad básica de los terminales finales de la red y no se toman decisiones de direccionamiento. La Figura 29 planteada anteriormente muestra la posición del controlador en la topología general de la red.

De forma simplificada, se puede apreciar en la Figura 48 los dispositivos que se quiere integrar a la arquitectura de red definida por software, por ello se considera que, si dentro de la red de distribución se cambian los equipos no compatibles, es posible establecer una comunicación indirecta entre el controlador de red y el switch del core de la red para de este modo gestionar el tráfico de la red desde el punto central.

Figura 48.

Arquitectura de red definida por software planteada



Si bien la arquitectura de red original no sufrirá muchos cambios a nivel físico , si considera la integración del elemento central de la nueva red el cual permitirá contar con una gestión centralizada para el tráfico de la red. Esta arquitectura en principio puede fácilmente adaptarse a entornos altamente congestionados y a su vez es bastante escalable, ya que permite que fácilmente puedan añadirse equipos más robustos en la red de distribución para mejorar el rendimiento, ya que al mantener comunicación indirecta con el controlador es más sencillo integrar los nuevos dispositivos a la red siempre y cuando estos cuenten con la compatibilidad necesaria para dicha integración.

Para integrar el controlador OpenDaylight, se establece una conexión física de alta velocidad entre este y los switches del core mediante interfaces de fibra óptica. La conexión de fibra óptica es la más indicada para estas conexiones debido a que ofrece mayores tasas de transferencia de datos y es inmune a condiciones adversas como el ruido electromagnético. Esta conexión permite crear un canal lógico seguro, a través del cual se gestionará el tráfico usando mensajes OpenFlow. El elemento que interactúa directamente entre estas dos partes es el protocolo OpenFlow, el cual comunica la interfaz southbound del controlador con el plano de datos de los switches haciendo uso de tres tipos de mensajes específicos que son: mensajes Controller-to-switch, mensajes simétricos y mensajes asimétricos.

Los mensajes OpenFlow, son necesarios para mantener la comunicación entre las entidades de la red, Según (Open Network Foundation, 2013), los mensajes cumplen las siguientes funciones:

- **Mensajes Controller-to-switch:** son mensajes enviados desde el controlador hacia el switch que pueden requerir o no de una respuestas, se emplean para solicitar información o funciones específicas del switch para procesos como el establecimiento de canal. Pueden ser mensajes de: cambio de estado, lectura de estado, mensajes Packet-in, mensajes de barrera, solicitud de rol o configuración asincrónica.
- **Mensajes simétricos:** son mensajes que pueden ser enviados por cualquiera de las dos partes y se emplean principalmente para verificar

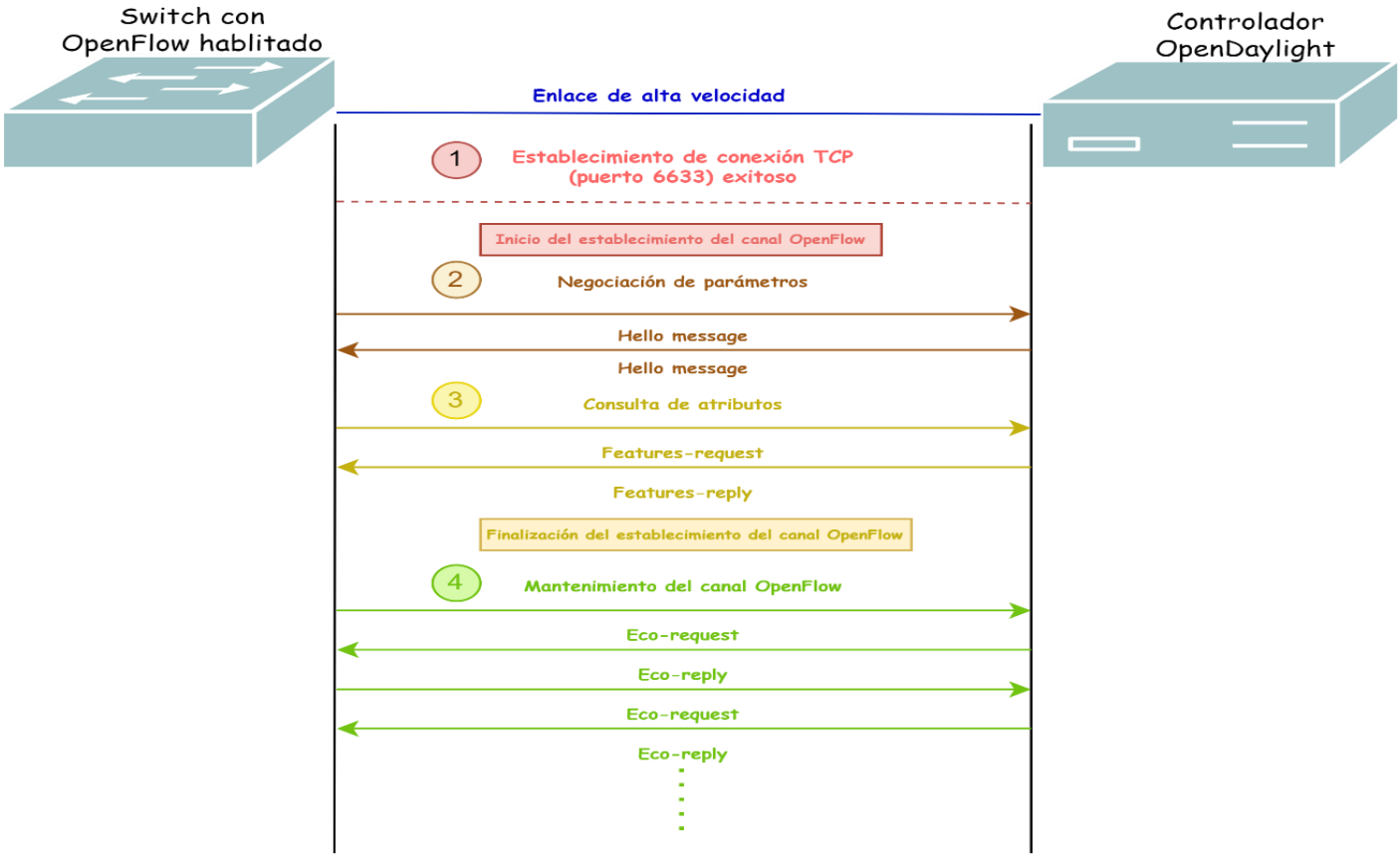
la existencia de conexión entre ambas partes. Algunos de estos son: mensajes hello, eco-request y eco-reply.

- **Mensajes asincrónicos:** son mensajes enviados por el switch hacia el controlador para notificar acciones como llegada de paquetes o cambios en el estado del conmutador. Algunos de estos mensajes son: mensajes Packet-in, mensajes Flow-removed, mensajes de error y mensajes port-status.

Dentro de la arquitectura de red definida por software que se propone deben existir procesos que garanticen la integración adecuada de los dispositivos de red y el controlador. Al hacer uso del protocolo OpenFlow y de los mensajes que define, el primer paso es el establecimiento y mantenimiento del canal OpenFlow, este realiza posterior a un establecimiento de conexión TCP entre el controlador y el switch a través del puerto 6633. El establecimiento del canal se explica mediante la **Figura 49**.

Figura 49.

Establecimiento y mantenimiento del canal OpenFlow entre switch y controlador



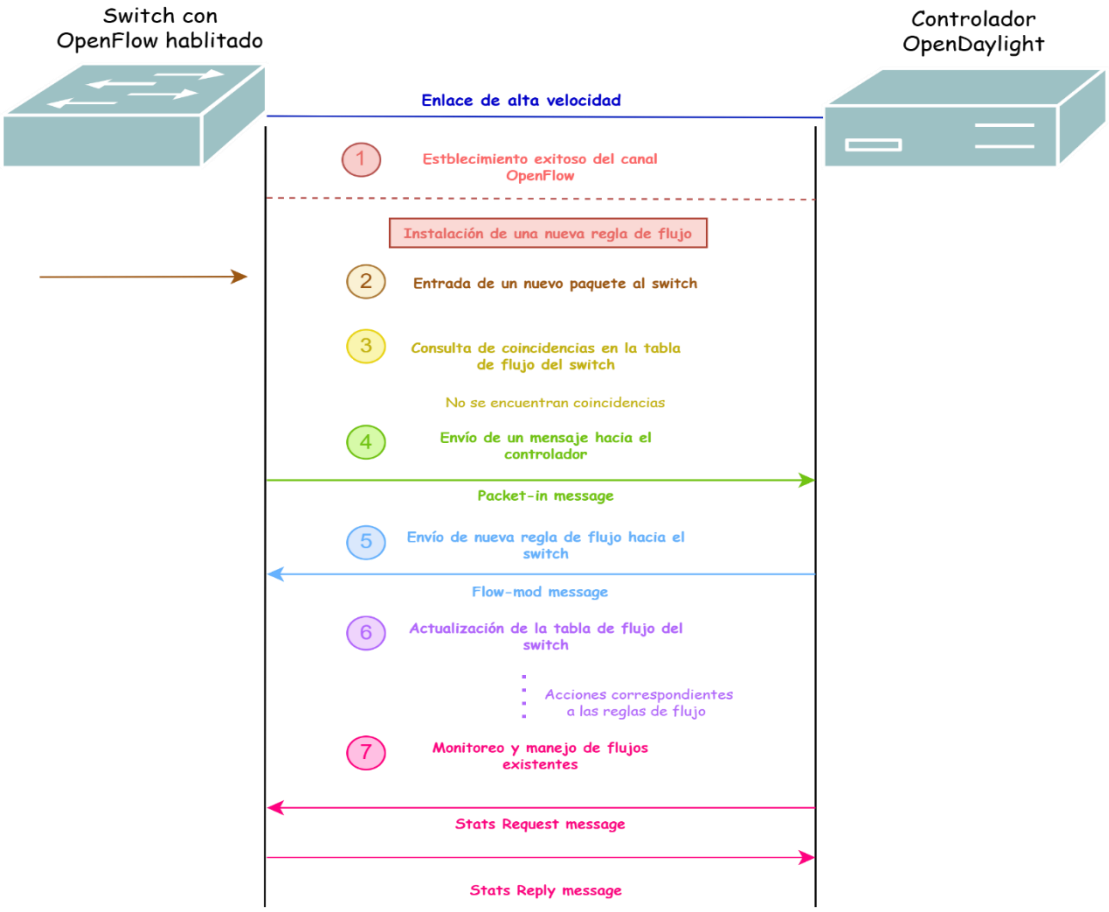
Una vez finalizado el establecimiento de conexión TCP, el controlador envía un mensaje simétrico hacia el switch, este es un mensaje hello que debe ser respondido por el switch para la negociación de parámetros como la versión del protocolo que se empleará. Posteriormente se deben intercambiar los mensajes de solicitud de características hacia el switch con un mensaje features-request, y este se responde con un features-reply que contiene información como formatos de la tabla de flujo y tamaño de búffer. Una vez se intercambian estos mensajes el canal OpenFlow para el reenvío se ha establecido, si por el contrario a lo largo de estos intercambios se encuentran mensajes de error es necesario revisar

las partes involucradas. Para el mantenimiento del canal se emplean los mensajes de tipo simétrico eco-request y eco-reply para supervisar el estado de las dos partes del canal, en caso de que un mensaje eco-request no tenga una respuesta se considera un fallo en el otro extremo del canal y el canal se cierra alertando de este percance.

Además de mantener el canal OpenFlow, la gestión de reglas de flujo es otro proceso crucial en la comunicación entre los switches y el controlador es la instalación de reglas de flujo en las tablas de flujo de los conmutadores. Este proceso ocurre cuando ingresan nuevos paquetes o flujos que no coinciden con las reglas de flujo previamente establecidas. En la **Figura 50** se detalla este procedimiento.

Figura 50.

Instalación de una nueva regla de flujo en un switch



Una vez que el canal OpenFlow se ha establecido correctamente, los dispositivos de red empiezan a operar reenviando paquetes. Al recibir un nuevo paquete en uno de los puertos de entrada del switch, este consulta la tabla de flujos buscando coincidencias con las cabeceras del paquete, tales como direcciones IP, direcciones MAC, números de puertos, entre otros. Si el paquete no coincide con ninguna de las reglas existentes, el switch no puede tomar una decisión autónoma sobre su reenvío y genera un mensaje Packet-in que se envía hacia el controlador. Este contiene información relevante del paquete contenida en sus cabeceras y el puerto de entrada al Switch.

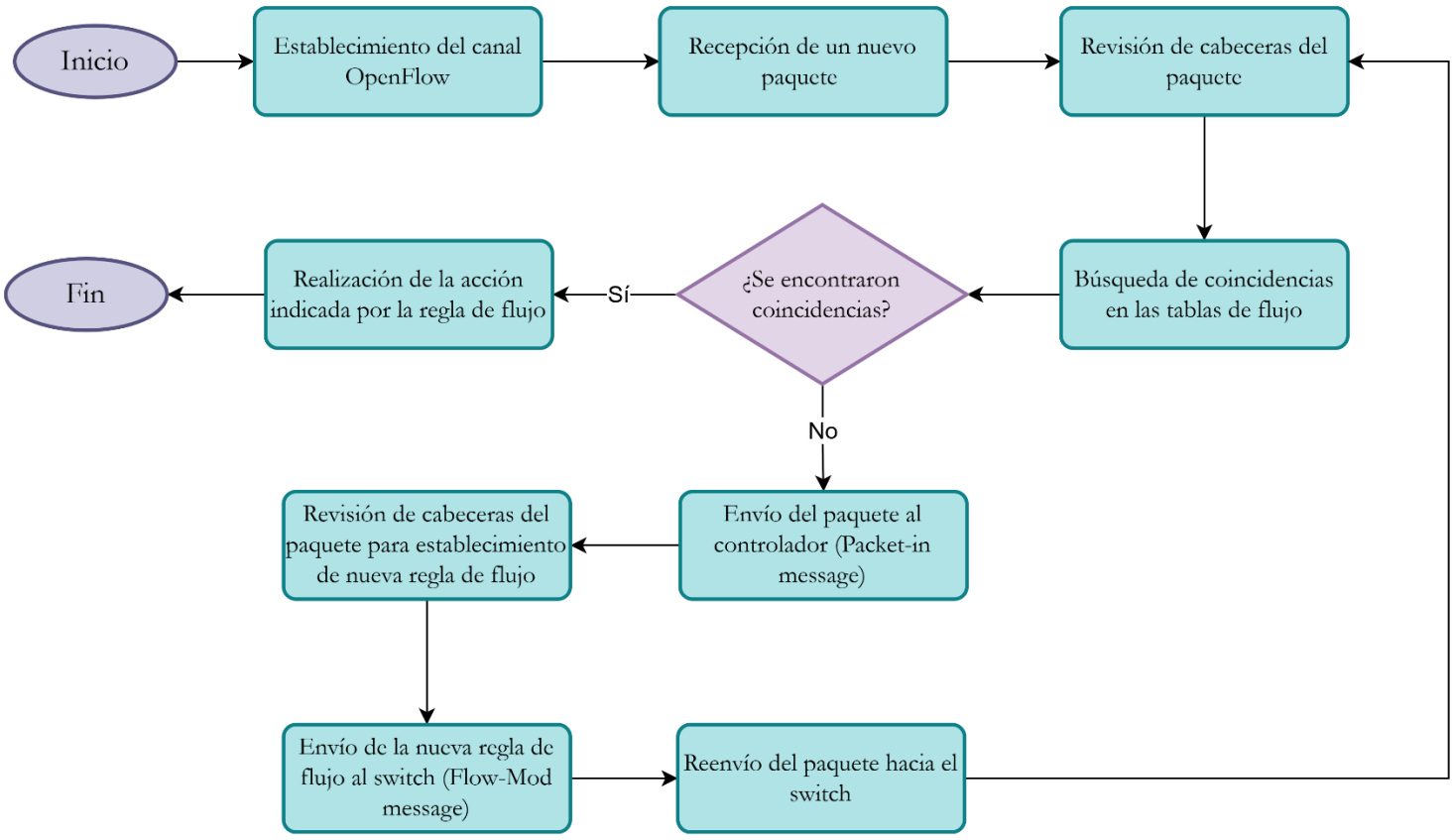
El controlador analiza el paquete y decide la acción correspondiente para este en función a las políticas y condiciones de la red. Posteriormente, envía un mensaje Flow-Mod hacia el switch con una nueva regla de flujo. Esta regla establece nuevas coincidencias y acciones para próximos paquetes o flujos que tengan las mismas características, así como la prioridad de la nueva regla y el tiempo de vida asignado, para así garantizar que futuros paquetes se procesen de forma directa por el switch sin realizar una nueva consulta al controlador.

Una vez instalada la nueva regla de flujo el controlador continúa con sus funciones de monitoreo enviando mensajes Stats-Request para obtener estadísticas del tráfico procesado por cada regla, el número de paquetes, bytes transmitidos y tiempo de vida restante de cada regla. El switch responde con un Stats-Reply, permitiendo al controlador recopilar información para evaluar el rendimiento de la red y realizar ajustes en tiempo real acorde a las necesidades de la red.

Dentro de una red definida por software la interacción entre los switches y el controlador debe ser constante, por lo que para la gestión de un paquete que ingrese por un puerto de uno de los switches se sigue el proceso especificado en la **Figura 51**.

Figura 51.

Procesamiento de un nuevo paquete en una SDN con el protocolo OpenFlow



El procesamiento de un nuevo paquete en una (SDN inicia con el establecimiento del canal OpenFlow entre el controlador y el switch, como se muestra en la Figura 49. Este canal permite que el controlador configure previamente las reglas de flujo en el switch de acuerdo con las necesidades de la red. Cuando un paquete nuevo llega a uno de los puertos del switch, el primer paso es la revisión de sus cabeceras para verificar si coincide con alguna de las reglas de flujo ya establecidas en las tablas del switch. Estas coincidencias se buscan en función de parámetros como la dirección IP, la dirección MAC, el número de puerto, entre otros. La verificación se realiza

en orden de prioridad, las reglas están organizadas para que la coincidencia se detecte tan pronto como el paquete cumpla con los criterios definidos en una entrada de la tabla.

Si el paquete coincide con una regla en la tabla de flujo, el switch ejecuta la acción asociada a esa regla, que puede ser Forward que corresponde al reenvío del paquete a un puerto de salida específico, **Modify** a la modificación de ciertos elementos de las cabeceras del paquete para ajustarlo a un procesamiento especial o **Drop** que es el descarte del paquete. Sin embargo, si el paquete no coincide con ninguna regla en la tabla, se aplica la regla "miss flow", enviándose el paquete al controlador mediante un mensaje Packet-In. El controlador analiza el paquete y, en caso de que se necesite gestionar futuros paquetes similares, puede instalar una nueva regla de flujo en el switch. Esta nueva regla se envía de regreso al switch en un mensaje Flow-Mod, como se muestra en la Figura 50, permitiendo que el paquete sea procesado de acuerdo con la nueva regla. Una vez que el switch recibe esta regla, repite el proceso de revisión y aplica las acciones necesarias, completando así el ciclo de procesamiento del paquete en la red SDN.

La propuesta de esta nueva arquitectura de red definida por software para la UTN busca centralizar la gestión del tráfico y facilitar la administración de la infraestructura de red en sus capas de core y distribución. La explicación del proceso de procesamiento de paquetes y de los mensajes de control entre el controlador y los switches permite visualizar cómo se establece y mantiene la comunicación en tiempo real, un aspecto clave para el funcionamiento eficiente de la SDN. . Esta

estructura centralizada no solo mejora la capacidad de respuesta de la red, sino que también sienta las bases para futuras expansiones y optimizaciones, alineándose con las necesidades de una infraestructura de red escalable y adaptable.

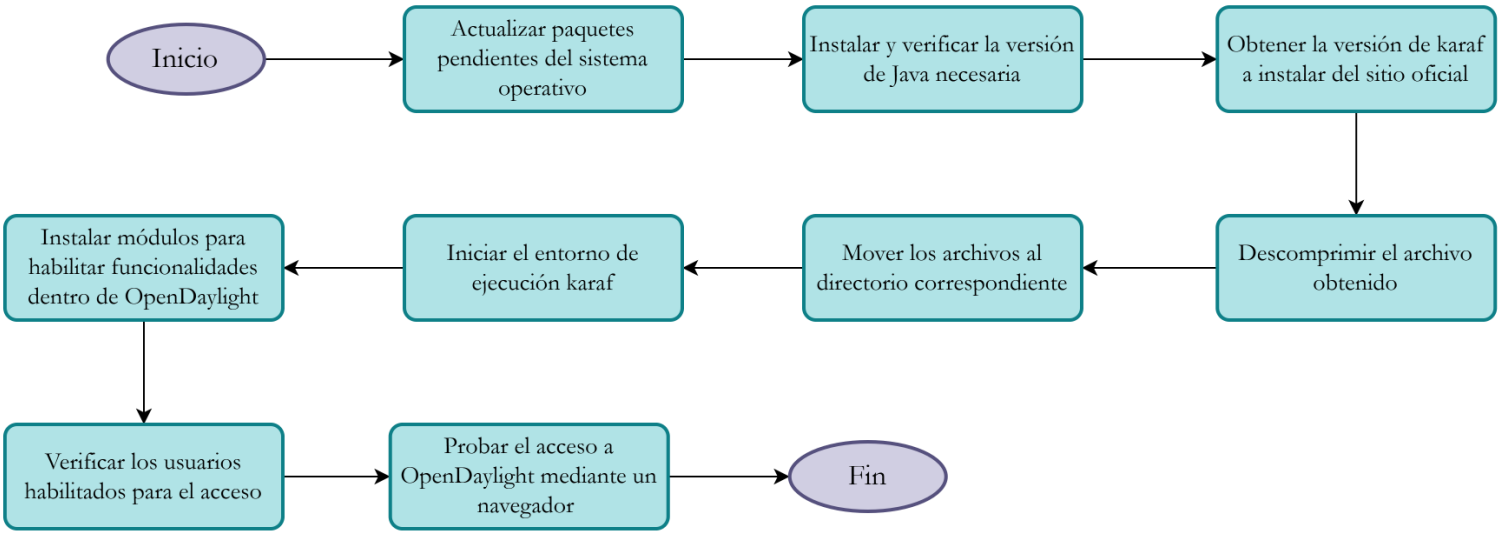
e) Instalación del controlador OpenDaylight

El controlador de red es el componente principal de una red definida por software, ya que centraliza la gestión y el control de los dispositivos de red mediante un plano de control separado. En la arquitectura propuesta para la red del datacenter de la UTN, se seleccionó OpenDaylight como controlador SDN debido a su flexibilidad y capacidad para interactuar con diversos protocolos, incluido OpenFlow 1.3, que es compatible con los switches Cisco Catalyst 9300 en la red. Para implementar OpenDaylight, se ha elegido Apache Karaf 0.8.4 como contenedor, una versión conocida por su estabilidad y compatibilidad. Karaf permite instalar y gestionar de manera modular los diferentes componentes o features de OpenDaylight, proporcionando un entorno adaptable y escalable.

La Figura 52 detalla los pasos necesarios para instalar y configurar el controlador OpenDaylight en el servidor, posterior a la instalación del sistema operativo, proceso descrito en el **Anexo 1**.

Figura 52.

Proceso de instalación del controlador OpenDaylight



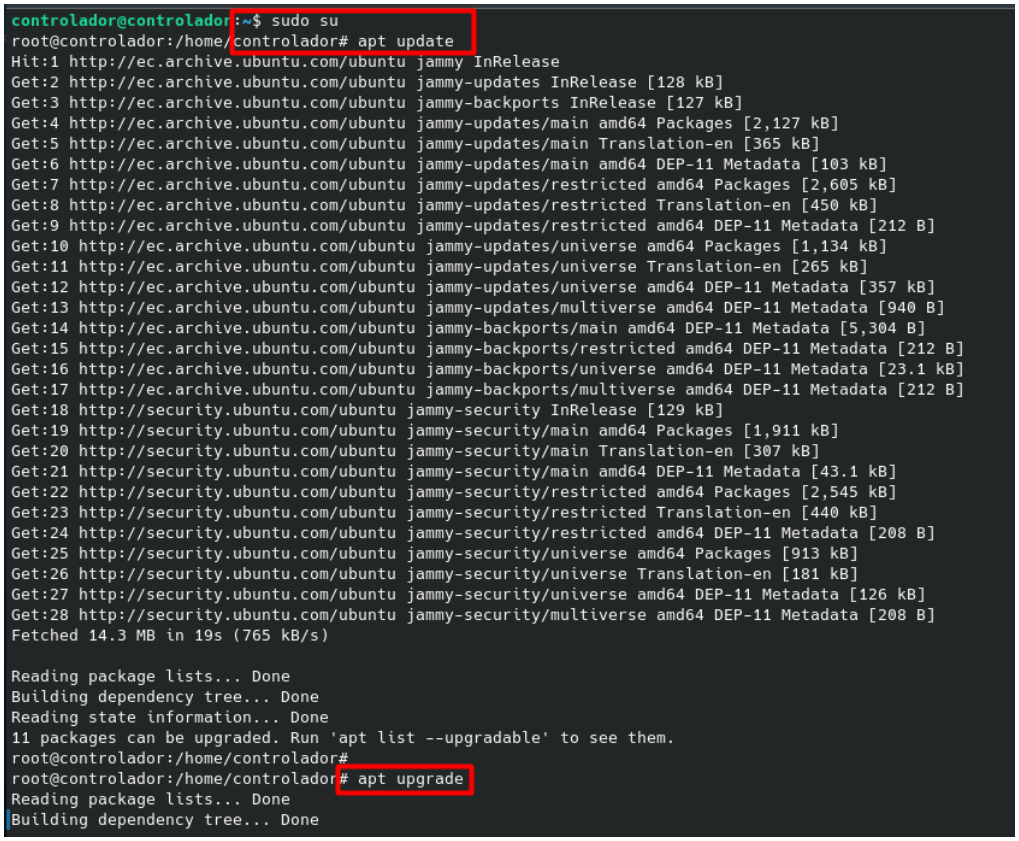
La instalación del controlador en las distribuciones de Linux se realiza de forma similar, por lo que en el caso específico de Ubuntu Server 22.04, se realiza siguiendo los pasos especificados a continuación:

- **Actualizar e instalar complementos.**

  1. El primer paso a realizar para realizar la instalación del controlador OpenDaylight es realizar la actualización de los paquetes incorporados dentro del sistema operativo, para ello es necesario contar con permisos de super usuario y consultar por los paquetes que requieren de actualización con el comando `apt update`, el mismo da pie a iniciar el proceso de actualización con el comando `apt upgrade`. La Figura 53 muestra el proceso de actualización realizado para el sistema operativo seleccionado y permite que las descargas se realicen de manera óptima.

Figura 53.

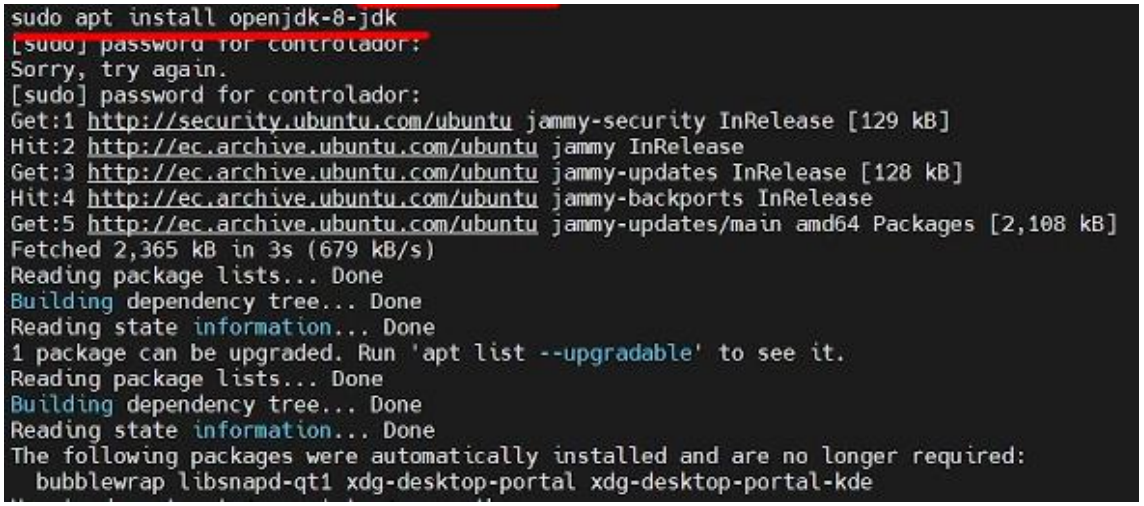
Actualización de paquetes dentro de Ubuntu Server



2. Una vez se ha completado el proceso de actualización, se procede con la descarga de la versión de Java necesaria para el funcionamiento del contener Karaf que se emplea para ejecutar el controlador OpenDaylight, esto se realiza con el comando `sudo apt install openjdk-8-jdk`. La Figura 54 muestra el proceso realizado.

Figura 54.

Instalación de OpenJDK en la versión necesaria



3. Finalmente, se procede a verificar que la versión de Java sea la correcta mediante el comando `java -version`, el cual muestra la versión empleada de OpenJDK que está siendo empleada por el sistema Operativo. La Figura 55 muestra el resultado de esta consulta indicando la versión 1.8 que corresponde a Java 8.

Figura 55.

Verificación de la versión de OpenJDK

- **Descargar contenedor Karaf**

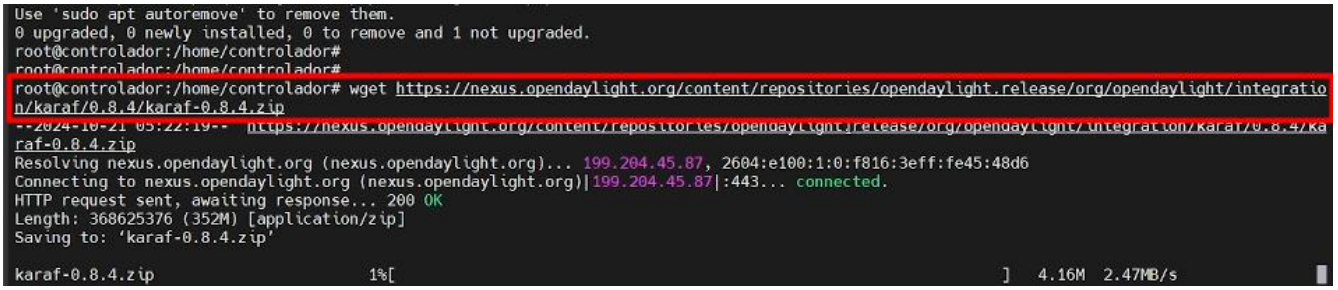
1. A continuación, es necesario acceder al sitio oficial de OpenDaylight (https://docs.opendaylight.org/en/latest/downloads.html) para obtener el enlace de descarga que será empleado posteriormente. Dentro se encuentra la documentación y las diferentes versiones disponibles, es

posible navegar en el sitio y seleccionar la que mejor se acople. En este caso se seleccionó la versión 0.8.4 que corresponde a las ediciones Nitrogen and Oxygen que se caracterizan por su alta compatibilidad y estabilidad.

2. Una vez se ha seleccionado la versión de karaf que se va a emplear, dentro del sistema operativo se realiza la descarga, para ello se debe seleccionar el directorio correspondiente y obtener el enlace directo. Dentro del sistema operativo seleccionado se pueden usar distintas herramientas para la descarga del archivo, en este caso se hizo uso del comando `wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/karaf/0.8.4/karaf-0.8.4.zip` como se muestra en la Figura 56.

Figura 56.

Descarga de karaf desde el sitio oficial de OpenDaylight



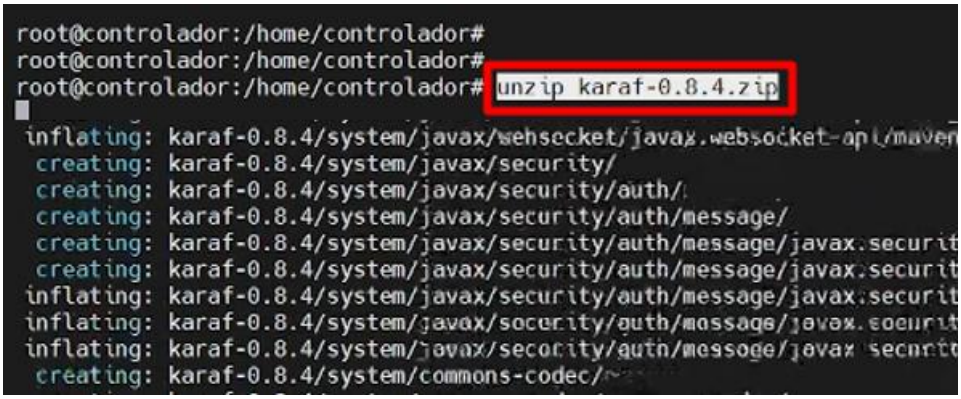
- **Descomprimir y mover archivos**

1. Con el archivo de Karaf descargado , se debe descomprimir para extraer los archivos necesarios. Esto se realiza con el comando `unzip karaf-0.8.4.zip` dentro del directorio en el que se encuentra el

archivo descargado. Este proceso permite que todos los componentes de Karaf estén disponibles en una estructura de carpetas accesible para su configuración y ejecución. La Figura 57 muestra el proceso de descompresión del archivo Karaf.

Figura 57.

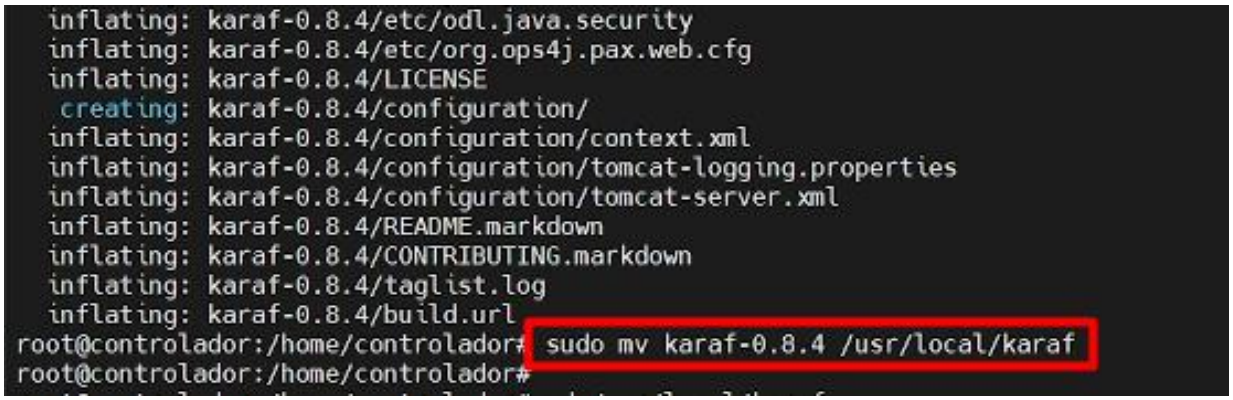
Descompresión de archivo Karaf



2. Posteriormente, mover la carpeta descomprimida a un directorio estándar. Esto se realiza con el comando sudo `mv karaf-0.8.4 /usr/local/karaf`, lo cual coloca a Karaf en el directorio /usr/local, que es el lugar habitual para softwares externos en sistemas Linux. Esto se evidencia en la Figura 58.

Figura 58.

Movimiento de archivos Karaf al directorio /usr/local

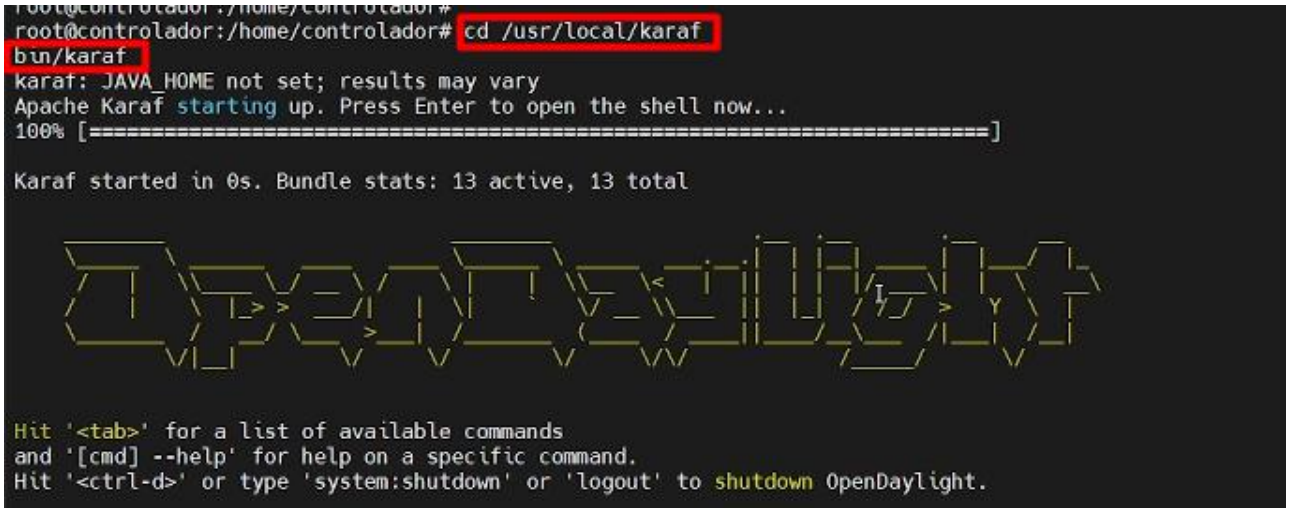


- **Ejecutar Karaf**

1. Antes de iniciar Karaf, es necesario navegar al directorio donde se ubicaron los archivos en el paso anterior. Para esto, se utiliza el comando `cd /usr/local/karaf` , este proceso se muestra en la Figura 59.
2. Una vez dentro del directorio, se puede iniciar Karaf con el comando `bin/karaf`. Esto ejecuta el contenedor de Karaf, permitiendo que el sistema esté listo para cargar los módulos y features de OpenDaylight. Este proceso se muestra en la Figura 59.

   Figura 59.

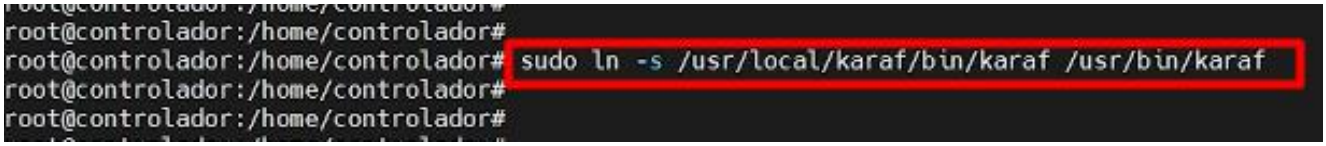
   Cambio de directorio a /usr/local/karaf



3. Para facilitar el acceso a Karaf desde cualquier ubicación en el sistema, se crea un enlace simbólico en el sistema mediante el comando `sudo ln -s /usr/local/karaf/bin/karaf /usr/bin/karaf`. Este enlace permite que Karaf se ejecute con el comando karaf, sin necesidad de especificar la ruta completa. Esto se puede evidenciar en la Figura 60.

Figura 60.

Ejecución de Karaf desde el directorio bin



- **Instalar módulos para OpenDaylight**

1. A continuación, se procede a instalar los módulos que habilitarán las funcionalidades básicas de OpenDaylight. Los comandos para instalar estos módulos son: `feature:install odl-restconf odl-openflowplugin-flow-services odl-openflowplugin-flow-services-rest odl-openflowplugin-southbound .` Estos features habilitan RESTCONF, que proporciona una API para administrar la red, y OpenFlow, que es esencial para la comunicación con los switches SDN. La Figura 61 muestra la instalación de los módulos.

Figura 61.

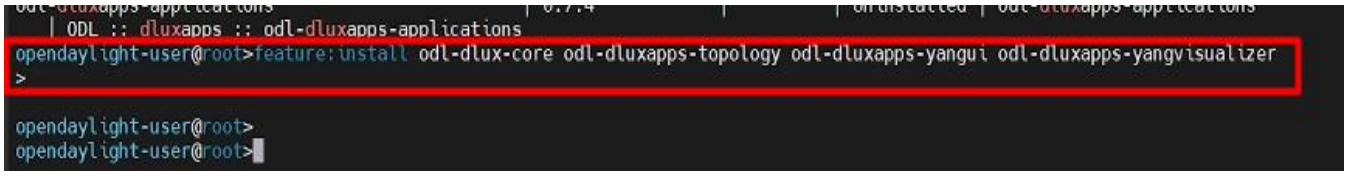
Instalación de módulos de OpenFlow y RESTCONF

2. Para habilitar la interfaz gráfica de usuario de OpenDaylight, se instalan los siguientes módulos: `feature:install odl-dlux-core odl-dluxapps-topology odl-dluxapps-yangui odl-dluxapps-yangvisualizer`. Estos módulos permiten visualizar la topología de red, explorar modelos YANG y acceder a una interfaz

gráfica que facilita la administración y visualización del estado de la red. Esto se evidencia en la Figura 62.

Figura 62.

Instalación de módulos DLUX para la interfaz gráfica

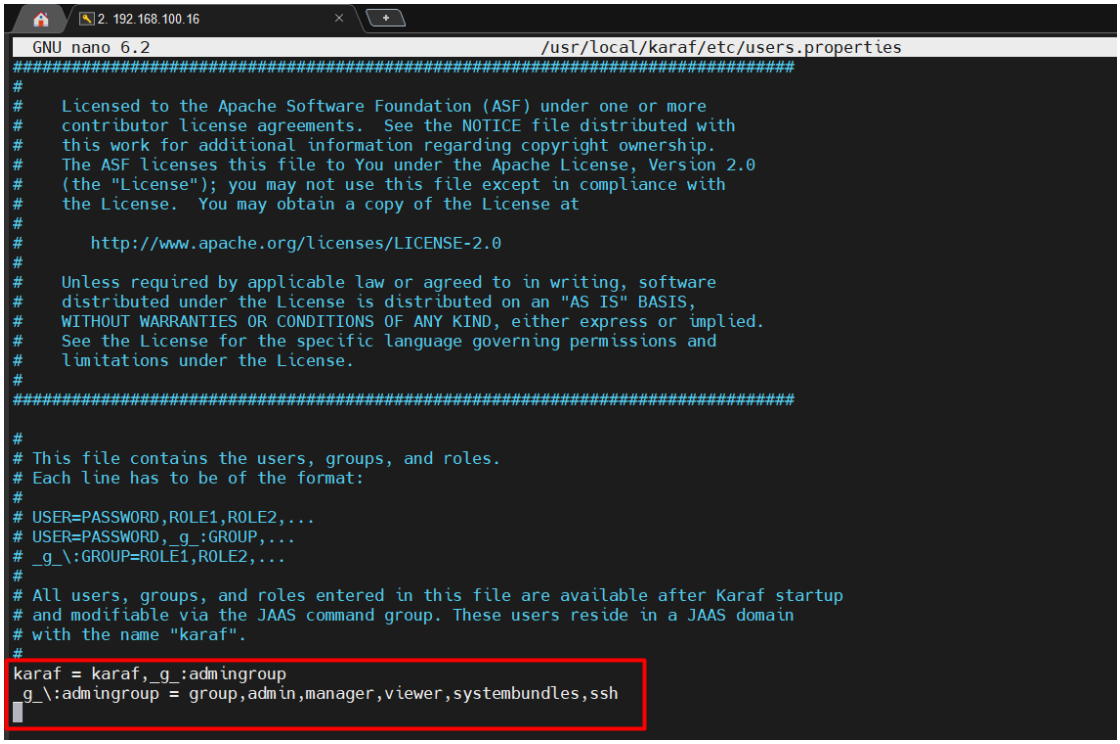


- **Verificar usuarios**

1. A continuación, es necesario establecer los permisos de acceso para usuarios dentro de la configuración del archivo definido dentro de la configuración de karaf. Para ello se debe abrir al archivo correspondiente con el comando: `nano /usr/local/karaf/etc/users.properties`.
2. Una vez dentro del archivo de configuración, se debe acceder al apartado que se encuentra sin comentar, se pueden seguir las indicaciones dentro del archivo considerando, el usuario y contraseña, que por defecto es “karaf” en ambos campos. Adicionalmente se debe definir un grupo y los roles como: admin, manager, viewer, entre otros que determinan las acciones que tiene permitido realizar el usuario. La Figura 63 muestra el contenido del archivo.

Figura 63.

Configuración del archivo de usuarios en Karaf

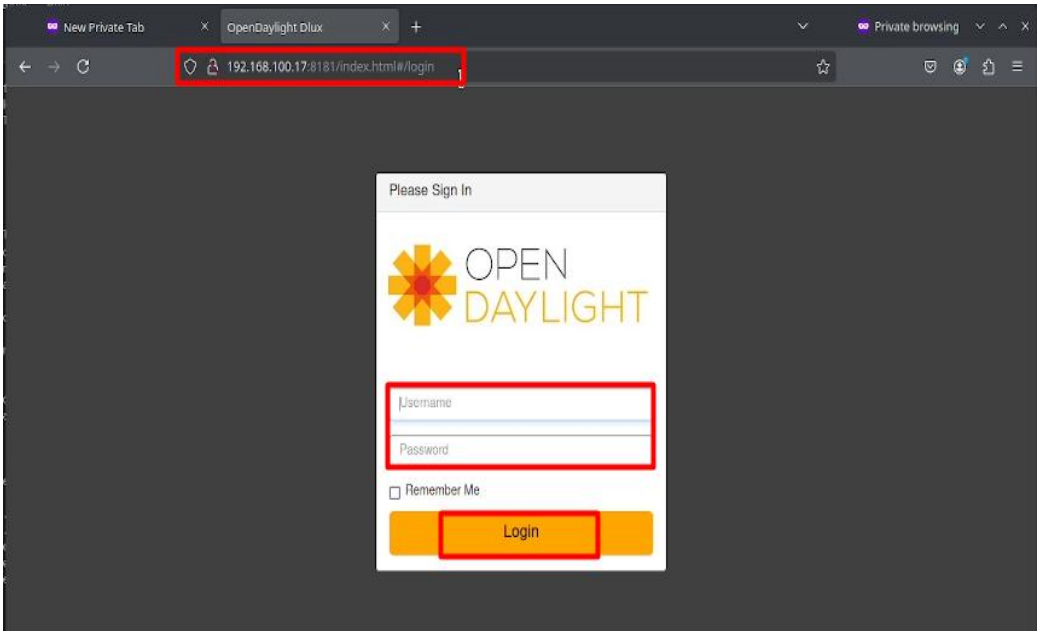


- **Acceso mediante interfaz gráfica.**

1. Finalmente, se debe comprobar el acceso a la interfaz gráfica mediante un navegador web, para ello se debe considerar que un host dentro de la misma red interna acceda haciendo uso de la dirección ip del sistema que contiene al controlador OpenDaylight con la siguiente dirección en el buscador: `http://<dirección ip del servidor>:8181/index.html`. La Figura 64 muestra la pantalla de autenticación en la que se deben ingresar las credenciales configuradas anteriormente.

Figura 64.

Acceso a la interfaz gráfica de OpenDaylight en DLUX



f) Configuración de dispositivos de red

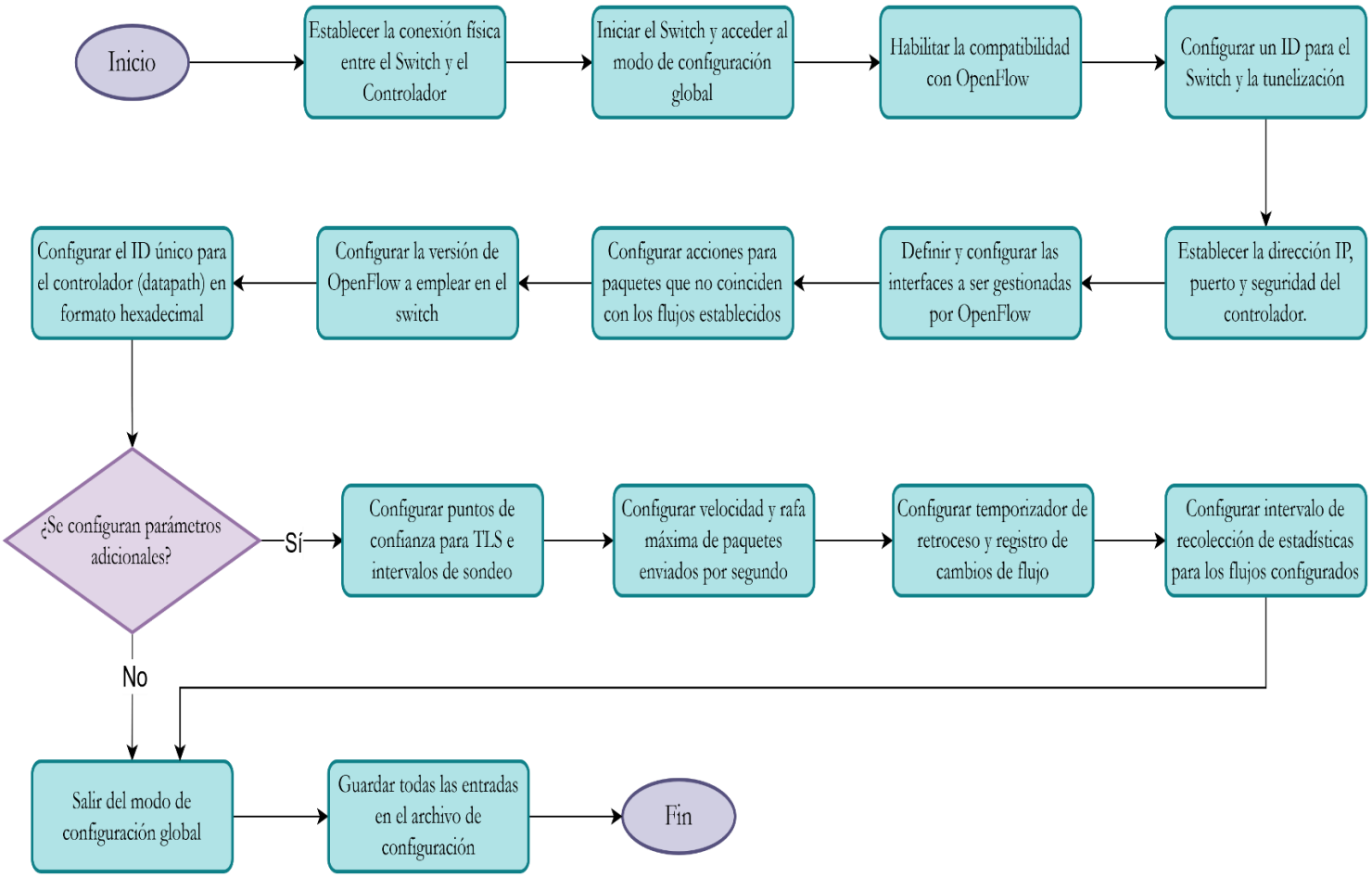
La configuración de dispositivos de red puede variar en base a las necesidades emergentes dentro de la red, pues estas determinan el comportamiento de los equipos en el proceso de encaminamiento y reenvío de paquetes de un punto a otro, por lo cual, dentro de un entorno de red definida por software la asociación de los dispositivos de red es el primer paso por considerar para de este modo se pueda contar con una administración centralizada y así gestionar el tráfico de la mejor forma.

El proceso de asociación de dispositivos al controlador SDN utilizando el protocolo OpenFlow es fundamental para garantizar una administración centralizada. Cisco proporciona una guía específica para habilitar OpenFlow en sus dispositivos compatibles, cuyas configuraciones se describen en el **Anexo 2**.

Una vez que todos los dispositivos de red sean compatibles con el protocolo OpenFlow, se puede proceder con el proceso de configuración inicial. Esto incluye la habilitación del protocolo en cada dispositivo, siguiendo las configuraciones recomendadas por el fabricante. La Figura 65, presenta un resumen del proceso para la configuración de OpenFlow en dispositivos Cisco.

Figura 65.

Proceso de habilitación y configuración de OpenFlow en dispositivos de red Cisco



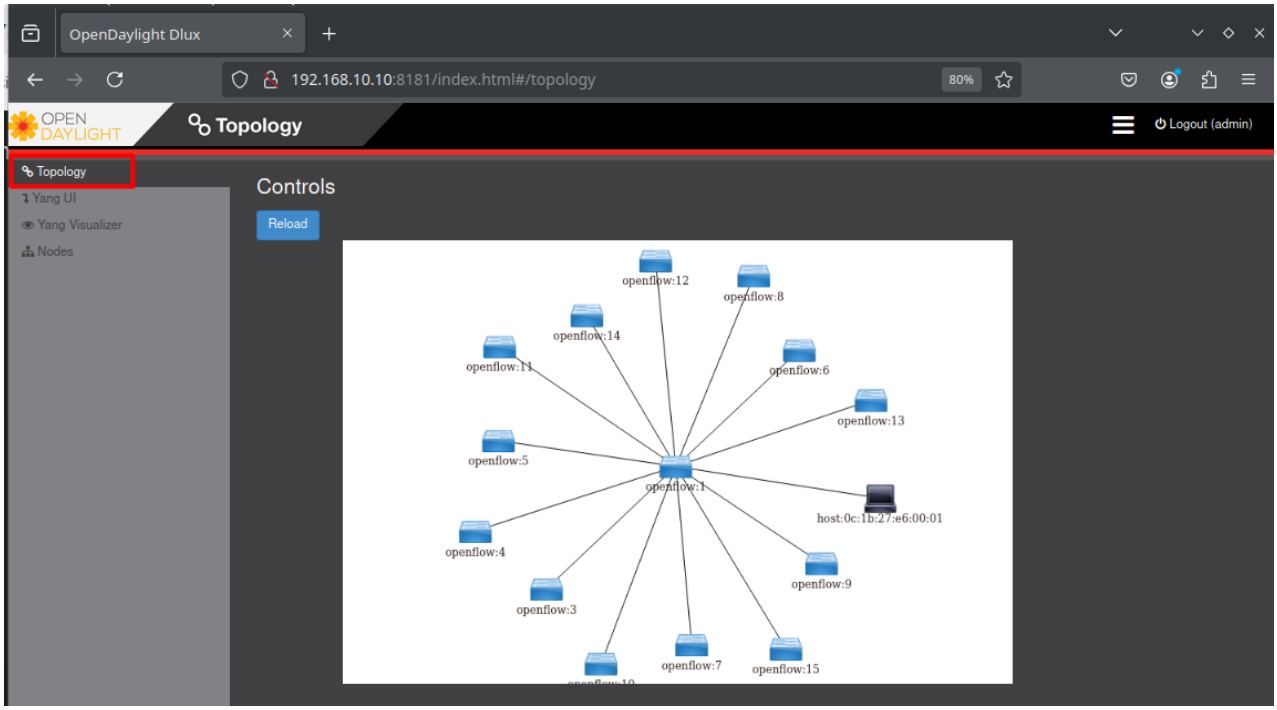
Tras configurar OpenFlow en los dispositivos, se debe verificar que el controlador haya reconocido correctamente cada equipo y sus datapaths para

facilitar la creación de reglas de flujo. Además, es necesario que el controlador pueda reconocer la topología de red para de este modo mejorar la administración y la visibilidad, para ello se requiere que desde la consola de OpenDaylight se agreguen módulos que permitan la detección y actualización dinámica de los dispositivos y sus enlaces para que así pueda detectar posibles bucles físicos que comprometan al rendimiento de la red. La instalación de estos módulos se realiza mediante el comando: `feature:install odl-openflowplugin-app-topology-manager topology-lldp-discovery`.

Para verificar que todos los dispositivos se encuentren vinculados ya al controlador y que la distribución de la red es la correcta, se accede a la interfaz gráfica a través de un navegador como se indicó en la Figura 64 y se accede al apartado de *Topology*, aquí se muestra la topología actual de red como es detectada por el controlador, se debe verificar que esta distribución sea correcta previo a realizar más configuraciones para evitar fallos en el funcionamiento de la SDN, dicha figura muestra un ejemplo para una red centralizada en el core.

Figura 66.

Visualización de la topología de red en la interfaz gráfica de OpenDaylight



Una vez verificada la topología y reconocidos cada uno de los dispositivos es posible establecer reglas de flujo asociadas a cada uno de los switches, para ello es posible establecerlas de forma manual desde la terminal del sistema operativo que aloja al servidor mediante el comando `curl` que establece solicitudes a la REST API de OpenDaylight. Sin embargo, al contar con varios dispositivos por configurar es más adecuada la automatización de la configuración para manejar de manera eficiente los dispositivos y evitar errores humanos. Por esta razón, se platea el desarrollo de scripts en Python que aprovechen la REST API del controlador para implementar reglas de flujo de forma programada

En la arquitectura de red actual se usan VLANs para segmentar y diferenciar el tráfico, mas esto implica configuraciones manuales en cada

uno de los dispositivos, por lo que, al considerar que en una SDN esta segmentación puede ser reemplazada por flujos de red que se configuran desde el controlador eliminando errores manuales y facilitando la gestión de redes grandes o dinámicas mientras mantiene el enfoque de segmentación tradicional. Además de permitir la generación de reglas más específicas que las VLANs tradicionales. Por esta razón, la configuración manual de VLANs se reemplaza por la creación de flujos mediante el script de Python mostrado en el **Anexo 3**. La composición de este script se describe a continuación:

- **Importación de módulos:** en primer lugar, es necesario considerar todos los módulos necesarios para la configuración, para ello se debe tener en cuenta que la REST API trabaja con solicitudes HTTP, por lo cual las librerías se deben importar de forma previa, así mismo se debe considerar el formato que en este caso es JSON, y el manejo de tiempo que es necesario para inducir delays en los procesos que se realizarán más adelante. El Bloque de código 1 corresponde al apartado de importación.

Bloque de código 1:*Importación de módulos de Python para el script*

```
import requests
import json
import time
```

- **Configuración de acceso al controlador:** en el Bloque de código 2 se especifica la configuración del controlador OpenDaylight, se incluye: su dirección IP, puerto de la REST API y las credenciales de acceso (usuario y contraseña). Esta información es esencial para poder enviar configuraciones hacia los switches mediante la REST API del controlador.

Bloque de código 2: *Definición de credenciales del controlador*

```
# Configuración de credenciales de acceso hacia el
controlador.

controller_ip = "192.168.10.10"
controller_port = "8181"
username = "admin"
password = "admin"
```

- **Definición de subredes para IPv4 e IPv6:** a modo de arreglo se definen todas las subredes asociadas a cada una de las VLANs de la red tradicional, para ello se les asigna un identificador y la dirección de subred tanto en ipv4 como en ipv6. De este modo es posible crear reglas de flujo para cada uno de los segmentos de red manteniendo la distribución lógica en dual stack que tiene la red tradicional. El Bloque de código 3 muestra la definición de arreglo, todas las redes deben definirse como se muestra para garantizar que la regla de flujo se cree correctamente caso contrario será omitida. Dicho bloque de código contiene un ejemplo de la configuración requerida para cada subred, el listado completo de todas las subredes se detalla en el **Anexo 3.**

Bloque de código 3*: Definición de subredes a considerar para la creación de flujos de red*

```
# Definición de las subredes definidas para cada una de
las vlans existentes en la red en IPv4 e IPv6.

vlans = [
    {"id": 1, "subnet_ipv4": "172.16.1.0/24",
"subnet_ipv6": "2801:10:6800:1::/64"},
    {"id": 2, "subnet_ipv4": "10.24.8.0/24",
"subnet_ipv6": "2801:10:6800:1024::/64"},
    {"id": 3, "subnet_ipv4": "172.16.3.0/24",
"subnet_ipv6": "2801:10:6800:3::/64"},
    {"id": 4, "subnet_ipv4": "172.16.4.0/24",
"subnet_ipv6": "2801:10:6800:4::/64"},
    {"id": 6, "subnet_ipv4": "172.16.6.0/24",
"subnet_ipv6": "2801:10:6800:6::/64"},
    {"id": 7, "subnet_ipv4": "172.16.7.0/24",
"subnet_ipv6": "2801:10:6800:7::/64"},
```

```
    {"id": 8, "subnet_ipv4": "172.16.8.0/22",
"subnet_ipv6": "2801:10:6800:8::/64"},
    {"id": 12, "subnet_ipv4": "172.16.12.0/24",
"subnet_ipv6": "2801:10:6800:12::/64"},
]
```

- **Creación de una función para la definición de reglas de flujo:** una vez definidas todas las subredes a considerar se procede a definir una función que genere y envíe las reglas de flujo que admitan que dispositivos de la misma subred tengan comunicación, para esta función se deben considerar los parámetros de entrada necesarios, estos incluyen el id del switch, el id de cada subred definido anteriormente, la subred y en este caso un indicador booleano para trabajar con IPv4 e IPv6, a continuación se define el identificador de la regla de flujo que debe ser único para cada una y finalmente determina el tipo de tráfico a considerar en esta regla de flujo. El Bloque de código 4 muestra la definición de estos parámetros

Bloque de código 4.

Definición de parámetros de entrada para la función de creacion de flujos

```
# Definición de una función para la creación de una
regla de flujo para tráfico dentro de la misma subred
basado en IPv4 o IPv6.

def create_intra_vlan_flow(switch_id, vlan_id, subnet,
is_ipv6=False): # Definición de variables de entrada
para la función

    flow_id = f"intra_vlan_{vlan_id}_{'ipv6' if is_ipv6
else 'ipv4'}" # Definición de identificador para la
regla de flujo

    eth_type = 34525 if is_ipv6 else 2048 # Definición
del tipo de tráfico (IPv4 o IPv6) en el protocolo
Ethernet
    ip_match_field = "ipv6" if is_ipv6 else "ipv4" #
Definición de los campos de coincidencia (ipv4-
source/Destination o ipv6-source/Destination)
```

A continuación, se construye la regla de flujo siguiendo el formato JSON que maneja la REST API de OpenDaylight, para ello se debe considerar, un identificador de la regla, los criterios para identificar el tráfico que coincida con la regla (match), las acciones que deben tomarse para los paquetes, la prioridad de la regla y el identificador de la tabla a la que pertenecerá la regla de flujo. El Bloque de código 5 muestra la regla de flujo en el formato indicado para enviarse a los dispositivos especificados.

Bloque de código 5: *Regla de flujo en formato JSON*

```
flow_data = {
        "flow": [
            {
                "id": flow_id, #Asiganción de
identificador del flujo
                "match": { #Definición de criterios de
coincudencia
                    "ethernet-match": {
                        "ethernet-type": {
                            "type": eth_type
#Coincidencia basada en tipo de ethernet
                        }
                    },
                    f"{ip_match_field}-source": subnet,
                    f"{ip_match_field}-destination":
subnet #Coincidencia basada en que IP de origen y
destino deben pertenecer a la misma subred
                },
                "instructions": {
                    "instruction": [
                        {
                            "order": 0,
                            "apply-actions": {
#Designación de acciones a aplicar
                                "action": [
                                    {
                                        "order": 0,
                                        "output-action":
{
                                            "output-
node-connector": "NORMAL" #Procesamiento normal del
paquetes como salida de la regla
                                        }
                                    }
                                ]
                            }
```

```
                        }
                    ]
                },
                "priority": 300, #Prioridad de la regla
                "table_id": 0 #ID de la tabla de flujo
donde se aplicará la regla
            }
        ]
    }
```

Posteriormente se construye la URL que enviará la regla hacia los dispositivos incluyendo en ella la información del controlador que ya se ha definido, la ruta de configuración de los nosotros definida por la REST API, el identificador del switch, la tabla de flujo y el id de la regla. A continuación, se define el contenido en su formato y se hace uso del método HTTP PUT para enviar la regla al controlador y finalmente, se maneja un código de respuesta que se imprimirá en consola acorde al código de respuesta que se reciba por parte del controlador, esto se evidencia en el Bloque de código 6.

Bloque de código 6: *Creación de la URL y envío de regla*

```
#Creación de la URL
url =
f"http://{controller_ip}:{controller_port}/restconf/conf
ig/opendaylight-
inventory:nodes/node/{switch_id}/table/0/flow/{flow_id}"
#Creación de la URL

headers = {'Content-Type': 'application/json'}
#Definición del contenido en formato JSON

response = requests.put(url, data=json.dumps(flow_data),
headers=headers, auth=(username, password)) #Envío de la
regla al controlador

#Definición de mensajes de respuesta

    if response.status_code == 200 or
response.status_code == 201:
        print(f"Flow {flow_id} successfully created on
{switch_id}") #Respuestas exitosas
    else:
        print(f"Error creating flow {flow_id} on
{switch_id}: {response.status_code} - {response.text}")
#Respuesta de error
```

- **Creación de función para automatizar la instalación de reglas:** a continuación, se define una función que permita aplicar las reglas en todos los switches especificados en el arreglo *"switches"*, y en cada uno de estos se cumple la iteración del ciclo *for* que define la creación de una regla de flujo para cada id definido tanto para su red IPv4 como para IPv6 llamando a la función definida anteriormente con un retardo de 1 segundo entre la creación de cada uno de los flujos para evitar que los switches se sobrecarguen y se cree un error en el envío de las reglas de flujo, este valor puede variar acorde a la capacidad de los dispositivos, entre mejor procesamiento tengan estos menor será el retardo que se deba asignar. El Bloque de código 7 indica el contenido de la función.

Bloque de código 7: *Creación de función de automatización de configuración*

```
#Definir los switches en base al identificador que tiene
el controlador: openflow:<datapath>


    switches = ["openflow:1", "openflow:22",
"openflow:3", "openflow:4", "openflow:5","openflow:6",
"openflow:7", "openflow:8", "openflow:9", "openflow:10",
"openflow:11", "openflow:12", "openflow:13",
"openflow:14", "openflow:15"]

    for switch in switches: # Iteración para pasar por
cada switch definido en el arreglo

        for vlan in vlans: # Iteración para pasar por
cada subred definida en el arreglo

            create_intra_vlan_flow(switch, vlan["id"],
vlan["subnet_ipv4"], is_ipv6=False) # Crear regla de
flujo para IPv4

            time.sleep(1)  # Delay para evitar
sobrecarga

            create_intra_vlan_flow(switch, vlan["id"],
vlan["subnet_ipv6"], is_ipv6=True) # Crear regla de
flujo para IPv6
```

```
            time.sleep(1)  # Delay para evitar
sobrecarga
```

- **Ejecución del script:** finalmente, el script ejecuta la función dando inicio al proceso de configuración de flujos de red dando paso a todas las acciones definidas previamente a las reglas de conmutación que admiten el tráfico dentro de la misma subred, y asegurando el tráfico correctamente segmentado. El Bloque de código 8 corresponde a esta acción.

Bloque de código 8: *Ejecución de la función que aplica toda la configuración*

```
# Ejecutar la aplicación de reglas intra-VLAN
apply_intra_vlan_rules()
```

Por otro lado, para el manejo del enrutamiento inter-vlan dentro de la arquitectura de red se ha tomado la decisión de usar un segundo controlador SDN, debido a que OpenDaylight ofrece prestaciones de enrutamiento virtual basada en plataformas en la nube que requieren del uso de virtualización más compleja. Por esta razón se selecciona el controlador RYU que ofrece una aplicación de enrutamiento denominada Rest Router la cual se acopla a la gestión de subredes internas.

Según (Y. Zhang et al., 2018), la implementación de múltiples controladores en una red definida por software ofrece ventajas significativas en términos de escalabilidad, consistencia, confiabilidad y balanceo de carga. El uso de más de un controlador dentro de la red permite distribuir la carga de trabajo y de este modo formar una estructura mucho más robusta para el manejo de la red.

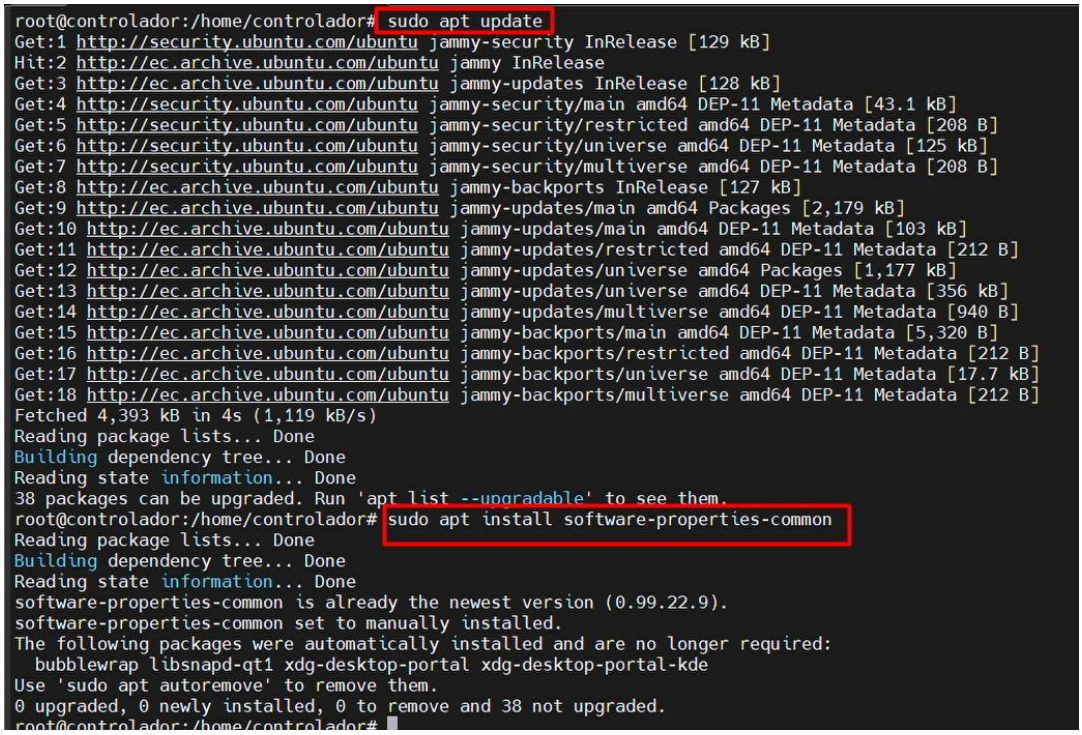
Gracias a la modularidad de OpenDaylight (ODL) es posible integrar de forma sencilla más de un controlador en el mismo sistema operativo haciendo que cada uno opere en puertos diferentes, logrando de este modo mantener el

control centralizado dentro de la red y permitiendo separar funciones entre ambos softwares de control, pues ODL destaca en la gestión del protocolo LLDP, aportando un adecuado descubrimiento de la red y ayudando a prevenir los bucles lógicos dentro de la red, mientras que RYU puede instalar reglas dinámicas y temporales dentro de las tablas de flujo de los conmutadores asociados. Al combinar ambos controladores es posible contar con una gestión más eficiente y escalable para la red.

El proceso de instalación de RYU dentro del sistema operativo en el que se gestiona ODL se realiza a través de Python que es el lenguaje en el que se basa todo el software y sus distintas aplicaciones, para ello se consideran los pasos expuestos a continuación.

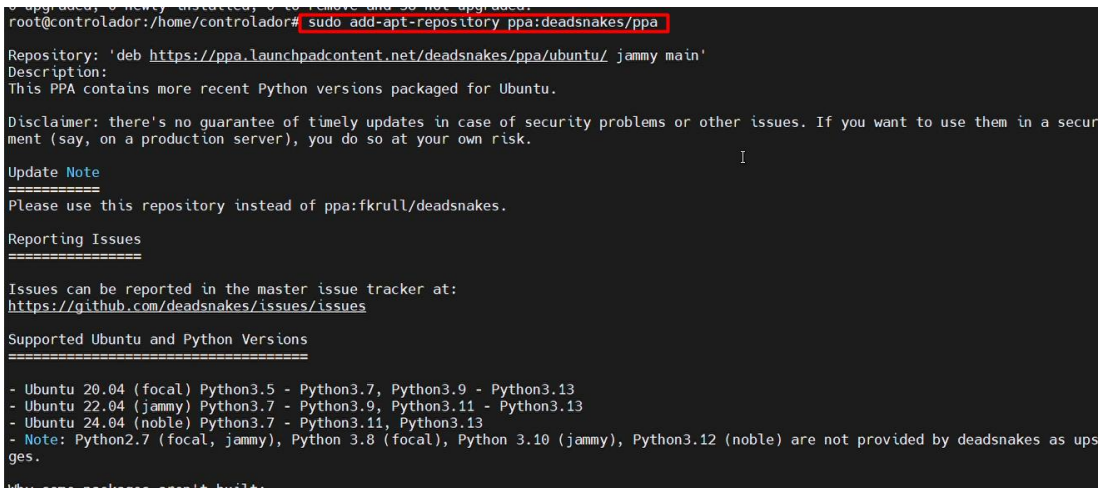
En primer lugar, se debe actualizar las librerías de Ubuntu para de este modo asegurar que se descargarán las versiones más recientes de los paquetes requeridos, para ello se usa el comando `sudo apt update`. Posteriormente se instala una herramienta que permita gestionar repositorios adicionales para volver más sencilla la instalación de softwares que no están dentro de los repositorios predeterminados, esto se realiza mediante el comando `sudo apt install software-properties-common`. La **Figura 67** muestra la ejecución de los comandos descritos anteriormente.

Figura 67.
Actualización de paquetes e instalación de dependencias



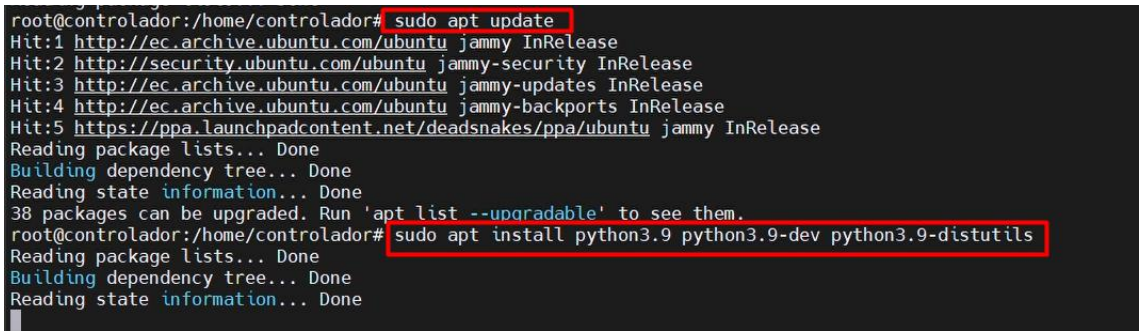
A continuación, se agrega el repositorio deadsnaks que contiene versiones específicas de Python que son necesarias para la instalación del controlador pero que no se hallan disponibles en los repositorios predeterminados de Ubuntu. Se hace uso del comando `sudo add-apt-repository ppa:deadsnakes/ppa`. Este proceso se muestra en la **Figura 68**.

Figura 68.
Agregación de repositorio para versiones específicas de Python



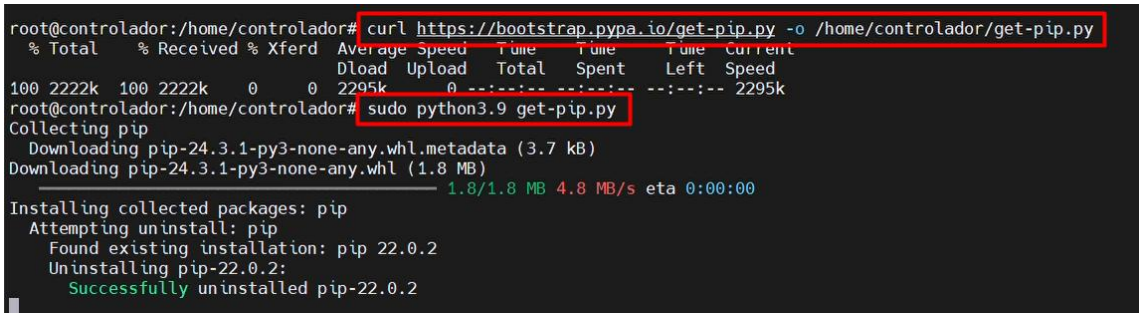
Posterior a añadir el nuevo repositorio, es necesario actualizar la lista de paquetes para que se incluyan los del repositorio añadido anteriormente, para ello se vuelve a ejecutar el comando `sudo apt update` . A continuación se procede con la instalación de Python 3.9 y sus dependencias como los archivos de desarrollo y las herramientas para la gestión y creación de paquetes Python mediante el comando `sudo apt install python3.9 python3.9-dev python3.9-distutils` . La **Figura 69** muestra la ejecución de los comandos antes mencionados

Figura 69.
Instalación de Python y sus dependencias



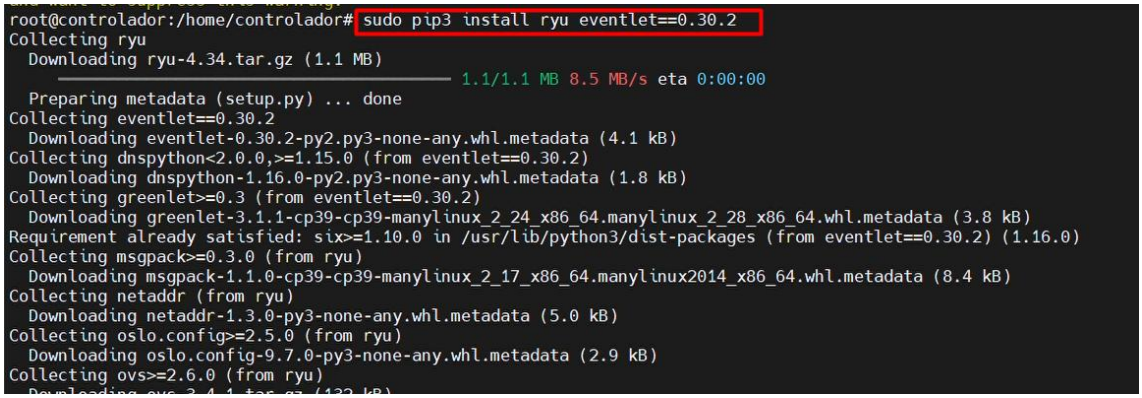
Una vez se cuenta con los requerimientos de Python necesarios, se procese a descargar el instalador de pip mediante el comando `curl https://bootstrap.pypa.io/get-pip.py -o /home/controlador/get-pip.py`, de este modo se obtiene el instalador y se almacena en el directorio `/home/controlador/`. Con el instalador correctamente descargado, se procede a ejecutar el instalador mediante el comando `sudo python3.9 get-pip.py` el cual especifica el uso de Python en su versión 3.9 como se evidencia en la **Figura 70**.

Figura 70.
Descarga y ejecución del instalador de pip



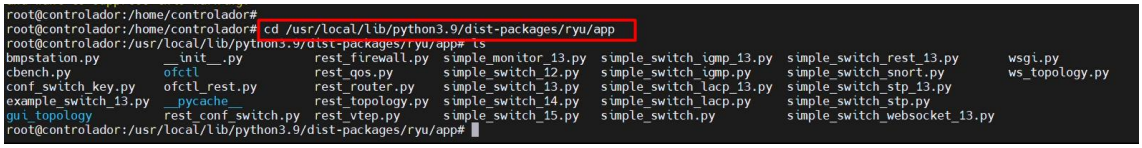
Posteriormente, se realiza la instalación de RYU y las dependencias específicas mediante el comando `sudo pip3 install ryu eventlet==0.30.2`. La **Figura 71** muestra el proceso antes mencionado especificando la descarga mediante pip del framework SDN RYU, y la biblioteca para manejo de concurrencia compatible con el controlador.

Figura 71.
Instalación de RYU y sus dependencias



A continuación, se accede al directorio de aplicaciones de RYU con el comando `cd /usr/local/lib/python3.9/dist-packages/ryu/app`, en este repositorio es donde se deben ejecutar el controlador y las aplicaciones deseadas. La **Figura 72** muestra las aplicaciones disponibles para su ejecución.

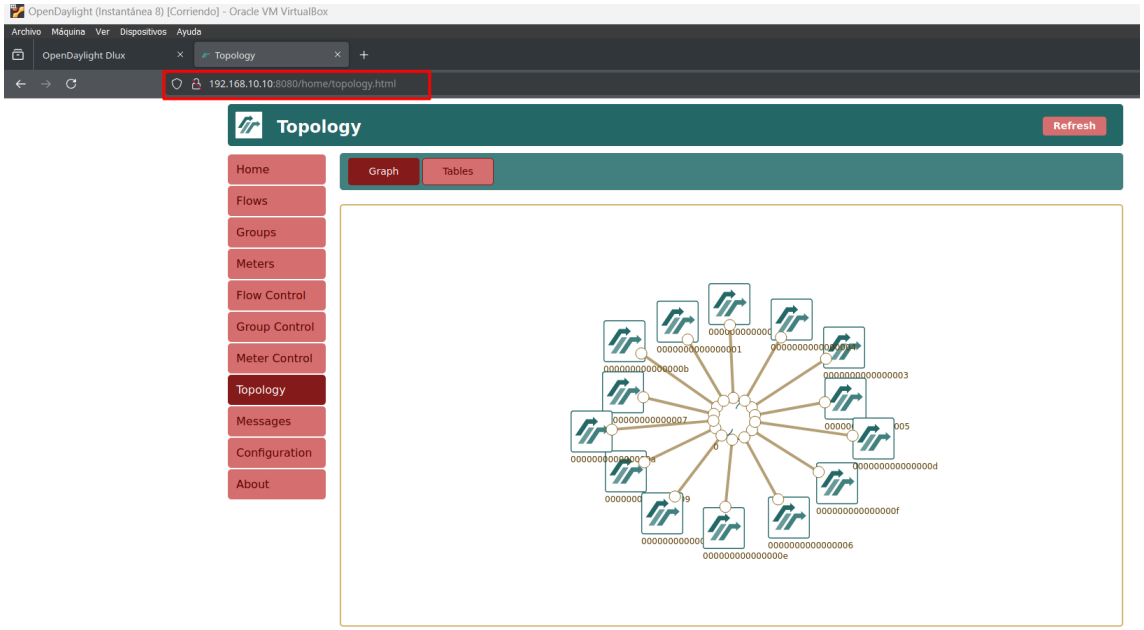
Figura 72.
Listado de aplicaciones disponibles para RYU



Finalmente, como adicional, es posible obtener una interfaz de gestión que permita visualizar de mejor manera la topología de red, las tablas de flujo y los flujos dentro de la red, para ello se debe descargar el repositorio de GitHub de FlowManager dentro de la carpeta apps de RYU mediante el comando `git clone https://github.com/martimy/flowmanager`.

Una vez el controlador se encuentra correctamente instalado, es necesario iniciar su ejecución, para ello se considera el comando `ryu-manager --ofp-tcp-listen-port 6644 flowmanager/flowmanager.py rest_router.py` , el cual define el puerto por el que se van a realizar las conexiones y los proceso de OpenFlow en este caso el puerto 6644 para evitar conflictos con OpenDaylight que está en ejecución simultánea, la inicialización de la interfaz de gestión de FlowManager para acceder mediante un navegador web mediante: `http://<ip del servidor>:8080/home/index.html` y finalmente, la ejecución de la aplicación Rest Router que será empleada para el enrutamiento de las redes internas. La interfaz de administración de FlowManager se puede apreciar en la **Figura 73**.

Figura 73.
Interfaz de FlowManager



Una vez se ejecuta ese comando es posible empezar con la configuración de los switches, en este caso la app de Rest Router hace que los conmutadores puedan adoptar funcionalidades de routers y en base a esto indicarle al controlador rutas para los distintos hosts que se traducen en reglas de flujo que se instalan temporalmente en las tablas de flujo de los conmutadores que se requieren evitando de este modo saturar a todo los conmutadores con reglas innecesarias. Para esto se debe considerar en primer lugar la asignación de gateways para cada subred, en este caso al tratarse de una arquitectura centrada en el core de la red, se definen las direcciones correspondientes en el switch de core. A continuación, dentro del mismo switch/router se deben definir rutas estáticas para las subredes para de este modo garantizar la convergencia de la red. Con la finalidad de automatizar este proceso se creó un script en Python que vuelva más sencilla esta asignación, el script para las redes IPv4 se encuentra completo dentro del **Anexo 4.** La composición del script se describe a continuación:

- **Importación de módulos:** en primera instancia se debe realizar la importación de módulos o bibliotecas de Python, en este caso se considera el módulo requests que permite realizar solicitudes HTTP de forma sencilla a trapes de los métodos propios de HTTP. El Bloque de código 9 muestra ese apartado.

  Bloque de código 9:*Importación de módulos de Python para el script*

```
import requests
```

- **Definición de la lista de gateways IPv4:** a continuación, se define dentro de un arreglo todos los gateways a considerar para el enrutamiento, para ello se deben indicar las direcciones IP junto con las máscaras de red correspondientes. El Bloque de código 10 muestra a modo de ejemplo el arreglo a considerar, el listado completo se define en el Anexo 4.

  Bloque de código 10:*Definición del arreglo de gateways*

```
gateways = [
"172.16.1.1/24", "172.16.4.1/24", "172.16.6.1/24",
"172.16.7.1/24",  "172.16.30.1/24" ]
```

- **Definición de la lista de rutas estáticas IPv4:** posteriormente se deben definir las rutas estáticas a implementar dentro de un arreglo que contenga la red de destino, y el gateway por el cual dicha red será alcanzable, de este modo el router podrá reenviar el tráfico hacia redes específicas. El Bloque de código 11 contiene un arreglo de ejemplo con la definición de algunas rutas estáticas, el arreglo completo se encuentra en el Anexo 4.

  Bloque de código 11: *Definición de arreglo de rutas*

```
routes = [
 {"destination": "172.16.1.0/24", "gateway":
"172.16.1.1"},
 {"destination": "172.16.4.0/24", "gateway":
"172.16.4.1"},
{"destination": "192.168.10.0/24", "gateway":
"192.168.10.1"}
]
```

- **Creación de la función para configurar gateways en el switch de core:** a continuación, se designa una función que envíe las direcciones IP del arreglo gateways hacia el switch correspondiente, para ello se usa un ciclo *for* que itere por cada elemento del arreglo y se envía mediante una solicitud HTTP cambiando el campo *address* en cada iteración. El Bloque de código 12 muestra esta función.

Bloque de código 12: *Definición de la función para la configuración de gateways*

```
def configure_gateways(): #Definición de la función
    print("Configurando gateways en Router 1...") #Impresión
    de mensaje de referencia en la pantalla
    for gw in gateways: #iteración para el arreglo gateways
      #Solicitud para enviar la dirección IP mediante la API
       response = requests.post(
           "http://localhost:8080/router/0000000000000001",
           json={"address": gw}
       )
       #Impresión de la respuesta al método HTTP
       print(f"Gateway     {gw}:     {response.status_code},
    {response.reason}")
```

- **Creación de la función para configurar rutas estáticas:** con todos los valores necesarios definidos y con los gateways ya configurados, se proceden a definir las rutas estáticas recorriendo el arreglo *routes* para enviarlos mediante solicitudes HTTP hacia la REST API y posteriormente imprime la respuesta a dichas solicitudes. El Bloque de código 13 muestra la función indicada.

Bloque de código 13: *Definición de la función para la configuración de rutas estáticas*

```
def configure_routes():
    print("Configurando rutas estáticas en Router 1...")
    for route in routes:
        response = requests.post(

"http://localhost:8080/router/0000000000000001",
            json=route
```

```
        )
        print(f"Ruta    {route}:    {response.status_code},
{response.reason}")
```

- **Bloque principal:** finalmente, se definen los puntos de entrada del script llamando a las funciones creadas anteriormente en orden para garantizar que la ejecución sea adecuada. El Bloque de código 14 muestra esta definición.

  Bloque de código 14: *Definición del bloque principal*

  ```
  if __name__ == "__main__":
  configure_gateways()
  configure_routes()
  ```

Si bien los controladores mencionados son altamente eficientes en la generación de reglas de flujo y en la centralización de la red, presentan limitaciones en el manejo de protocolos de red específicos. Estas limitaciones derivan de que su diseño no incluye la capacidad de gestionar tablas de enrutamiento completas, responder a solicitudes de redirección de paquetes ni calcular rutas dinámicas, funciones esenciales para garantizar el enrutamiento correcto del tráfico IPv6, a diferencia del soporte ya consolidado para IPv4.

En este contexto, los módulos existentes en los controladores SDN carecen de capacidades fundamentales para actuar como routers. Estos módulos dependen exclusivamente de las reglas de flujo generadas por el controlador, lo que significa que no almacenan tablas de enrutamiento ni procesan solicitudes de descubrimiento o cálculo de rutas (Open Daylight, n.d.). Esto limita significativamente su capacidad para gestionar las complejidades del enrutamiento en entornos IPv6, donde protocolos como ICMPv6, NDP (Neighbor Discovery Protocol), la redistribución de rutas y la fragmentación de paquetes son esenciales. Por ello, se deben considerar soluciones complementarias que extiendan las capacidades de los controladores SDN en estos aspectos.

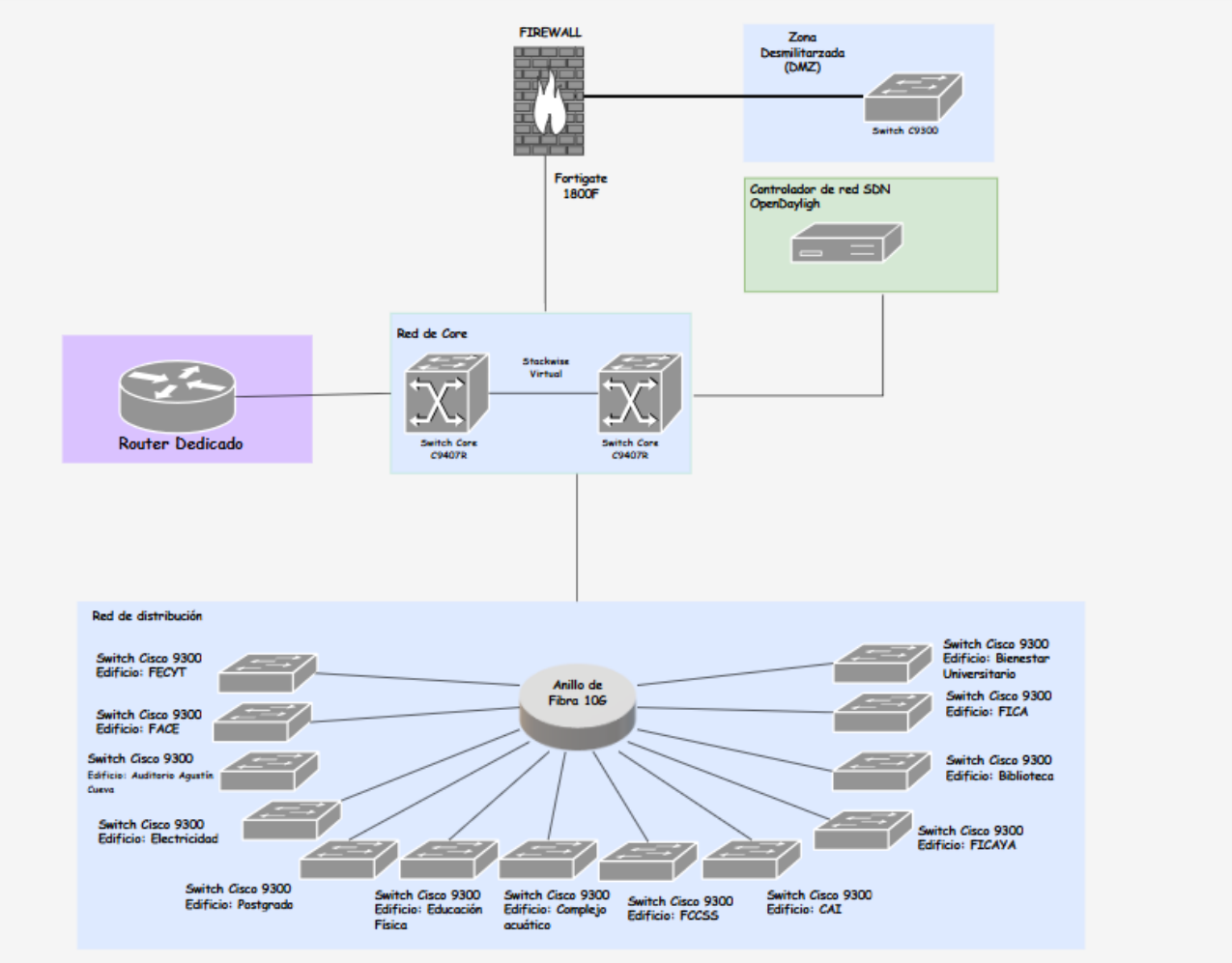
Como medida temporal hasta que se amplíe el soporte de los controladores, se propone la incorporación de un router dedicado para gestionar exclusivamente el enrutamiento. Este dispositivo actuará como un nodo especializado encargado de manejar los gateways de las subredes y garantizar el reenvío correcto de paquetes. Con esta solución, se asegura que las limitaciones actuales de los controladores SDN no interfieran con la continuidad operativa de la red ni con su capacidad para satisfacer las demandas de tráfico IPv6. Además, el router permitirá una integración eficiente con el controlador SDN, proporcionando la información necesaria para que este genere reglas de flujo específicas en los switches, lo que asegura un reenvío de paquetes conforme a las políticas definidas.

La incorporación de un router dedicado no solo resuelve las limitaciones en la gestión del tráfico IPv6, sino que también simplifica la configuración del controlador SDN. Al delegar las funciones de enrutamiento avanzadas al router, el controlador puede centrarse en su objetivo principal: centralizar y gestionar dinámicamente las reglas de flujo a nivel de la capa de enlace de datos. Este enfoque fortalece tanto la escalabilidad como el rendimiento de la red, aspectos esenciales para una arquitectura SDN moderna.

A nivel físico, el router se conectará directamente al switch de core de la red, como se ilustra en la Figura 74. Esta configuración busca minimizar la latencia y garantizar un alto rendimiento en la distribución del tráfico. En esta arquitectura, el router asume la gestión completa del enrutamiento y actúa como un nodo que redistribuye las rutas aprendidas hacia el controlador. Con esta información, el controlador genera reglas de flujo temporales que se implementan

en los switches, asegurando una comunicación eficiente entre dispositivos sin la necesidad de gestionar directamente las tablas de enrutamiento en el controlador.

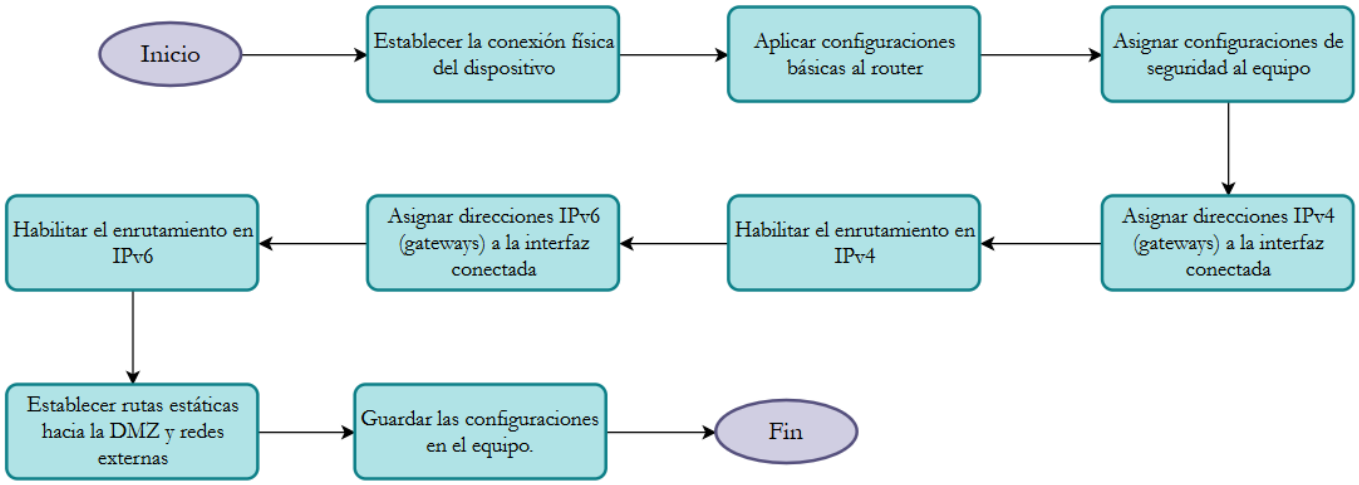
Figura 74.
Arquitectura Híbrida para el enrutamiento



Esta arquitectura híbrida representa una solución robusta para mantener una red dual-stack operativa, preservando la centralización y escalabilidad características de la SDN. Aunque se plantea como una solución temporal hasta que los controladores SDN incluyan soporte completo para IPv6, su implementación garantiza una transición efectiva hacia una infraestructura más avanzada, proporcionando mayor flexibilidad, centralización y capacidad de adaptación ante las demandas futuras de la red.

La configuración específica del dispositivo puede variar dependiendo del fabricante, ya que cada equipo implementa ciertas particularidades en sus comandos y funcionalidades. Sin embargo, es posible establecer un patrón general de configuración que garantice la convergencia adecuada dentro de la red. Este patrón, ilustrado en la *Figura 75*, proporciona una guía básica aplicable a diferentes entornos. En el caso de dispositivos Cisco, se recomienda seguir las configuraciones detalladas en el Anexo 5, las cuales han sido diseñadas para cumplir con los requisitos operativos de esta red.

Figura 75.
Proceso de configuración para el router



En OpenDaylight, las reglas de flujo se crean utilizando la información proporcionada por el router dedicado, aprovechando las tablas de enrutamiento y los datos de resolución de direcciones obtenidos mediante ARP para IPv4 y NDP para IPv6. El router actúa como un nodo que anuncia las rutas de red y responde a solicitudes de resolución de direcciones, permitiendo al controlador SDN generar reglas específicas a nivel de capa MAC. Estas reglas asocian direcciones IP con sus respectivas direcciones MAC y puertos físicos en los switches, asegurando un reenvío preciso y eficiente del tráfico en la red.

La configuración de los dispositivos dentro de la arquitectura propuesta de red definida por software representa un paso crítico para garantizar una integración eficiente entre los elementos físicos y el controlador SDN. A través de los procesos descritos, se establecen los cimientos para una red centralizada y escalable, donde los dispositivos no solo responden a configuraciones específicas, sino que trabajan en conjunto bajo un modelo de administración centralizada lo cual refuerza la capacidad de la red para adaptarse a demandas futuras, manteniendo la flexibilidad y el rendimiento que caracterizan a las soluciones basadas en SDN. La configuración planteada no solo solventa las necesidades actuales, sino que también prepara la infraestructura para evolucionar hacia tecnologías más avanzadas.

4.4 Presupuesto estimado para la infraestructura SDN

La migración hacia una arquitectura de red definida por software requiere ajustes en la infraestructura existente, lo que conlleva una inversión económica considerable. Esta inversión incluye la adquisición de nuevos equipos, licencias y componentes esenciales para garantizar una transición exitosa y el funcionamiento eficiente de la nueva arquitectura de red.

Tomando como referencia los dispositivos detallados anteriormente, se presenta en este apartado un presupuesto estimado para los componentes necesarios. Los costos varían en función de los dispositivos seleccionados, por lo que se incluyen precios aproximados correspondientes al año 2024 para ofrecer una perspectiva actualizada.

Para solventar la necesidad de infraestructura para el controlador de red se consideraron como opciones tres servidores de alto rendimiento descritos en el apartado **Descripción de recursos de hardware por adquirir**, dentro de la **Tabla**

29 se muestran los valores de mercado para cada uno de los modelos mencionados con anterioridad.

Tabla 29.

Costo de las opciones de servidores para el controlador de red.

Modelo del equipo	Costo total
HPE ProLiant DL380 Gen10	$ 4,303.30
Dell PowerEdge R740	$ 4,153.80
Cisco UCS C220 M6	$ 5 576.99

Para sustituir los dispositivos de red no compatibles con la nueva tecnología a emplear se presentaron opciones del fabricante cisco, mismas que forman parte de la seria Catalyst que actualmente predominan en la arquitectura tradicional, la **Tabla 30** detalla los costos de los dispositivos. Además, se debe considerar que las licencias requeridas deben ser: Universal (Network Essentials) o Universal (Network Advantage), pues son las que incluyen la compatibilidad con el protocolo OpenFlow 1.3. Estas licencias se incluyen el precio de los dispositivos ya que se compran en conjunto con cada uno de los dispositivos, los precios referenciales se obtuvieron mediante el análisis en sitios de comparación especializados como itprice (https://itprice.com/) que recopilan información de distintos sitios web asociados a Cisco Partners autorizados.

Tabla 30.

Costo de las opciones de switches para la red de distribución

Modelo del equipo	Costo unitario	Tipo de licencia	Cantidad de switches requerida	Costo total
Cisco Catalyst 9300	$11,974.90	Network Advantage	8	$95,799.2
Cisco Catalyst 9300	$11,945.20	Network Essentials.	8	$95,561.6
Cisco Catalyst 9300X	$14,017.40	Network Advantage/ Network Essentials	8	$112,139.2
Cisco Catalyst 9400	$23,104.11	Network Advantage/ Network Essentials	8	$184,832.88
Cisco Catalyst 9500	$28,637.65	Network Advantage/ Network essentials	8	$229,101.2

Se debe destacar que, existen muchos factores que pueden incrementar o disminuir el precio de los dispositivos. El tipo de interfaces de conexión, el número de puertos y otras especificaciones pueden influir directamente en los costos. En este caso, los valores se han calculado considerando switches de 48 puertos, siguiendo el

esquema de la arquitectura actual. Además, es importante tener en cuenta que los precios oficiales no son publicados por el fabricante directamente, sino que son obtenidos de sitios de compra autorizados y de sitios de comparación de precios especializados.

La selección del modelo final debe basarse en el presupuesto asignado y las necesidades específicas de la red. Modelos como los switches Cisco Catalyst 9300, 9300X y 9500 representan soluciones fijas, mientras que el Catalyst 9400 ofrece una opción modular con mayor escalabilidad, pero con una inversión inicial más alta, ya que tomando como base el número de ranuras disponibles en el chasis se requieren de más tarjetas de línea lo cual incrementa el coste final del dispositivo.

La integración de un router dedicado en la arquitectura híbrida propuesta requiere considerar modelos que se ajusten a las necesidades específicas de la red institucional. Estos equipos deben ser capaces de manejar el enrutamiento de tráfico IPv4 e IPv6, resolver las solicitudes de capa 3 y proporcionar la información necesaria para que el controlador SDN pueda generar reglas de flujo específicas. La **Tabla 31** muestra los precios referenciales para los equipos propuestos para cumplir dicho rol.

Tabla 31.

Costo de las opciones de routers para la arquitectura híbrida

Modelo del equipo	**Costo total**
CISCO ISR 4451-X	$ 26,935.00
Juniper MX204	$ 12,152.05
MikroTik CCR2004-16G-2S+	$ 465.00

Los valores presentados en este apartado proporcionan una guía aproximada de los costos que implica la migración a la arquitectura de redes definidas por software para la red actual del data center de la UTN, sin embargo la suma de estos valores en la combinación aceptada no abarca un presupuesto final, pues existen factores como las variaciones en costos de mercado de los dispositivos, las especificaciones finales que se deseen en cada dispositivo y otros gastos adicionales asociados directamente a la conexión física, entre estos se pueden encontrar elementos como conectores, herramientas de instalación, cables y demás insumos necesarios que no se han detallado en las tablas pasadas. Con esta base se determina que todos los componentes deben ajustarse a las necesidades operativas de la red para garantizar una implementación sostenible de la nueva arquitectura.

Conclusiones Y Recomendaciones

Conclusiones

El desarrollo de una guía metodológica para la migración del datacenter de la Universidad Técnica del Norte a una arquitectura de redes definidas por software proporciona una herramienta clave en la inclusión de tecnologías modernas. Este proyecto permite optimizar el uso de recursos existentes dentro del datacenter además de promover la inclusión de soluciones avanzadas orientadas a la mejora de la administración y escalabilidad de la red para adaptarse a las nuevas demandas de la red.

El análisis detallado de los conceptos fundamentales de las redes definidas por software, junto con el estudio de las tecnologías y estándares aplicables, permitió construir un marco conceptual sólido que sustenta las decisiones tomadas a lo largo de este trabajo. Este enfoque facilitó la identificación de las mejores alternativas tecnológicas para la migración de la arquitectura de red tradicional hacia una infraestructura definida por software.

La evaluación de la infraestructura de la red actual permitió identificar las limitaciones de la arquitectura tradicional, como la necesidad constante de revisiones de configuración de forma manual y la dependencia de elementos aislados como el procesamiento individual de los dispositivos. Este análisis fundamentó la propuesta de realizar una migración a una nueva arquitectura SDN, que mediante el control centralizado simplifica la gestión de la red y mejora la capacidad de respuesta y adaptación.

La propuesta de implementación de una nueva arquitectura de redes definidas por software, basada en soluciones de código abierto, permitió identificar y establecer requerimientos clave tanto de hardware como de software esenciales para garantizar el

correcto funcionamiento de la red bajo el protocolo OpenFlow. La evaluación de equipos compatibles aseguró la integración de dispositivos modernos que permitan una gestión eficiente y centralizada de la red ofreciendo mayos escalabilidad que permita adaptar la red a nuevas demandas tecnológicas.

Las reglas de flujo establecidas mediante el protocolo OpenFlow constituyen el núcleo funcional de la arquitectura de redes definidas por software, ya que permiten una gestión precisa y dinámica del tráfico de red. Al implementar reglas específicas basadas en información de la red que se actualiza periódicamente, se garantiza una comunicación eficiente y segmentada, eliminando la necesidad de configuraciones manuales repetitivas lo que mejora la escalabilidad y flexibilidad de la red simplificando la administración.

Las limitaciones actuales de las redes definidas por software para procesos como el enrutamiento directo de IPv6 no representan un obstáculo insuperable para la migración de la red, pues la implementación de una solución híbrida, que integra un router dedicado para manejar el enrutamiento de IPv6 y complementar las capacidades de los controladores SDN, representa una respuesta viable dichas limitaciones. Esta estrategia asegura la continuidad operativa de una red dual-stack mientras mantiene la centralización, escalabilidad y flexibilidad propias de SDN, ofreciendo una transición eficiente hacia una arquitectura más avanzada.

La guía metodológica desarrollada constituye un recurso estratégico y práctico para el DDTI de la UTN, ofreciendo una orientación estructurada y precisa para el proceso de migración hacia una arquitectura de redes definidas por software. Su diseño abarca desde la evaluación inicial de la infraestructura existente hasta la implementación de configuraciones específicas, permitiendo una transición planificada y eficiente. Además, su enfoque detallado asegura que tanto los aspectos técnicos como operativos sean

abordados de manera comprensible, sentando las bases para una administración centralizada y adaptable a las necesidades tecnológicas futuras de la institución.

Recomendaciones

Se recomienda que el DDTI de la Universidad Técnica del Norte adopte la guía metodológica diseñada en esta investigación como recurso principal para planificar y ejecutar la migración hacia redes definidas por software. Este documento, desarrollado específicamente para las necesidades de la red institucional, ofrece una estructura clara y detallada que reduce los riesgos asociados al proceso, optimiza el aprovechamiento de los recursos actuales y facilita una transición ordenada.

Para garantizar la operatividad eficiente de la nueva arquitectura, se recomienda implementar programas de formación especializados en tecnologías SDN para el personal encargado de la administración de la red. Estos programas deben abordar la gestión del controlador OpenDaylight, el protocolo OpenFlow y la configuración de reglas de flujo, proporcionando los conocimientos necesarios para operar y mantener la red. Además, se sugiere complementar esta capacitación con una actualización continua en tecnologías emergentes para facilitar una toma de decisiones más informada y efectiva.

Se recomienda priorizar la adquisición de dispositivos de red compatibles con el protocolo OpenFlow y servidores robustos capaces de alojar controladores como OpenDaylight. Estos equipos deben seleccionarse considerando su capacidad para manejar reglas de flujo de manera eficiente, además de ofrecer escalabilidad y flexibilidad que permitan adaptarse a las crecientes demandas tecnológicas y al crecimiento continuo de la red institucional.

Dado que las reglas de flujo constituyen el núcleo operativo de las redes SDN, se recomienda implementar un sistema de monitoreo continuo que utilice protocolos como

LLDP, ARP y NDP para optimizar la detección y gestión de la red. Este monitoreo permitirá identificar y resolver rápidamente posibles ineficiencias, ajustando las configuraciones de manera dinámica para garantizar un rendimiento óptimo y una segmentación eficiente del tráfico.

Como solución temporal a las limitaciones actuales en el manejo de IPv6 en los controladores SDN, se recomienda incorporar un router físico especializado que gestione funciones críticas de enrutamiento y proporcione al controlador la información necesaria para crear reglas de flujo específicas. Paralelamente, se sugiere investigar y evaluar las actualizaciones de controladores y protocolos que permitan integrar de forma nativa el enrutamiento IPv6 en la arquitectura SDN, mejorando el rendimiento y optimizando los recursos disponibles.

Para incrementar las capacidades de la red SDN, se recomienda implementar módulos adicionales en OpenDaylight, enfocados en seguridad y manejo avanzado del tráfico. Estos módulos permitirán una mejor gestión de la red institucional y responderán eficientemente a sus necesidades cambiantes además de integrar nuevas funcionalidades que a largo plazo permitan visualizar resultados óptimos para los grandes flujos de tráfico que se procesan dentro de la red.

Se recomienda mantener un registro detallado de todas las configuraciones implementadas, tanto en el controlador como en los dispositivos de red, para facilitar futuras expansiones o adaptaciones. Además, se sugiere revisar y actualizar periódicamente la guía metodológica, incorporando nuevas herramientas y prácticas emergentes de las redes definidas por software enfocados a llevar una actualización constante en las innovaciones presentadas para mejorar el rendimiento de la red implementada en la UTN.

Referencias Bibliográficas

Alberti, A. M. (2012). *Software-Defined Networking: Perspectives, Requirements, and Challenges*. https://www.researchgate.net/publication/236156555

Alcatel-Lucent. (2009). *Alcatel-Lucent 7705 Manual (Page 2 of 164) | ManualsLib*. https://www.manualslib.com/manual/2534396/Alcatel-Lucent-7705.html?page=2#manual

almacen-informtico.com. (n.d.). *Cisco Catalyst 9300 - C9300-48P-A : Almacen Informatico*. Retrieved March 30, 2024, from https://www.almacen-informatico.com/cisco_catalyst-9300-c9300-48p-a_3510962_p.htm

Awati, R. (2023, November). *data plane*. https://www.techtarget.com/searchnetworking/definition/data-plane-DP

BasuMallick, C. (2022, February 10). *Top 10 Software-Defined Networking (SDN) Solutions in 2022 - Spiceworks*. https://www.spiceworks.com/tech/networking/articles/best-sdn-solutions/

Bernal, I., & Mejía, D. (2016). Las Redes Definidas por Software y los Desarrollos Sobre Esta Temática en la Escuela Politécnica Nacional. *Revista Politécnica* .

BRAINCORP TECNOLOGIA E INFORMATICA S.A. (n.d.). *Cisco Switch Catalyst 9200 48-port PoE+ 4x10G uplink Switch C9200L-48P-4X-A*. Retrieved March 30, 2024, from https://braincorp.com.ve/producto/cisco-switch-catalyst-9200-48-port-poe-4x10g-uplink-switch-c9200l-48p-4x-a/

Cabaj, K., Wytrębowicz, J., Kukliński, S., Radziszewski, P., & Dinh, K. T. (2014). SDN Architecture Impact on Network Security. *Position Papers of the 2014 Federated*

Conference on Computer Science and Information Systems, *3*, 143–148. https://doi.org/10.15439/2014f473

CCNA desde Cero. (2017). *Convenciones de Nomenclatura Cisco IOS - CCNA desde Cero*. https://ccnadesdecero.es/convenciones-nomenclatura-cisco-ios/#1_Trenes_y_familias_de_versiones_del_software_IOS_de_Cisco

Cisco. (n.d.). *OpenFlow*. http://www.cisco.com/go/cfn.

Cisco. (2008). *Cisco Catalyst 2960-S and 2960 Series Switches with LAN Base Software Enhanced Network Security, Availability, and Manageability for Small and Medium-Sized Businesses*.

Cisco. (2009). *catalyst-3750E-datasheet*.

Cisco. (2016, July 25). *Consolidated Platform Configuration Guide, Cisco IOS Release 15.2(5)E (Catalyst 2960-X Switches) - OpenFlow [Cisco Catalyst 2960-X Series Switches] - Cisco*. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-2_5_e/configuration_guide/b_1525e_consolidated_2960x_cg/b_1525e_consolidated_2960x_cg_chapter_01001101.html#id_33232

Cisco. (2017). *Cisco Catalyst 5500 Controller*.

Cisco. (2018). *Cisco Catalyst 3850 Series Switches*.

Cisco. (2019). *Cisco Catalyst 9400 Series Switch Data sheet Cisco public*.

Cisco. (2020a). *Cisco Catalyst 9200 Series Switches*.

Cisco. (2020b). *Cisco Catalyst 9800 Controller*.

Cisco. (2020c). *Cisco HyperFlex Fabric Interconnect Hardware Live Migration Guide*.

Cisco. (2020d, July 31). *High Availability Configuration Guide, Cisco IOS XE Amsterdam 17.3.x (Catalyst 9400 Switches) - Configuring Cisco StackWise Virtual [Support] - Cisco*. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9400/software/release/17-3/configuration_guide/ha/b_173_ha_9400_cg/configuring_cisco_stackwise_virtual.html

Cisco. (2021a). *Cisco HyperFlex HX220c M5, HX220c M5 All Flash, and HX220c M5 All NVMe Nodes Data Sheet*.

Cisco. (2021b). *OpenFlow*. http://www.cisco.com/go/cfn.

Cisco. (2021c, June 24). *Conozca OpenFlow en los switches Catalyst serie 9000 - Cisco*. https://www.cisco.com/c/es_mx/support/docs/switches/catalyst-9300-series-switches/217210-understand-openflow-on-catalyst-9000-ser.html#anc3

Cisco. (2024a). *Cisco Catalyst 9300 Series Switches*.

Cisco. (2024b, June 6). *Cisco Application Centric Infrastructure (ACI) Design Guide - Cisco*. https://www.cisco.com/c/en/us/td/docs/dcn/whitepapers/cisco-application-centric-infrastructure-design-guide.html

Cisco. (2024c, June 10). *Versiones recomendadas para las plataformas Catalyst 9200/9300/9400/9500/9600 - Cisco*. https://www.cisco.com/c/es_mx/support/docs/switches/catalyst-9300-series-switches/214814-recommended-releases-for-catalyst-9200-9.html

codilime. (n.d.). *What is an SDN controller*. Retrieved December 29, 2023, from https://codilime.com/glossary/sdn-controller/

datacenter360. (n.d.). *Switch Cisco Catalyst 2960-Plus 48PST-S (48PST-S) - Data Center 360*. Retrieved March 30, 2024, from https://datacenter360.net/producto/networking/switch-cisco-catalyst-2960-plus-48pst-s-48pst-s/

DC Parts. (n.d.). *Gabinete HP BladeSystem Platinum c7000 - DC Parts*. Retrieved April 1, 2024, from https://dcparts.lat/producto/caja-hp-bladesystem-platinum-c7000/

De Peña Núñez, G. (2023). *Migración del Core de red a SDN Cisco ACI*. Universitat Oberta de Catalunya.

DIGITALIFE. (n.d.). *Servidor HP Proliant Dl360 Gen9 Rack 1U Xeon E5-2650 V4 32GB DDR4 sin Disco No Os - Digitalife eShop*. Retrieved April 1, 2024, from https://www.digitalife.com.mx/productos/servidor-hp-proliant-dl360-gen9-rack-1u-xeon-e5-2650-v4-32gb-ddr4-sin-disco-no-os

ds3comunicaciones.com. (n.d.). *WS-C2960-48TC-L Switch Administrable capa L2 48 puertos 10/100, 02 puertos 10G fibra SFP LAN Base image Cisco Catalyst 2960S*. Retrieved March 30, 2024, from https://ds3comunicaciones.com/cisco/WS-C2960-48TC-L.html

Eftimie, A., & Borcoci, E. (2020). SDN controller implementation using OpenDaylight: Experiments. *2020 13th International Conference on Communications, COMM 2020 - Proceedings*, 477–481. https://doi.org/10.1109/COMM48946.2020.9142044

García Centeno, A., Manuel Rodríguez Vergel, C., Anías Calderón, C., & Camilo Casmartiño Bondarenko, F. (2014). Controladores SDN, elementos para su selección y evaluación. *Revista Telem@tica*, *13*(3), 10–20. http://revistatelematica.cujae.edu.cu/index.php/tele

Gartner Peer Insights. (n.d.). *Top Cisco Meraki Dashboard Likes & Dislikes 2024 | Gartner Peer Insights*. Retrieved June 30, 2024, from https://www.gartner.com/reviews/market/enterprise-wired-wireless-lan-access-infrastructure/vendor/cisco/product/cisco-meraki-dashboard/likes-dislikes

Give 1 Life. (n.d.). *Chassis HP E-C7000 G3 + 16 X Blades BL460C Gen9 E5-2620 V3 512GB RAM 192CO/384TH + 2X HP 10GbE Pass-Thru*. Retrieved April 1, 2024, from https://www.give1life.com/dell-m1000e-v1-1-blade-16-x-m630-512gb-ram-192co-384th-2x-m8428-k-copia/

Goransson, P., Black, C., & Culver, T. (2017). *Software Defined Networks: A Comprehensive Approach*.

Göransson, P., & Black, Chuck. (2014). *Software defined networks : a comprehensive approach* (E. Steve, H. Kaitling, & G. Punithavathy, Eds.). Elsevier Inc.

Haleplidis, E., Pentikousis, K., Denazis, S., Hadi Salim, J., Meyer, J., & Koufopavlou, O. (2015). *RFC:7426 Software-Defined Networking (SDN): Layers and Architecture Terminology* (7426).

Hashemi-Pou, C. (2023). *What is the CIA Triad? | Definition from TechTarget*. https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA

Hata, H. (2013). A study of requirements for SDN switch platform. *ISPACS 2013 - 2013 International Symposium on Intelligent Signal Processing and Communication Systems*, 79–84. https://doi.org/10.1109/ISPACS.2013.6704525

Hend Abdelgader, E., Kenz A., B., & Hadel, Y. (2019). Software Defined Networking. *International Conference on Sciences and Techniques of Automatic Control & Computer Engineering (STA)*.

Hoang, D. B., & Pham, M. (2015, November 20). On software-defined networking and the design of SDN controllers. *2015 International Conference on the Network of the Future, NOF 2015*. https://doi.org/10.1109/NOF.2015.7333307

HP. (2014). *SERVIDOR HPE PROLIANT DL360 GEN9 - SERVIDORES HP SERVIDORES HP*. https://servidoreshp.com.mx/servidor-hpe-proliant-dl360-gen9/

Huang, T., Yu, F. R., Zhang, C., Liu, J., Zhang, J., & Liu, Y. (2017). A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. *IEEE Communications Surveys and Tutorials*, *19*(2), 891–917. https://doi.org/10.1109/COMST.2016.2630047

Huitema, C. (2015, September 9). *What is Cisco's SDN Strategy for the Data Center?* https://blogs.cisco.com/datacenter/dc-sdn-strategy

Icy Science. (2023). *¿Qué es un dispositivo heredado? - definición de techopedia*. https://es.theastrologypage.com/legacy-device

intercompras.com. (n.d.-a). *Controlador Inalámbrico Cisco 5508, Dispositivo Administrador de Red, 12 MAPs, 8 puertos Ethernet, Fast Ethernet - AIR-C*. Retrieved March 30, 2024, from https://intercompras.com/p/controlador-inalambrico-cisco-dispositivo-administrador-red-maps-54331

intercompras.com. (n.d.-b). *Switch Cisco Catalyst 2960, 24 puertos 10/100 + 2 puertos SFP Combo - WS-C2960-24TC-L*. Retrieved March 30, 2024, from

https://intercompras.com/p/switch-cisco-catalyst-puertos-puertos-sfp-combo-lan-base-35193

ISO/IEC. (2023). *ISO/IEC 25010:2023 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model*. https://www.iso.org/standard/78176.html

IT Corporation. (n.d.). *Cisco Catalyst 9800-L (Cobre) Controladora inalámbrica*. Retrieved March 30, 2024, from https://web.itclatam.com/producto/cisco-catalyst-9800-l-cobre-controladora-inalambrica/

IT Planet. (n.d.). *Cisco Wireless Controller 5508 comprar a precios económicos*. Retrieved March 30, 2024, from https://it-planet.com/es/cisco/cisco-wireless-controller-5508.html

JMTelcom. (n.d.). *Firewall Fortigate 1800F | JMTelcom*. Retrieved March 30, 2024, from https://www.jmtelcom.com/product/firewall-fortigate-1800f/

Juniper. (n.d.). *Understanding Support for OpenFlow on Devices Running Junos OS | Junos OS | Juniper Networks*. Retrieved April 28, 2024, from https://www.juniper.net/documentation/us/en/software/junos/sdn-openflow/topics/concept/junos-sdn-openflow-support-overview.html

Kant, K. (2009). Data center evolution. A tutorial on state of the art, issues, and challenges. *Computer Networks*, *53*(17), 2939–2965. https://doi.org/10.1016/j.comnet.2009.10.004

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey.

Proceedings of the IEEE, *103*(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

Kumar, S. N., Poladi, P. K., Thirupathi, V., Sandeep, C., Kumar, S. N., Pramod Kumar, P., & Assistant, S. (2019). A COMPREHENSIVE REVIEW ON SDN ARCHITECTURE, APPLICATIONS AND MAJOR BENIFITS OF SDN. *International Journal of Advanced Science and Technology*, *28*(20), 607–614. https://www.researchgate.net/publication/339771419

Malwarebytes. (2024, January 17). *What is Firmware? | Firmware Definition and Examples*. https://www.malwarebytes.com/cybersecurity/computer/what-is-firmware

MercadoIT. (n.d.). *Porque comprar Switch Catalyst Cisco 3850 | MercadoIT*. Retrieved March 30, 2024, from https://www.mercadoit.com/blog/analisis-opinion-it/porque-comprar-switch-catalyst-cisco-3850/

Montez, M. (2006). *A new technological innovation, HP BladeSystem c-class*.

mosupervs.live. (n.d.). *Cisco HX220c-M5 HyperFlex - Online Store*. Retrieved April 1, 2024, from https://mosupervs.live/product_details/1198806.html

Nadeau, T. D., & Gray, K. (2013). *SDN: Software Defined Networks* (L. Mike & B. Meghan, Eds.; First).

Nokia. (n.d.). *7705 Service Aggregation Router | Nokia*. Retrieved March 18, 2024, from https://www.nokia.com/networks/ip-networks/7705-service-aggregation-router/

Nokia. (2023). *Nokia 7705 Service Aggregation Router*.

OceanTech. (2023). *Cisco Catalyst 3750-E WS-C3750E-48PD-S V03 48 Port Gigabit Ethernet Switch | eBay*. https://www.ebay.com/itm/125927594690

Open Daylight. (n.d.). *Overview — BGPCEP master documentation*. Retrieved December 6, 2024, from https://docs.opendaylight.org/projects/bgpcep/en/latest/bgp/bgp-user-guide-overview.html#bgp-in-sdn

Open Netowrking Foundation. (2014). *SDN Migration Considerations and Use Cases ONF Solution Brief*. www.opennetworking.org

Open Network Foundation. (2013). *OpenFlow Switch Specification*. http://www.opennetworking.org

Open Networking Foundation. (2015). *Framework for SDN: Scope and Requirements*. www.opennetworking.org

Operé, M. (2016, September 13). *Indicadores de evaluación cualitativos y cuantitativos | Capital Humano*. https://grupo-pya.com/indicadores-de-evaluacion-cualitativos-y-cuantitativos/

Ormeño Rojas, N. F. (2019, May 15). *ISO 25010 y el desarrollo de software | by Nicolás F. Ormeño Rojas | Medium*. https://normeno.medium.com/iso-25010-y-el-desarrollo-de-software-112393a4b341

Paliwal, M., Shrimankar, D., & Tembhurne, O. (2018). Controllers in SDN: A Review Report. *IEEE Access*, *6*, 36256–36270. https://doi.org/10.1109/ACCESS.2018.2846236

Plesant, N. (2023, February 27). *Redes definidas por software (SDN): Por qué su organización lo necesita | Digi International*. https://es.digi.com/blog/post/software-defined-networking

Portal ISO 25000. (n.d.). *ISO/IEC 25010*. Retrieved June 9, 2024, from https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

PST Colombia. (n.d.). *Servidor HP ProLiant DL360 Gen9 SATA / SAS – SFF*. Retrieved April 1, 2024, from https://servidoresalmacenamientoredes.com/home/13-servidor-hp-proliant-dl360-gen9-sata-sas-sff.html

Raza, M. H., Sivakumar, S. C., Nafarieh, A., & Robertson, B. (2014). A Comparison of Software Defined Network (SDN) Implementation Strategies. *Procedia Computer Science*, *32*, 1050–1055. https://doi.org/10.1016/J.PROCS.2014.05.532

RenewTech. (n.d.). *CISCO WS-C3750E-48TD-S - Cisco CAT 3750E 48 10/100/1000+2 10GEX2 265W IPB*. Retrieved March 18, 2024, from https://www.renewtech.es/cisco-ws-c3750e-48td-s.html

Rodríguez Herlein, D. R., Talay, C. A., González, C. N., & Marrone, L. A. (n.d.). *Explorando las redes definidas por software (SDN)*.

Roger. (2022, June 24). *Redes Definidas por Software (SDN): Tipos, Ventajas y Aplicaciones*. https://community.fs.com/es/user/112

Rosencrance, R., English, J., & Brunke, J. (2022, May). *software-defined networking (SDN)*. https://www.techtarget.com/searchnetworking/definition/software-defined-networking-SDN

Rouse, M. (2012, May 7). *Dual Stack Network*. Dual Stack Network

Router-Switch.com. (n.d.-a). *C9300-48T-E - Cisco Switch Catalyst 9300*. Retrieved March 30, 2024, from https://www.router-switch.com/c9300-48t-e.html

Router-Switch.com. (n.d.-b). *C9407R - Cisco Switch Catalyst 9400*. Retrieved March 30, 2024, from https://www.router-switch.com/c9407r.html

Salman, O., Elhajj, I. H., Kayssi, A., & Chehab, A. (2016). SDN controllers: A comparative study. *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, 1–6. https://doi.org/10.1109/MELCON.2016.7495430

Sanchez, D. (2022, July 5). *¿QUÉ ES EL BACKPLANE EN SWITCH ETHERNET? - MPPS VS GBPS*. https://info.ita.tech/blog/que-es-backplane-switch-ethernet-mpps-vs-gbps#:~:text=%C2%BFQu%C3%A9%20significa%20%22%20Mpps%20%22%20en,los%20Switches%20%2F%20Conmutadores%20de%20paquetes.

Schaller, S., & Hood, D. (2017). Software defined networking architecture standardization. *Computer Standards & Interfaces*, *54*, 197–202. https://doi.org/10.1016/J.CSI.2017.01.005

Segmentify. (2023, May 17). *Application Programming Interface (API)*. https://segmentify.com/glossaries/api/

Stallings, W. (2014). *Data and Computer Communications* (10 th). http://www.pearsonhighered.com/stallings/

Switch-Wifi. (n.d.). *Cisco Catalyst 9800-L (Fibra Optica) Controladora inalámbrica – Switch-Wifi*. Retrieved March 30, 2024, from https://switch-wifi.com/producto/cisco-catalyst-9800-l/

Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras* (5th ed.). Pearson Educación.

Technologies Inc, F. (2021). *FortiGate 1800F Series Data Sheet*.

Ting Wang, Zhiyang Su, Yu Xia, & Hamdi, M. (2014). Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing. *IEEE Access*, *2*, 1481–1496. https://doi.org/10.1109/ACCESS.2014.2383439

Tonitrus. (n.d.). *Cisco - WS-C3750E-48PD-S - Catalyst 3750E 48 10/100/1000 PoE+2*10GE(X2),750W,IPB s/w*. Retrieved March 18, 2024, from https://www.tonitrus.com/es/redes/cisco/switch/cisco-catalyst-3750-e-switch/10104165-003-cisco-ws-c3750e-48pd-s-catalyst-3750e-48-10/100/1000-poe-2-10ge-x2-750w-ipb-s/w/

TrustRadius. (n.d.). *Page 3 of 5 Pros and Cons of Cisco Meraki MX 2024*. Retrieved July 7, 2024, from https://www.trustradius.com/products/cisco-meraki-mx/reviews?qs=pros-and-cons&f=50#overview

VMware. (2020, April 27). *Redes definidas por software*. https://www.vmware.com/es/topics/glossary/content/software-defined-networking.html

Zhang, H., Tang, F., & Barolli, L. (2019). Efficient flow detection and scheduling for SDN-based big data centers. *Journal of Ambient Intelligence and Humanized Computing*, *10*(5), 1915–1926. https://doi.org/10.1007/S12652-018-0783-6/TABLES/2

Zhang, Y., Cui, L., Wang, W., & Zhang, Y. (2018). A survey on software defined networking with multiple controllers. *Journal of Network and Computer Applications*, *103*, 101–118. https://doi.org/10.1016/j.jnca.2017.11.015

Anexos

Anexo 1: Instalación de sistema operativo Ubuntu Server 22.04

Objetivo:

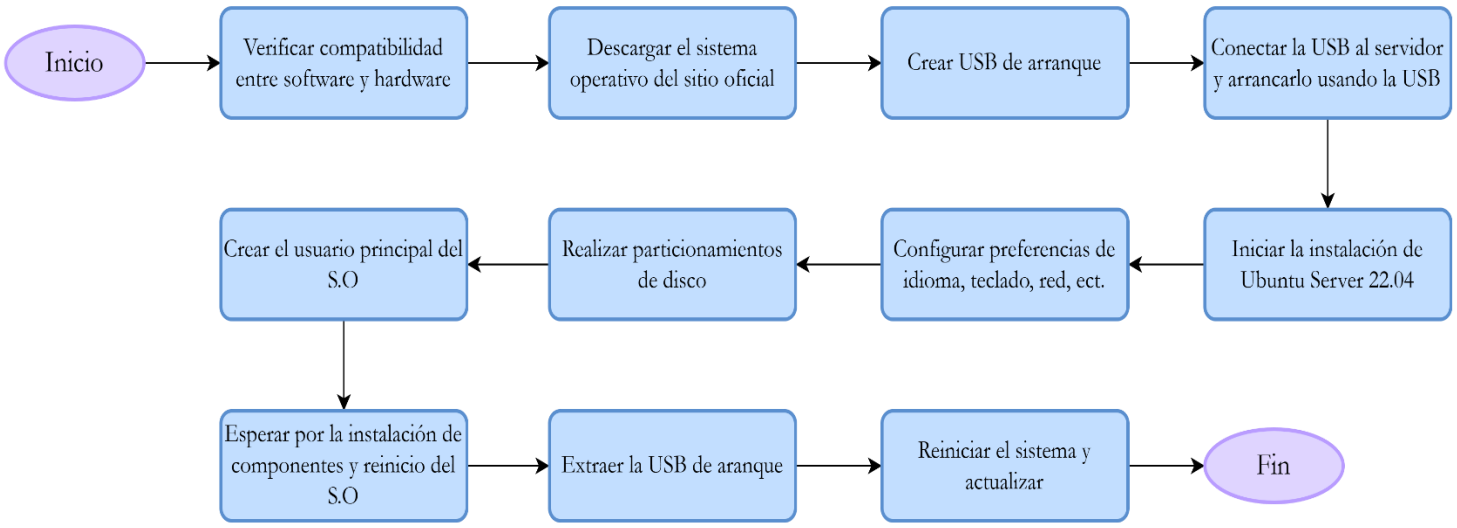
- El propósito de este anexo es proporcionar una guía detallada del proceso de instalación del sistema Operativo Ubuntu Server 22.04 como parte de la configuración del controlador de red OpenDaylight para la migración de la arquitectura de red tradicional a una red definida por software.

Ubuntu Server 22.04 es una distribución de Linux empleada en entornos de servidores debido a la estabilidad que ofrece para el desarrollo de aplicaciones de distintas índoles, el soporte extendido para la actualización e integración de distintos paquetes y la compatibilidad con tecnologías emergentes. Su operación como sistema operativo base dentro de dispositivos como servidores permite que los administradores cuenten con un sistema versátil para la instalación de softwares de monitoreo o control para la red.

La instalación de un sistema operativo dentro de un servidor es necesario para la gestión de recursos y servicios, por lo que, en base al propósito del componente de hardware se debe optar por el sistema operativo óptimo. La Figura 1 describe el procedimiento a realizar.

Figura 1.

Diagrama de flujo del proceso de instalación de Ubuntu Server 22.04 en un servidor



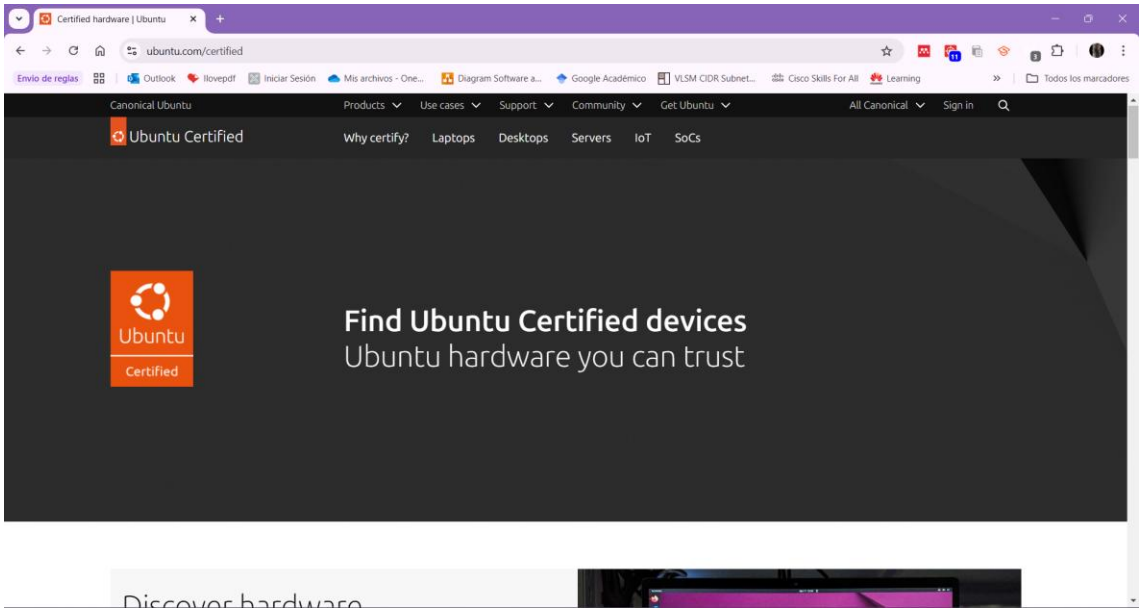
Tomando como referencia el proceso descrito en la Figura 1, para instalar el sistema operativo Ubuntu Server 22.04, se pueden seguir los pasos descritos a continuación:

- **Verificación de compatibilidad entre software y hardware.**

  1. Previamente a instalar un sistema operativo en cualquier hardware, es importante conocer si estos son compatibles para de este modo garantizar que todos los componentes del equipo vayan a ser reconocidos por el sistema operativo. Para ello se debe consultar la documentación oficial del fabricante del dispositivo y validar en el sitio **Ubuntu Certified (https://ubuntu.com/certified).** El sitio web oficial se muestra en la Figura 2.

Figura 2.

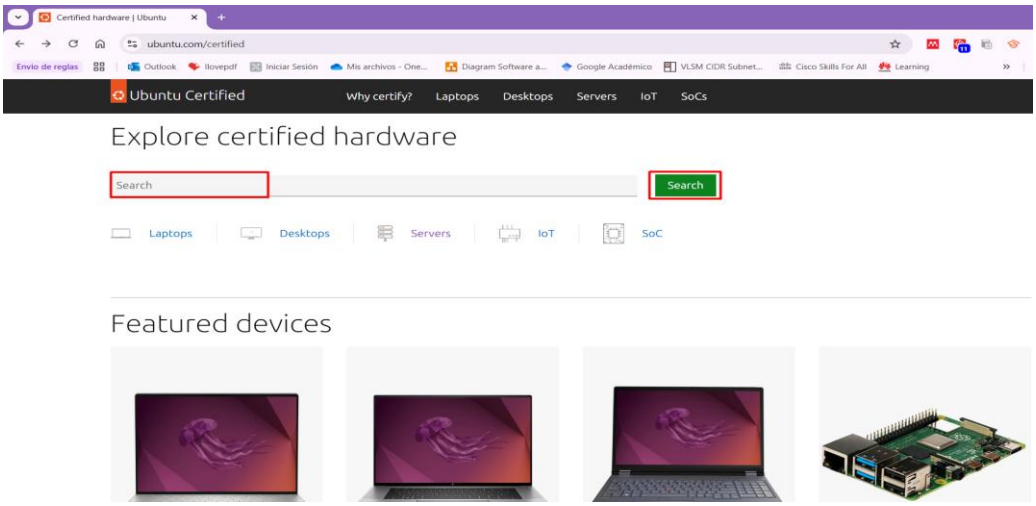
Vista inicial del sitio Ubuntu Certified



2. Para comprobar la compatibilidad, se debe realizar la búsqueda del dispositivo sobre el que se desea instalar el sistema operativo y realizar la búsqueda con el botón Search, este desplegará todos las coincidencias encontradas en distintas categorías. La barra de búsqueda se muestra en la Figura 3.

Figura 3.

Barra de búsqueda de dispositivos compatibles con sistemas operativos Ubuntu.

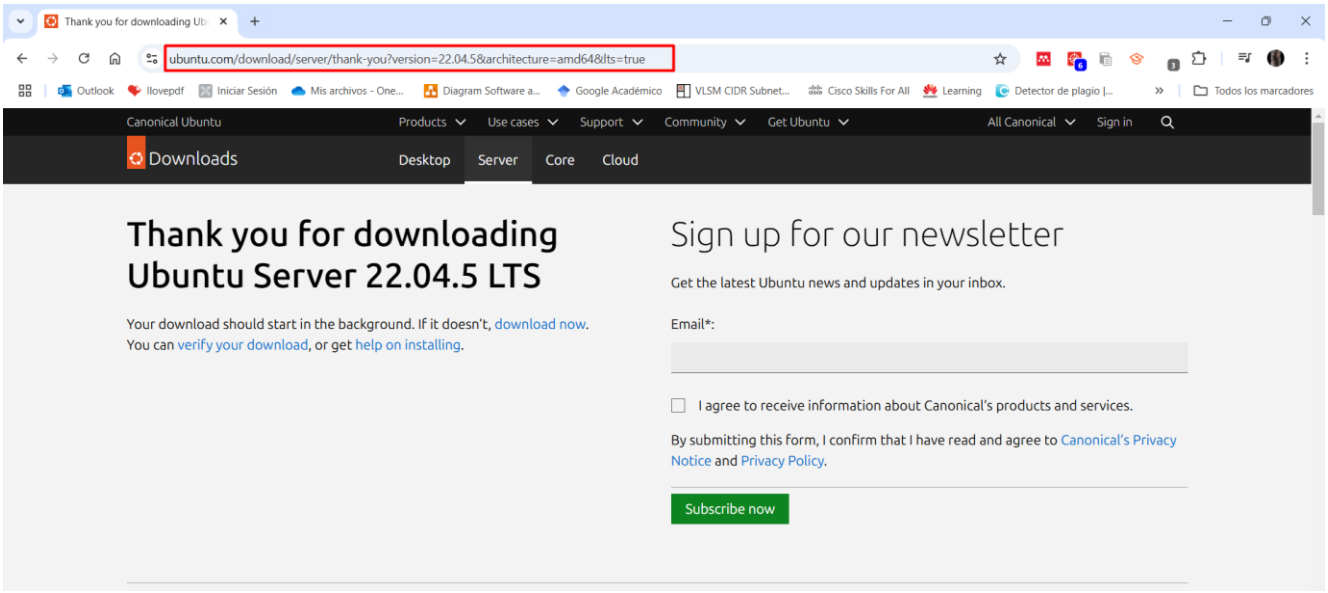


- **Descarga del sistema operativo y creación de la unidad de arranque.**

1. Posterior a la revisión de las versiones compatibles con el servidor seleccionado con la instalación, se debe buscar la versión específica, en este caso “Ubuntu Server 22.04” y acceder al sitio oficial para la descarga: https://ubuntu.com/download/server/thank-you?version=22.04.5&architecture=amd64&lts=true. La Figura 4 muestra sitio oficial que despliega la descarga de la versión indicada.

Figura 4.

Sitio oficial de Ubuntu para la descarga del sistema operativo.



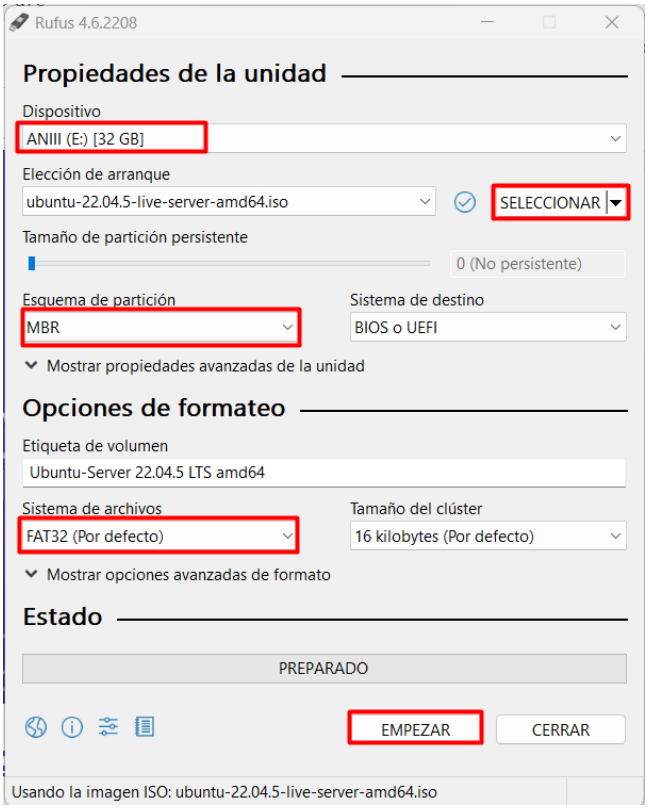
- **Creación de la unidad de arranque.**

1. Con la descarga finalizada se procede a crear la unidad de arranque que se va a emplear para bootear al dispositivo, el archivo es tipo iso, por lo que para que pueda ser usado se hará uso de un memoria USB. Para esto es necesario hacer uso de un software para la creación de unidades de arranque como Rufus, este puede obtenerse en su sitio oficial: https://rufus.ie/es/, se debe descargar e instalar en una PC.

2. Una vez instalado el software Rufus, se procede con la creación de la unidad de arranque, para ello se conecta la memoria USB (preferiblemente vacía) a la PC y se deben seleccionar los parámetros a considerar entre ellos: el dispositivo que se va a establecer como unidad de arranque (la memoria USB), el archivo que se va a setear (archivo .iso descargado), y aspectos como el esquema de partición y el sistema de archivos y se presiona el botón "EMPEZAR". Esta configuración se evidencia en la Figura 5.

Figura 5.

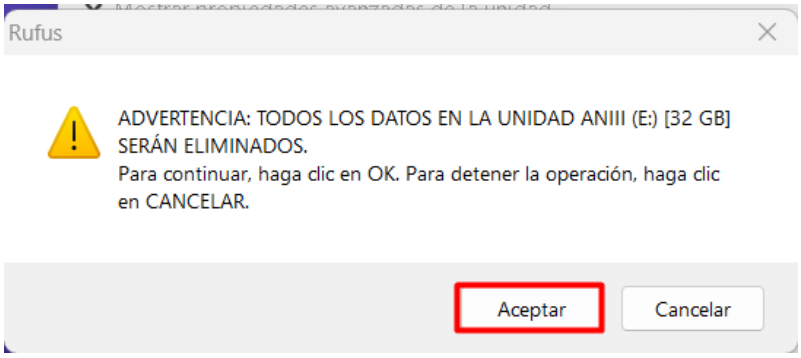
Parámetros configurados para la creación de la unidad de arranque.



3. Con la configuración definida el software solicitará una configuración para realizar un formateo previo de la memoria USB para que esta opere como unidad de arranque para instalar el sistema operativo indicado, se debe aceptar esto como se indica en la Figura 6, y se prosigue hasta que el software indique que se ha concluido el proceso.

Figura 6.

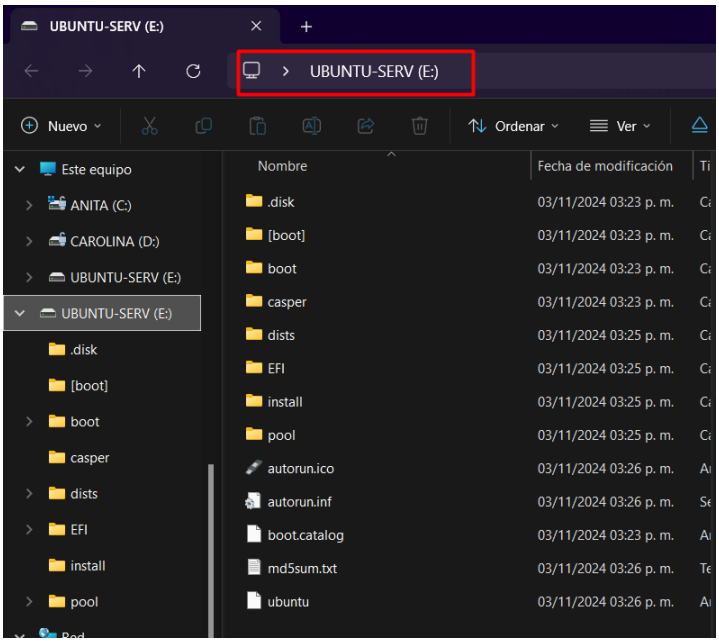
Advertencia de formateo de la memoria USB.



4. Una vez finalizado el proceso, se puede comprobar que la memoria USB contiene todos los archivos necesarios para actuar como unidad de arranque como se muestra en la Figura 7.

Figura 7.

Memoria USB configurada como unidad de arranque con el Sistema Operativo Ubuntu Server 22.04



- **Conexión y arranque del servidor.**

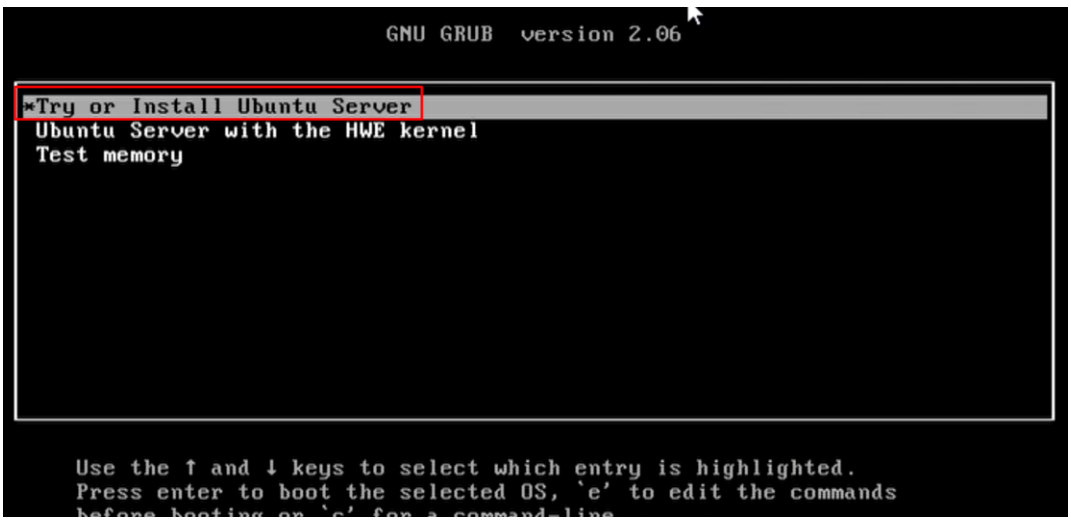
    1. Previo a iniciar el servidor, se debe considerar la conexión de un monitor que permita visualizar las acciones realizadas y un teclado para la introducción de información. Además, se debe conectar en una de las ranuras USB la unidad de arranque configurada con anterioridad.
    2. Con las conexiones realizadas se procede a iniciar el servidor, si este reconoce inmediatamente como unidad de arranque a la memoria USB se continua con el proceso de instalación tradicional; caso contrario se debe realizar la configuración de arranque de acuerdo con las especificaciones del fabricante del servidor.

- **Instalación y configuración del sistema operativo.**

    1. En cuanto el dispositivo reconozca como unidad de arranque a la memoria USB, se debe seleccionar con el teclado la opción de instalar el sistema operativo para iniciar con el proceso, esta selección se muestra en la Figura 8.

Figura 8.

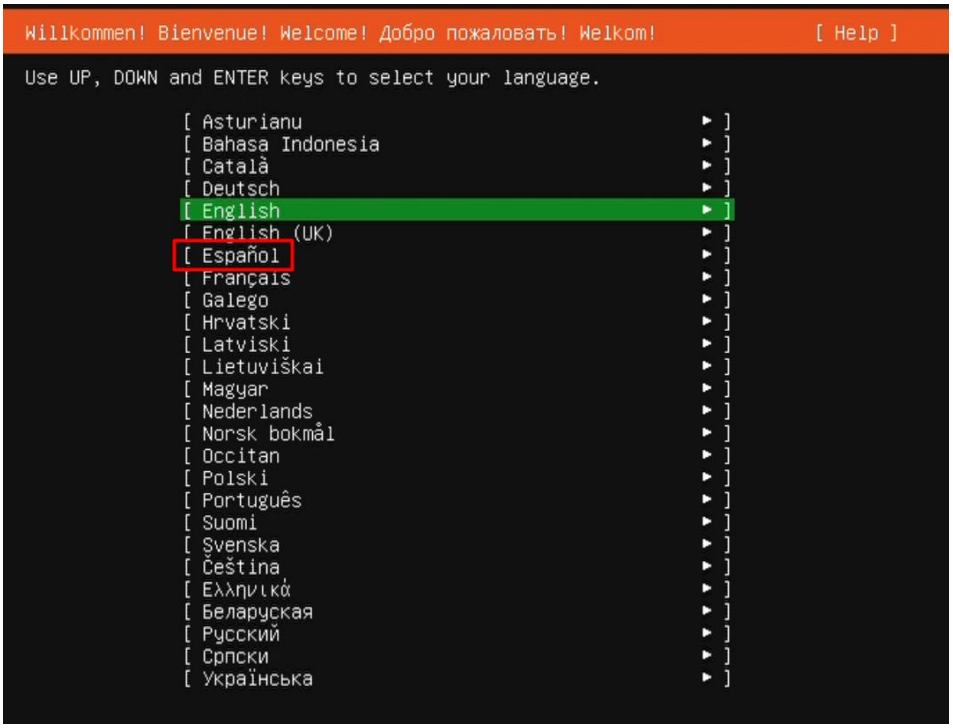
Selección de la opción de instalación de Ubuntu Server



2. La primer configuración a realizar indica el idioma en el que se instalará el sistema operativo, para ello se realizar la búsqueda con las flechas del teclado y la selección con la barra de espacios, la Figura 9 muestra el menú desplegado.

Figura 9.

Selección del idioma de instalación de Ubuntu Server.



3. Otra configuración relevante es la selección de tipo de teclado, su distribución y la variante para que de este modo la introducción de comandos sea lo más sencilla posible y se acople al dispositivo de entrada que se use con el servidor, si se conoce se puede seleccionar directamente d ellos menús desplegables, como se muestra en la Figura 10, o se puede identificar mediante la prueba realizada en la opción posterior previo a continuar.

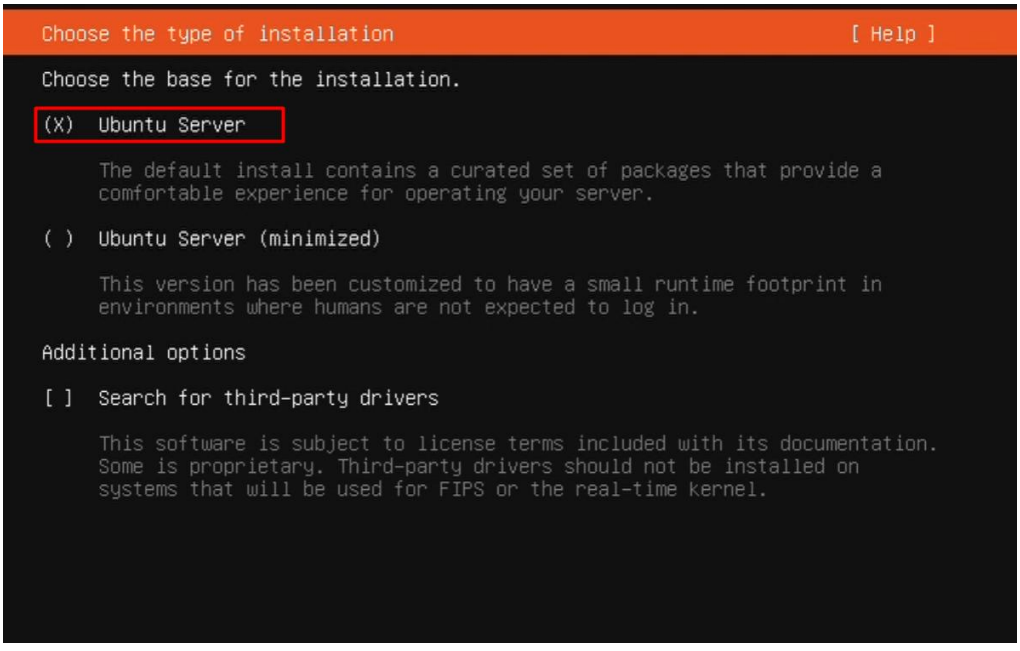
Figura 10.

Selección del idioma del teclado

4. Posteriormente, se selecciona la base para la instalación que especifica el tipo de paquetes que se van a instalar. En este caso se selecciona Ubuntu Server como se aprecia en la Figura 11.

Figura 11.

Selección de la base para instalación.

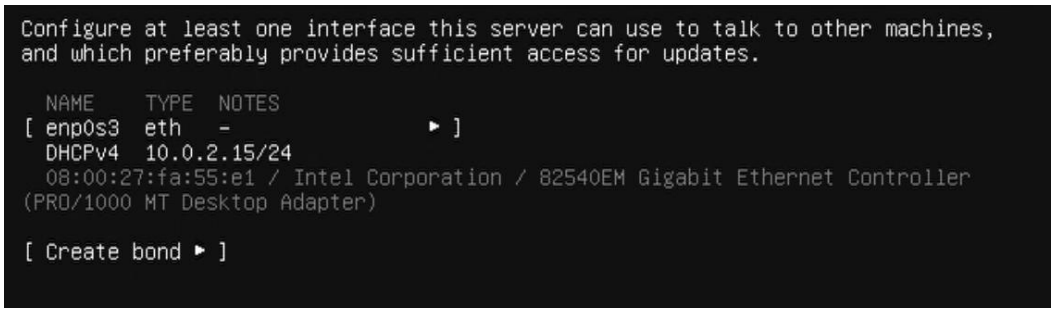


5. A continuación, se debe realizar la configuración de interfaces de red que debe hacerse en base a las interfaces del servidor, estas se detectan de forma

automática y pueden ser editadas para que correspondan a la configuración deseada. La Figura 12 muestra un ejemplo donde se especifica el nombre de la interfaz, dirección IP y la información del fabricante.

Figura 12.

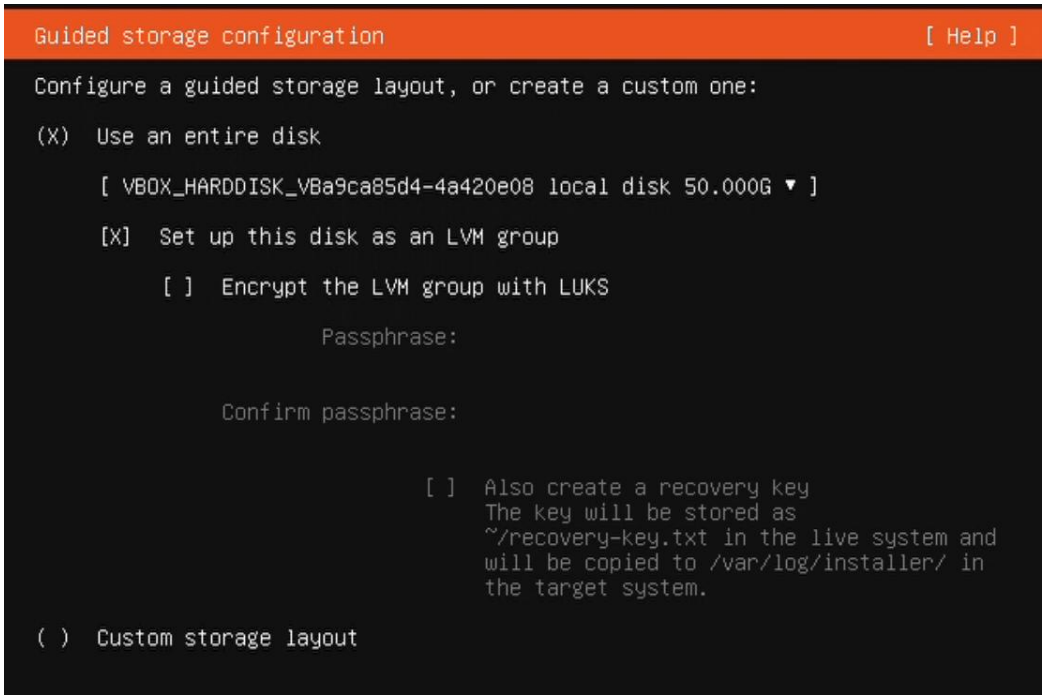
Configuración de adaptadores de red



6. A continuación, se realiza el particionamiento de disco, que puede configurarse utilizando la opción predeterminada del instalador o mediante una partición personalizada según los requisitos del servidor y el uso que se le dará. Para servidores destinados a aplicaciones o bases de datos, es recomendable asignar una mayor cantidad de espacio a la partición de datos, generalmente montada en /var o /srv. La partición raíz (/) debe tener suficiente espacio para el sistema operativo y las aplicaciones básicas. Para almacenar archivos de usuario o registros que ocupen mucho espacio, puedes crear una partición separada para /home o /var/log para evitar que el crecimiento de estos datos afecte el sistema principal. Adicionalmente, es recomendable asignar una pequeña partición /boot para los archivos de arranque y una swap en caso de que el servidor tenga recursos de memoria limitados. La Figura 13 muestra el menú de selección.

Figura 13.

Menú de selección del tipo de particionamiento



Posteriormente se deben crear las credenciales del usuario principal para el sistema Operativo, estas incluyen un nombre de usuario, el nombre del servidor y una contraseña que tenga la robustez pertinente para no generar una brecha de seguridad. La Figura 14 muestra los campos a considerar en este paso.

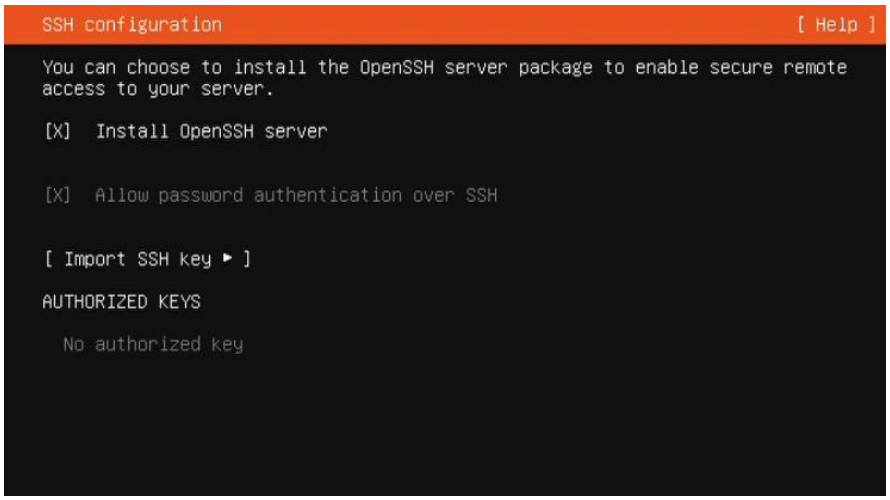
Figura 14.

Configuración de credenciales de usuario

7. Finalmente, se puede seleccionar si instalar de forma predeterminada un servidor SSH que admita conexiones remotas, esta es muy útil para la administración del equipo por lo que se puede habilitar la opción y al finalizar la instalación realizar las configuraciones de seguridad pertinentes, o se puede omitir y realizar la instalación del servidor y su configuración posteriormente. Este paso se evidencia en la Figura 15.

   Figura 15.

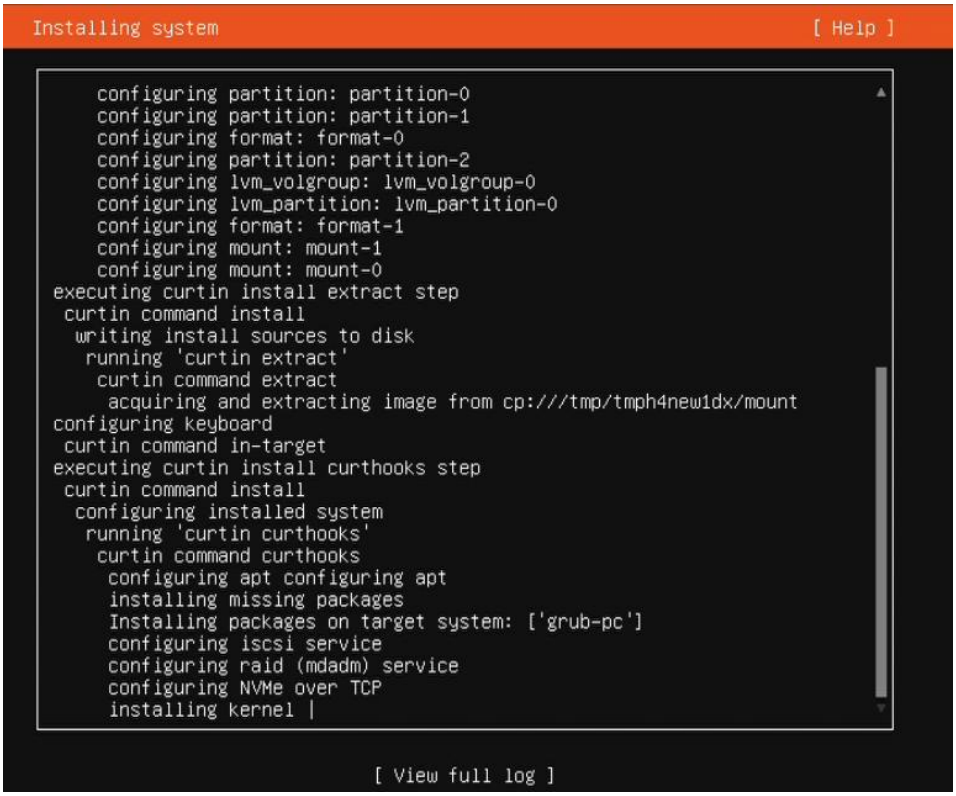
   Instalación de servidor OpenSSH



8. Finalmente se procede con la instalación de todos los componentes del sistema operativo, este proceso se despliega en la pantalla para visualizar el proceso que se está realizando como se aprecia en la Figura 16, puede tomar unos minutos por lo que se debe mantener encendido el servidor hasta que se muestre el mensaje de finalización.

Figura 16.

Instalación de componentes propios del sistema operativo

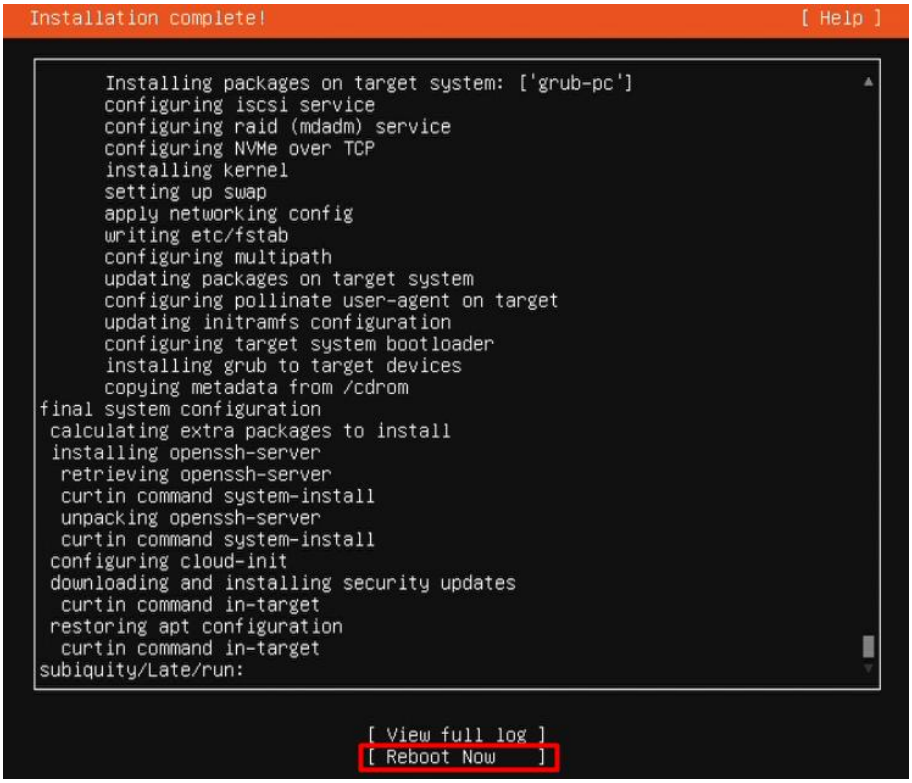


- **Reinicio y actualización.**

1. Con todos los componentes instalados, se procede con el reinicio del sistema operativo, para ello se debe seleccionar la opción “Reboot Now” que aparece en la parte inferior de la pantalla y es posible retirar la unidad de arranque empleada en el proceso de instalación. La Figura 17 muestra la pantalla final y la selección de la opción mencionada.

Figura 17.

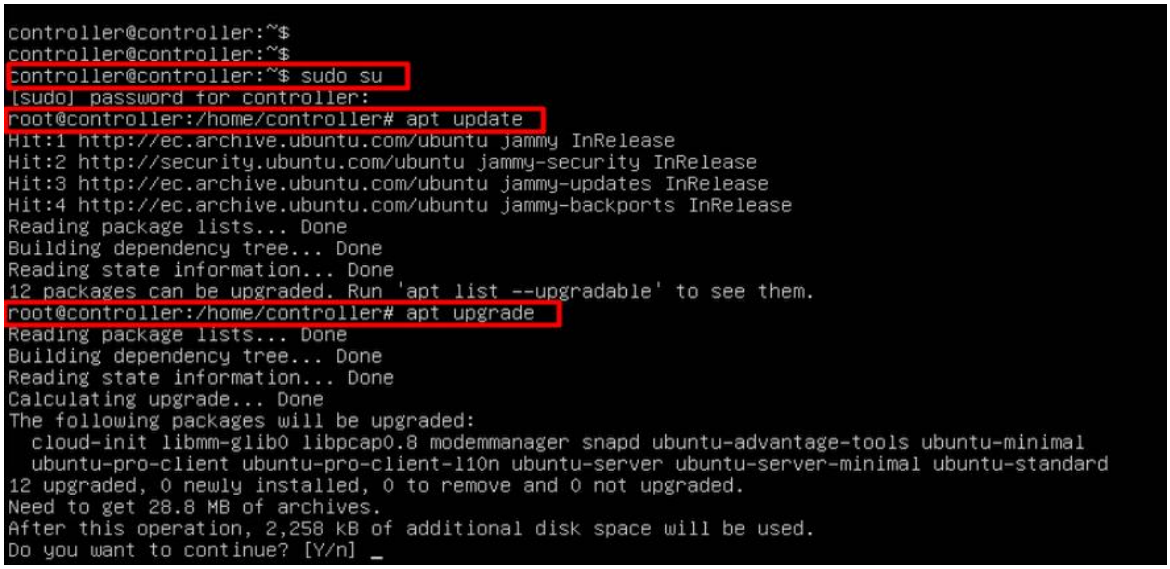
Finalización de la actualización y reinicio del sistema operativo



2. Con el sistema operativo reiniciado, es necesario realizar una actualización de todos los paquetes básicos del sistema, para lo cual con el comando `sudo su`, se accede a privilegios de administrador para solicitar que se verifiquen los paquetes a actualizar con el comando `apt update` y posteriormente la descarga de sus actualizaciones con el comando `apt upgrade`. Este proceso se evidencia en la Figura 18.

Figura 18.

Actualización de paquetes en el sistema operativo.



Una vez finalizado este proceso el servidor está listo para continuar con instalación de nuevos paquetes o la aplicación de otras configuraciones, es posible trabajar con el mediante conexión directa o por acceso remoto para interfaz de comandos, adicionalmente es posible la instalación de distintas interfaces gráficas para el manejo de archivos dentro del servidor, se pueden considerar alternativas de bajo consumo de recursos como : XFCE, MATE. Cinnamon, LXQt, entre otras.

Anexo 2: Configuración de OpenFlow en equipos Cisco

Objetivo:

- El propósito de este anexo es proporcionar una guía detallada del proceso de configuración del modo OpenFlow en dispositivos del fabricante Cisco siguiendo los pasos adjuntos en la documentación oficial del fabricante.

Cisco ha acoplado dentro de sus protocolos admitidos la compatibilidad con el protocolo OpenFlow para el manejo de redes definidas por software, por lo cual ha definido dentro de su documentación los comandos para permitir que sus dispositivos operen en esta modalidad, los pasos descritos en (Cisco, 2021b) se encuentran detallados en la Tabla 1.

Tabla 1.

Pasos para la configuración de OpenFlow en un dispositivo Cisco.

# Paso	Comando o acción	Descripción
Paso 1	**enable** **Ejemplo:** `Device> enable`	Habilita el modo EXEC privilegiado. Puede solicitar contraseña si se ha configurado previamente.
Paso 2	**configure terminal** **Ejemplo:** `Device# configure terminal`	Accede al modo de configuración global.
Paso 3	**feature openflow** **Ejemplo:** `Device(config)# feature openflow`	Habilita la comptibilidad con OpenFlow Agent en el dispositivo.
Paso 4	**openflow** **Ejemplo:** `Device(config)# openflow`	Habilita la compatibilidad con OpenFlow Agent en el dispositivo.
Paso 5	**switch** *logical-switch-id* **pipeline** *logical-id* **Ejemplo:**	Especifica una ID para el dispositivo lógico que se empleará para la conmutación con OpenFlow

	`Device(config-ofa-switch)#` **`switch 1 pipeline 1`**	y entra en el modo de configuración del mismo. Solo se admite el ID de conmutador o switch lógico 1. Configura la canalización. Es un paso obligatorio en la configuración de cualquier switch lógico, el ID de canalización admitido es 1.
Paso 6	**controller** [**ipv4** *ip-address*] [**port** *tcp-port*] [**vrf** *vrf-name*] [**security** {**none** | **tls**}] **Ejemplo:** `Device(config-ofa-switch)#` **`controller ipv4 10.1.1.1 tcp 6633`**	Especifica la Dirección IPv4 y el número de puerto empleado por el controlador para conectarse al switch lógico. Se repite este paso si se necesita configurar switches adicionales. Se pueden configurar hasta 8 conmutadores. Si se usa TLS en este paso, se configuran los puntos de confianza de TLS en el siguiente paso. De no especificarse, predeterminadamente, lo conmutadores utilizan el puerto TCP 6633. El switch lógico establece una conexión con el controlador.
Paso 7	**of-port interface** *interface-name* **Ejemplo:** `Device(config-ofa-switch)#` **`of-port interface GigabitEthernet1/0/23`** `Device(config-ofa-switch)#` **`of-port interface TenGigabitEthernet1/1/2`**	Agrega interfaces a la configuración del switch lógico. Las pautas son: - No abreviar el tipo de interfaz. El nombre debe escribirse completo y de forma correcta como se muestra en el ejemplo. - Si se abrevian palabras la interfaz no será configurada. - La interfaz debe definirse únicamente para el switch lógico OpenFlow.

		Este paso se puede repetir para todas las interfaces necesarias.
Paso 8	**default-miss** *action-for-unmatched-flows* **Ejemplo:** `Device(config-ofa-switch)#` **`default-miss continue-controller`**	Configura la acción a realizar para los paquetes que no coinciden con ninguno de los flujos definidos, admite las siguientes opciones: - Reenviar paquetes usando tablas de enrutamiento tradicionales. - Reenviar paquetes al controlador. - Descartar paquetes. La opción definida por defecto es reenviar paquetes usando tablas de enrutamiento tradicionales.
Paso 9	**protocol-version {1.0 | 1.3 | negotiate}** **Ejemplo:** `Device(config-ofa-switch)#` **`protocol-version negotiate`**	Configura la versión del protocolo, los valores admitidos son: - 1.0: configura el dispositivo para conectarse únicamente con controladores 1.0. - 1.3: configura el dispositivo para conectarse únicamente con controladores 1.3. - negociate: inicia un proceso de negociación del protocolo con el controlador. El dispositivo emplea la versión 1.3 para la negociación. El valor predeterminado es 1.0.
Paso 10	**Shutdown** **Ejemplo:** `Device(config-ofa-switch)#` **`shutdown`**	Deshabilita el switch lógico cerrando la conexión TCP/IP y elimina los flujos del plano de datos.

Paso 11	**datapath-id** *datapath-id* **Ejemplo:** `Device(config-ofa-switch)# datapath-id 0x222`	Configura una ID de ruta de datos única para el switch. Este paso es obligatorio para la configuración, se debe introducir un valor hexadecimal de 64 bits.
Paso 12	**tls trust-point local** *local-trust-point* **remote** *remote-trust-point* **Ejemplo:** `Device(config-ofa-switch)# tls trust-point local myCA remote myCA`	Comando opcional, especifica los puntos de confianza TLS locales y remotos que se emplearán para la conexión del controlador.
Paso 13	**probe-interval** *probe-interval* **Ejemplo:** Device(config-ofa-switch)# **probe-interval 7**	Comando opcional, configura el intervalo en segundos en el que el controlador realiza sondeo. Una vez definido el valor de tiempo, si el switch no ha recibido ningún mensaje del controlador envía un mensaje (echo-request) hacia el controlador esperando recibir una respuesta (eco-reply). Si no se recibe ningún mensaje durante este intervalo el switch asume que el controlador está inactivo y cierra la conexón e intenta reconectarse periódicamente. El valor predeterminado es 5 segundos y el rango se encuentra entre 5 y 65535 segundos.
Paso 14	**rate-limit packet_in** *controllet-packet-rate* **burst** *maximum-packets-to-controller* **Ejemplo:** `Device(config-ofa-switch)# rate-limit packet_in 300 burst 50`	Comando opcional, configura la velocidad y la ráfaga máximas de paquetes enviados al controlador por segundo. Este valor predeterminado es cero, que indica que se permite un valor indefinido de ráfaga y velocidad de paquetes.

		El límite de velocidad para OpenFlow no está relacionado con el límite de velocidad de dispositivo (plano de datos) configurado por COPP.
Paso 15	**max-backoff** *backoff-timer* **Ejemplo:** `Device(config-ofa-switch)# max-backoff 8`	Comando opcional, configura el tiempo en segundos que el dispositivo debe esperar antes de intentar una conexión con el controlador. Inicialmente, el dispositivo intenta iniciar una conexión con frecuencia, a medida que incrementa el número de intentos fallidos, el dispositivo reduce la frecuencia de intentos de conexión y el periodo de espera se hace más largo. El temporizador de retroceso configura el periodo máximo en el que el dispositivo espera entre cada reintento. El valor predeterminado es 8 segundos, y el rango es de 1 a 65535 segundos.
Paso 16	**logging flow-mod** **Ejemplo:** `Device(config-ofa-switch)# logging flow-mod`	Comando opcional, habilita el registro de cambios de flujo, incluyendo adición, eliminación y modificación de flujos. El registro de cambios de flujo es actividad con alto consumo de CPU y no debe habilitarse para grandes cantidades de flujo y se encuentra deshabilitado por defecto. Los cambios de flujo se registran en syslog y se pueden ver usando el comando **show logging.**

Paso 17	**statistics collection-interval** *interval* **Ejemplo:** `Device(config-ofa-switch)# statistics collection-interval 7`	Configura el intervalo de recopilación de estadísticas en segundos para todos los flujos configurados de OpenFlow. Las pautas dadas son: - El valor de intervalo por defecto es 7 segundos, y el máximo es de 82 segundos. - Si se establece el intervalo en cero se deshabilita la recopilación de estadísticas. - Los flujos con un valor de tiempo de espera de inactividad menores a los intervalos de 2* se rechazan. El valor del intervalo configurado se muestra en la salida del comando show openflow switch 1.
Paso 18	**end** **Ejemplo:** `Device(config-ofa-switch)# end`	Regresa al modo EXEC privilegiado. Puede ser reemplazado por presionar las teclas **Ctrl-Z** para salir directamente al modo de configuración global del dispositivo.
Paso 19	**copy running-config startup-config** **Ejemplo:** `Device# copy running-config startup-config`	Comando opcional, guarda todas las entradas en el archivo de configuración del dispositivo.

Fuente: Adaptado de (Cisco, 2021b)

Anexo 3: Script para la creación de flujos en IPv4 e IPv6

Objetivo:

- La finalidad de este anexo es definir un script ejecutable con pyhton3 dentro del sistema operativo que aloja al controlador para la instalación de flujos que sustituyan la configuración de vlans tradicionales dentro de los dispositivos en IPv4 e IPv6 para la conmutación de paquetes de la misma subred como se haría en tráfico "intra-vlan".

De forma previa a la definición del script, se debe considerar que el sistema operativo tenga instaladas las funcionalidades de python mediante el comando `apt install pip3`, y la creación del archivo mediante el comando `sudo nano <nombre del archivo>.py` para su posterior ejecución. La ejecución del archivo se realizará mediante el comando `python3 <nombre del archivo>.py`

```
#-----------------Creación de flujos para tráfico de la misma
subred-----------------

# Importació de módulos de pyhton necesarios para la
configuración.

import requests
import json
import time

# Configuración de credenciales de acceso hacia el controlador.

controller_ip = "192.168.10.10"
controller_port = "8181"
username = "admin"
password = "admin"

# Definición de las subredes definidas para cada una de las
vlans existentes en la red en IPv4 e IPv6.

vlans = [
    {"id": 1, "subnet_ipv4": "172.16.1.0/24", "subnet_ipv6":
"2801:10:6800:1::/64"},
    {"id": 2, "subnet_ipv4": "10.24.8.0/24", "subnet_ipv6":
"2801:10:6800:1024::/64"},
    {"id": 3, "subnet_ipv4": "172.16.3.0/24", "subnet_ipv6":
"2801:10:6800:3::/64"},
```

```
    {"id": 4, "subnet_ipv4": "172.16.4.0/24", "subnet_ipv6":
"2801:10:6800:4::/64"},
    {"id": 6, "subnet_ipv4": "172.16.6.0/24", "subnet_ipv6":
"2801:10:6800:6::/64"},
    {"id": 7, "subnet_ipv4": "172.16.7.0/24", "subnet_ipv6":
"2801:10:6800:7::/64"},
    {"id": 8, "subnet_ipv4": "172.16.8.0/22", "subnet_ipv6":
"2801:10:6800:8::/64"},
    {"id": 12, "subnet_ipv4": "172.16.12.0/24", "subnet_ipv6":
"2801:10:6800:12::/64"},
    {"id": 14, "subnet_ipv4": "172.16.14.0/24", "subnet_ipv6":
"2801:10:6800:14::/64"},
    {"id": 16, "subnet_ipv4": "172.16.16.0/24", "subnet_ipv6":
"2801:10:6800:16::/64"},
    {"id": 18, "subnet_ipv4": "172.16.18.0/24", "subnet_ipv6":
"2801:10:6800:18::/64"},
    {"id": 20, "subnet_ipv4": "172.16.20.0/24", "subnet_ipv6":
"2801:10:6800:20::/64"},
    {"id": 22, "subnet_ipv4": "172.16.22.0/24", "subnet_ipv6":
"2801:10:6800:22::/64"},
    {"id": 24, "subnet_ipv4": "172.16.24.0/23", "subnet_ipv6":
"2801:10:6800:24::/64"},
    {"id": 26, "subnet_ipv4": "172.16.26.0/24", "subnet_ipv6":
"2801:10:6800:26::/64"},
    {"id": 28, "subnet_ipv4": "172.16.28.0/24", "subnet_ipv6":
"2801:10:6800:28::/64"},
    {"id": 30, "subnet_ipv4": "172.16.30.0/24", "subnet_ipv6":
"2801:10:6800:30::/64"},
    {"id": 32, "subnet_ipv4": "172.16.32.0/24", "subnet_ipv6":
"2801:10:6800:32::/64"},
    {"id": 40, "subnet_ipv4": "172.17.40.0/23", "subnet_ipv6":
"2801:10:6800:40::/64"},
    {"id": 42, "subnet_ipv4": "172.17.42.0/24", "subnet_ipv6":
"2801:10:6800:42::/64"},
    {"id": 44, "subnet_ipv4": "172.16.44.0/24", "subnet_ipv6":
"2801:10:6800:44::/64"},
    {"id": 48, "subnet_ipv4": "172.17.48.0/23", "subnet_ipv6":
"2801:10:6800:48::/64"},
    {"id": 52, "subnet_ipv4": "172.16.52.0/24", "subnet_ipv6":
"2801:10:6800:52::/64"},
    {"id": 56, "subnet_ipv4": "172.17.56.0/23", "subnet_ipv6":
"2801:10:6800:56::/64"},
    {"id": 60, "subnet_ipv4": "172.16.60.0/24", "subnet_ipv6":
"2801:10:6800:60::/64"},
    {"id": 64, "subnet_ipv4": "172.17.64.0/23", "subnet_ipv6":
"2801:10:6800:64::/64"},
    {"id": 68, "subnet_ipv4": "172.16.68.0/24", "subnet_ipv6":
"2801:10:6800:68::/64"},
    {"id": 72, "subnet_ipv4": "172.17.72.0/23", "subnet_ipv6":
"2801:10:6800:72::/64"},
    {"id": 76, "subnet_ipv4": "172.16.76.0/24", "subnet_ipv6":
"2801:10:6800:76::/64"},
    {"id": 80, "subnet_ipv4": "172.17.80.0/23", "subnet_ipv6":
"2801:10:6800:80::/64"},
    {"id": 84, "subnet_ipv4": "172.16.84.0/24", "subnet_ipv6":
"2801:10:6800:84::/64"},
```

```
    {"id": 88, "subnet_ipv4": "172.17.88.0/23", "subnet_ipv6":
"2801:10:6800:88::/64"},
    {"id": 92, "subnet_ipv4": "172.16.92.0/24", "subnet_ipv6":
"2801:10:6800:92::/64"},
    {"id": 96, "subnet_ipv4": "172.17.96.0/23", "subnet_ipv6":
"2801:10:6800:96::/64"},
    {"id": 98, "subnet_ipv4": "172.16.98.0/24", "subnet_ipv6":
"2801:10:6800:98::/64"},
    {"id": 100, "subnet_ipv4": "172.16.100.0/24", "subnet_ipv6":
"2801:10:6800:100::/64"},
    {"id": 128, "subnet_ipv4": "172.20.128.0/19", "subnet_ipv6":
"2801:10:6800:128::/64"},
    {"id": 160, "subnet_ipv4": "172.21.160.0/21", "subnet_ipv6":
"2801:10:6800:160::/64"},
    {"id": 192, "subnet_ipv4": "172.23.192.0/19", "subnet_ipv6":
"2801:10:6800:192::/64"},
    {"id": 201, "subnet_ipv4": "172.24.201.0/24", "subnet_ipv6":
"2801:10:6800:201::/64"},
    {"id": 202, "subnet_ipv4": "172.25.202.0/24", "subnet_ipv6":
"2801:10:6800:202::/64"},
    {"id": 203, "subnet_ipv4": "172.25.203.0/24", "subnet_ipv6":
"2801:10:6800:203::/64"},
    {"id": 204, "subnet_ipv4": "172.25.204.0/24", "subnet_ipv6":
"2801:10:6800:204::/64"},
    {"id": 205, "subnet_ipv4": "172.25.205.0/24", "subnet_ipv6":
"2801:10:6800:205::/64"},
    {"id": 1000, "subnet_ipv4": "192.168.10.0/24",
"subnet_ipv6": "2001:db8:1000::/64"}
]

# Definición de una función para la creación de una regla de
flujo para tráfico dentro de la misma subred basado en IPv4 o
IPv6.

def create_intra_vlan_flow(switch_id, vlan_id, subnet,
is_ipv6=False):
    flow_id = f"intra_vlan_{vlan_id}_{'ipv6' if is_ipv6 else
'ipv4'}"
    eth_type = 34525 if is_ipv6 else 2048
    ip_match_field = "ipv6" if is_ipv6 else "ipv4"

    flow_data = {
        "flow": [
            {
                "id": flow_id,
                "match": {
                    "ethernet-match": {
                        "ethernet-type": {
                            "type": eth_type
                        }
                    },
                    f"{ip_match_field}-source": subnet,
                    f"{ip_match_field}-destination": subnet
                },
                "instructions": {
                    "instruction": [
```

```
                        {
                            "order": 0,
                            "apply-actions": {
                                "action": [
                                    {
                                        "order": 0,
                                        "output-action": {
                                            "output-node-
connector": "NORMAL"
                                        }
                                    }
                                ]
                            }
                        }
                    ]
                },
                "priority": 300,
                "table_id": 0
            }
        ]
    }
    url =
f"http://{controller_ip}:{controller_port}/restconf/config/opend
aylight-inventory:nodes/node/{switch_id}/table/0/flow/{flow_id}"
    headers = {'Content-Type': 'application/json'}
    response = requests.put(url, data=json.dumps(flow_data),
headers=headers, auth=(username, password))

    if response.status_code == 200 or response.status_code ==
201:
        print(f"Flow {flow_id} successfully created on
{switch_id}")
    else:
        print(f"Error creating flow {flow_id} on {switch_id}:
{response.status_code} - {response.text}")

# Aplicación de reglas de flujo intra-VLAN para IPv4 e IPv6 con
delay para evitar sobrecargar a los switches con envíos masivos.

def apply_intra_vlan_rules():

#Definir los switches en base al identificador que tiene el
controlador: openflow:<datapath>

    switches = ["openflow:1", "openflow:22", "openflow:3",
"openflow:4", "openflow:5",
                "openflow:6", "openflow:7", "openflow:8",
"openflow:9", "openflow:10",
                "openflow:11", "openflow:12", "openflow:13",
"openflow:14", "openflow:15"]

    for switch in switches:
        for vlan in vlans:
            # Crear regla de flujo para IPv4
            create_intra_vlan_flow(switch, vlan["id"],
vlan["subnet_ipv4"], is_ipv6=False)
```

```
            time.sleep(1)  # Delay para evitar sobrecarga

            # Crear regla de flujo para IPv6
            create_intra_vlan_flow(switch, vlan["id"],
vlan["subnet_ipv6"], is_ipv6=True)
            time.sleep(1)  # Delay para evitar sobrecarga

# Ejecutar la aplicación de reglas intra-VLAN
apply_intra_vlan_rules()
```

Anexo 4: Script para enrutamiento entre subredes internas IPv4

Objetivo:

- La finalidad de este anexo es definir un script ejecutable con pyhton3 dentro del sistema operativo que aloja al controlador para el enrutamiento de las subredes internas IPv4 definidas con anterioridad sustituyendo al enrutamiento inter-vlan que opera dentro de la arquitectura tradicional.

De forma previa a la definición del script, se debe considerar que el sistema operativo tenga instaladas las funcionalidades de python mediante el comando `apt install pip3`, y la creación del archivo mediante el comando `sudo nano <nombre del archivo>.py` para su posterior ejecución. La ejecución del archivo se realizará mediante el comando `python3 <nombre del archivo>.py`

```
import requests

# Lista de gateways IPv4 para Router 1
gateways = [
    "172.16.1.1/24", "172.16.4.1/24", "172.16.6.1/24",
"172.16.7.1/24",
    "172.16.8.1/22", "172.16.12.1/24", "172.16.14.1/24",
"172.16.16.1/24",
    "172.16.18.1/24", "172.16.20.1/24", "172.16.22.1/24",
"172.16.24.1/23",
    "172.20.128.1/19", "172.21.160.1/21", "172.23.192.1/19",
"172.24.201.1/24",
    "172.25.202.1/24", "172.25.203.1/24", "172.25.204.1/24",
"172.25.205.1/24",
    "192.168.10.1/24", "172.17.56.1/23", "172.16.60.1/24",
"172.17.64.1/23",
    "172.16.68.1/24", "172.16.26.1/24", "172.17.80.1/23",
"172.16.84.1/24",
    "172.16.32.1/24", "172.17.72.1/23", "172.16.76.1/24",
"172.17.88.1/23",
    "172.16.92.1/24", "172.17.48.1/23", "172.16.52.1/24",
"172.17.96.1/23",
    "172.16.98.1/24", "172.16.100.1/24", "172.17.40.1/23",
"172.17.42.1/24",
    "172.16.44.1/24", "172.16.28.1/24", "172.16.30.1/24"
```

```
]

# Lista de rutas estáticas IPv4 para Router 1
routes = [
    {"destination": "172.16.1.0/24", "gateway": "172.16.1.1"},
    {"destination": "172.16.4.0/24", "gateway": "172.16.4.1"},
    {"destination": "172.16.6.0/24", "gateway": "172.16.6.1"},
    {"destination": "172.16.7.0/24", "gateway": "172.16.7.1"},
    {"destination": "172.16.8.0/22", "gateway": "172.16.8.1"},
    {"destination": "172.16.10.0/24", "gateway": "172.16.10.1"},
    {"destination": "172.16.12.0/24", "gateway": "172.16.12.1"},
    {"destination": "172.16.14.0/24", "gateway": "172.16.14.1"},
    {"destination": "172.16.16.0/24", "gateway": "172.16.16.1"},
    {"destination": "172.16.18.0/24", "gateway": "172.16.18.1"},
    {"destination": "172.16.20.0/24", "gateway": "172.16.20.1"},
    {"destination": "172.16.22.0/24", "gateway": "172.16.22.1"},
    {"destination": "172.16.24.0/23", "gateway": "172.16.24.1"},
    {"destination": "172.16.26.0/24", "gateway": "172.16.26.1"},
    {"destination": "172.16.28.0/24", "gateway": "172.16.28.1"},
    {"destination": "172.16.30.0/24", "gateway": "172.16.30.1"},
    {"destination": "172.16.32.0/24", "gateway": "172.16.32.1"},
    {"destination": "172.17.40.0/23", "gateway": "172.17.40.1"},
    {"destination": "172.17.42.0/23", "gateway": "172.17.42.1"},
    {"destination": "172.16.44.0/24", "gateway": "172.16.44.1"},
    {"destination": "172.17.48.0/23", "gateway": "172.17.48.1"},
    {"destination": "172.16.52.0/24", "gateway": "172.16.52.1"},
    {"destination": "172.17.56.0/23", "gateway": "172.17.56.1"},
    {"destination": "172.16.60.0/24", "gateway": "172.16.60.1"},
    {"destination": "172.17.64.0/23", "gateway": "172.17.64.1"},
    {"destination": "172.16.68.0/24", "gateway": "172.16.68.1"},
    {"destination": "172.17.72.0/23", "gateway": "172.17.72.1"},
    {"destination": "172.16.76.0/24", "gateway": "172.16.74.1"},
    {"destination": "172.17.80.0/23", "gateway": "172.17.80.1"},
    {"destination": "172.16.84.0/23", "gateway": "172.16.84.1"},
    {"destination": "172.17.88.0/23", "gateway": "172.17.88.1"},
    {"destination": "172.16.92.0/24", "gateway": "172.16.92.1"},
    {"destination": "172.17.96.0/23", "gateway": "172.17.96.1"},
    {"destination": "172.16.98.0/24", "gateway": "172.16.98.1"},
    {"destination": "172.16.100.0/24", "gateway":
"172.16.100.1"},
    {"destination": "172.20.128.0/19", "gateway":
"172.20.128.1"},
    {"destination": "172.21.160.0/21", "gateway":
"172.21.160.1"},
    {"destination": "172.23.192.0/19", "gateway":
"172.23.192.1"},
    {"destination": "172.24.201.0/24", "gateway":
"172.24.201.1"},
```

```
    {"destination": "172.25.202.0/24", "gateway":
"172.25.202.1"},
    {"destination": "172.25.203.0/24", "gateway":
"172.25.203.1"},
    {"destination": "172.25.204.0/24", "gateway":
"172.25.204.1"},
    {"destination": "172.25.205.0/24", "gateway":
"172.25.205.1"},
    {"destination": "192.168.10.0/24", "gateway":
"192.168.10.1"}
]

def configure_gateways():
    print("Configurando gateways en Router 1...")
    for gw in gateways:
        response = requests.post(
            "http://localhost:8080/router/0000000000000001",
            json={"address": gw}
        )
        print(f"Gateway {gw}: {response.status_code},
{response.reason}")

def configure_routes():
    print("Configurando rutas estáticas en Router 1...")
    for route in routes:
        response = requests.post(
            "http://localhost:8080/router/0000000000000001",
            json=route
        )
        print(f"Ruta {route}: {response.status_code},
{response.reason}")

if __name__ == "__main__":
    configure_gateways()
    configure_routes()
```

Anexo 5: Configuración para un Router Cisco

Objetivo:

- La finalidad de este anexo es definir las configuraciones básicas a realizar en un router Cisco para la asignación de direcciones IPv4 e IPv6 y la definición de rutas estáticas para un enrutamiento básico en la red interna y salida hacia la WAN.

Dentro de la red definida por software es necesario contra con un dispositivo que realice la resolución de peticiones ARP y que almacene rutas para el encaminamiento de los paquetes en base a las reglas de flujo establecidas por el controlador, para ello se consideran las siguientes configuraciones a insertar en el CLI de un router Cisco.

```
--------------Configuración de direcciones IPv4-----------------
enable
configure terminal
-----------Asignación de IPs a interfaces----------------------
interface GigabitEthernet0/0
ip address 172.16.3.4 255.255.255.0
ip address 172.16.1.1 255.255.255.0 secondary
ip address 172.16.3.4 255.255.255.0 secondary
ip address 172.16.4.1 255.255.255.0 secondary
ip address 172.16.6.1 255.255.255.0 secondary
ip address 172.16.7.1 255.255.255.0 secondary
ip address 172.16.8.1 255.255.252.0 secondary
ip address 172.16.12.1 255.255.255.0 secondary
ip address 172.16.14.1 255.255.255.0 secondary
ip address 172.16.16.1 255.255.255.0 secondary
ip address 172.16.18.1 255.255.255.0 secondary
ip address 172.16.20.1 255.255.255.0 secondary
ip address 172.16.22.1 255.255.255.0 secondary
ip address 172.16.24.1 255.255.254.0 secondary
ip address 172.16.26.1 255.255.255.0 secondary
ip address 172.16.28.1 255.255.255.0 secondary
```

```
ip address 172.16.30.1 255.255.255.0 secondary
ip address 172.16.32.1 255.255.255.0 secondary
ip address 172.17.40.1 255.255.254.0 secondary
ip address 172.17.42.1 255.255.255.0 secondary
ip address 172.16.44.1 255.255.255.0 secondary
ip address 172.17.48.1 255.255.254.0 secondary
ip address 172.16.52.1 255.255.255.0 secondary
ip address 172.17.56.1 255.255.254.0 secondary
ip address 172.16.60.1 255.255.255.0 secondary
ip address 172.17.64.1 255.255.254.0 secondary
ip address 172.16.68.1 255.255.255.0 secondary
ip address 172.17.72.1 255.255.254.0 secondary
ip address 172.16.76.1 255.255.255.0 secondary
ip address 172.17.80.1 255.255.254.0 secondary
ip address 172.16.84.1 255.255.255.0 secondary
ip address 172.17.88.1 255.255.254.0 secondary
ip address 172.16.92.1 255.255.255.0 secondary
ip address 172.17.96.1 255.255.254.0 secondary
ip address 172.16.98.1 255.255.255.0 secondary
ip address 172.16.100.1 255.255.255.0 secondary
ip address 172.20.128.1 255.255.224.0 secondary
ip address 172.21.160.1 255.255.248.0 secondary
ip address 172.23.192.1 255.255.224.0 secondary
ip address 172.24.201.1 255.255.255.0 secondary
ip address 172.25.202.1 255.255.255.0 secondary
ip address 172.25.203.1 255.255.255.0 secondary
ip address 172.25.204.1 255.255.255.0 secondary
ip address 172.25.205.1 255.255.255.0 secondary
no sh

interface GigabitEthernet1/0
ip address 192.168.10.1 255.255.255.0
no sh
------------Definición de rutas estáticas--------------------
```

```
ip route 10.24.8.0 255.255.255.0 172.16.3.1
ip route 0.0.0.0 0.0.0.0 172.16.3.1

--------------Ipv6--------------------------
configure terminal
---------------Configuración de direcciones IPv6----------------
interface GigabitEthernet0/0
ipv6 address 2801:10:6800:4::1/64
 ipv6 address 2801:10:6800:1::1/64
 ipv6 address 2801:10:6800:3::4/64
 ipv6 address 2801:10:6800:6::1/64
 ipv6 address 2801:10:6800:7::1/64
 ipv6 address 2801:10:6800:8::1/64
 ipv6 address 2801:10:6800:12::1/64
 ipv6 address 2801:10:6800:14::1/64
 ipv6 address 2801:10:6800:16::1/64
 ipv6 address 2801:10:6800:18::1/64
 ipv6 address 2801:10:6800:20::1/64
 ipv6 address 2801:10:6800:22::1/64
 ipv6 address 2801:10:6800:24::1/64
 ipv6 address 2801:10:6800:26::1/64
 ipv6 address 2801:10:6800:28::1/64
 ipv6 address 2801:10:6800:30::1/64
 ipv6 address 2801:10:6800:32::1/64
 ipv6 address 2801:10:6800:40::1/64
 ipv6 address 2801:10:6800:42::1/64
 ipv6 address 2801:10:6800:44::1/64
 ipv6 address 2801:10:6800:48::1/64
 ipv6 address 2801:10:6800:52::1/64
 ipv6 address 2801:10:6800:56::1/64
 ipv6 address 2801:10:6800:60::1/64
 ipv6 address 2801:10:6800:64::1/64
 ipv6 address 2801:10:6800:68::1/64
 ipv6 address 2801:10:6800:72::1/64
```

```
 ipv6 address 2801:10:6800:76::1/64
 ipv6 address 2801:10:6800:80::1/64
 ipv6 address 2801:10:6800:84::1/64
 ipv6 address 2801:10:6800:88::1/64
 ipv6 address 2801:10:6800:92::1/64
 ipv6 address 2801:10:6800:96::1/64
 ipv6 address 2801:10:6800:98::1/64
 ipv6 address 2801:10:6800:100::1/64
 ipv6 address 2801:10:6800:128::1/64
 ipv6 address 2801:10:6800:160::1/64
 ipv6 address 2801:10:6800:192::1/64
 ipv6 address 2801:10:6800:201::1/64
 ipv6 address 2801:10:6800:202::1/64
 ipv6 address 2801:10:6800:203::1/64
 ipv6 address 2801:10:6800:204::1/64
 ipv6 address 2801:10:6800:205::1/64
 no shutdown

interface GigabitEthernet1/0
 ipv6 address 2001:db8:1000::1/64
 no shutdown
exit

------------Habilitación de enrutamiento--------------------
ipv6 unicast-routing
ipv6 multicast-routing

------------Definición de rutas estáticas--------------------
ipv6 route 2801:10:6800:1024::/64 2801:10:6800:3::1
ipv6 route ::/0 2801:10:6800:3::1
```