UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y DISEÑO DIGITAL
CARRERA DE TELEMÁTICA

Trabajo de Integración Curricular previa a la obtención del Grado Académico de Ingeniero en Telemática.

PROYECTO DE INVESTIGACIÓN:

ARQUITECTURA DE RED DEFINIDA POR SOFTWARE REFERENCIAL PARA UN PROVEEDOR DE SERVICIOS DE INTERNET

AUTOR:

JOHN DANIEL PINCAY MURILLO

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN:

ING. EMILIO RODRIGO ZHUMA MERA, MSc.

QUEVEDO – LOS RÍOS – ECUADOR

2025

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **JOHN DANIEL PINCAY MURILLO**, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

JOHN DANIEL PINCAY MURILLO

C.I: 120531442-8

CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN

El suscrito, **Emilio Rodrigo Zhuma Mera,** Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante **John Daniel Pincay Murillo**, realizó el Proyecto de Investigación de grado titulado "**Arquitectura de red definida por software referencial para un proveedor de servicios de internet**", previo a la obtención del título de **Ingeniero en Telemática**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Emilio Rodrigo Zhuma Mera, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

El suscrito, **Emilio Rodrigo Zhuma Mera**, mediante el presente cumplo en presentar a usted, el informe de proyecto de Investigación "**Arquitectura de red definida por software referencial para un proveedor de servicios de internet**" Presentado por el estudiante **John Daniel Pincay Murillo**, egresado de la Carrera de Telemática, que fue revisado bajo mi dirección según resolución del Consejo Directivo de la Facultad de Ciencias de la Computación y Diseño Digital, que se ha desarrollado de acuerdo al Reglamento de la Unidad de Integración Curricular de la Universidad Técnica Estatal de Quevedo y cumple con el requerimiento de análisis del sistema COMPILATIO el cual avala los niveles de originalidad en un 99% y similitud 1%, del trabajo investigativo. Valido este documento para que el estudiante siga con los trámites pertinentes, de acuerdo como lo establece el Reglamento.


CERTIFICADO DE ANÁLISIS
magister

TESIS PINCAY MURILLO JOHN

< 1% Textos sospechosos

< 1% Similitudes
< 1 % similitudes entre comillas
0 % entre las fuentes mencionadas
0% Idiomas no reconocidos

Nombre del documento: TESIS PINCAY MURILLO JOHN.pdf
ID del documento: f7f67742fb55ca720b4aaf2cdd07314ee82f121a
Tamaño del documento original: 953,68 kB

Depositante: EMILIO RODRIGO ZHUMA MERA
Fecha de depósito: 18/11/2025
Tipo de carga: interface
fecha de fin de análisis: 18/11/2025

Número de palabras: 13.262
Número de caracteres: 93.077

Ing. Emilio Rodrigo Zhuma Mera, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y DISEÑO DIGITAL

CARRERA DE TELEMÁTICA

PROYECTO DE INVESTIGACIÓN

TÍTULO:

"ARQUITECTURA DE RED DEFINIDA POR SOFTWARE REFERENCIAL PARA UN PROVEEDOR DE SERVICIOS DE INTERNET"

Presentado al Consejo Directivo de la Facultad de Ciencias de la Computación y Diseño Digital como requisito previo a la obtención del título de ingeniero en telemática.

Aprobado por:

PRESIDENTE DEL TRIBUNAL

Ing. Santiago Javier Meneses Narváez, MSc.

MIEMBRO DEL TRIBUNAL

Ing. José Luis Tubay Vergara, MSc.

MIEMBRO DEL TRIBUNAL

Ing. Anthony Limber Morán Cabezas, MSc.

QUEVEDO – LOS RÍOS – ECUADOR

2025

AGRADECIMIENTO

La memoria del corazón es el agradecimiento, siempre he llevado esa frase conmigo, y estoy inmensamente agradecido con Dios por todo lo que me ha brindado a lo largo de mi vida, esto prioriza a mi familia, ya que Él me la dio y han sido mi pilar fundamental para este logro, a mi mamita Geoconda, a mi papito Johnny, a mis ñaños Joyce y Jordy, y a mis pequeños sobrinos Sebastián y Joycita, sin su apoyo constante a lo largo de los años no hubiera podido lograr nada, han sido una luz en mis días más oscuros, pero por ustedes he continuado hasta lograr mis objetivos. A mis demás familiares que siempre han estado pendientes de mi tía Primitiva, mi prima Marola, mi primo Kevin, muchas gracias, así mismo a todos aunque no los nombre específicamente los llevo en mi corazón.

Agradezco de manera enfática a mis docentes, quienes me han guiado durante tantos años para aprender a ser un profesional, a mi tutor de la tesis el ingeniero Emilio Zhuma, estoy muy agradecido no sólo por su tutoría, sino también por paciencia y sus enseñanzas en las carreras de grado. Llevo en el corazón a mis compañeros de clases, gracias por su apoyo, no dudo de que serán profesionales increíbles. Agradezco de corazón a dos amigos Elena y Jhonatan Vasquez quienes me proveyeron materiales electrónicos para realizar mi trabajo de investigación, los llevo en mis oraciones todos los días.

DEDICATORIA

Este trabajo está dedicado a Dios por darme las facultades necesarias para lograr incluso aquello que parecía imposible. A mis amados padres Geoconda y Johnny, quienes siempre supieron proveer lo necesario, tanto económico como emocional, a mis amados hermanos Joyce y Jordy quienes siempre me ayudan con todo lo que hacen y pueden, a mis amados sobrinos Sebastián y Joycita porque sus risas son paz para mi corazón.

Y a todos aquellos que de una u otra manera han contribuido a mi desarrollo profesional, esto es por y para ustedes.

RESUMEN Y PALABRAS CLAVE

Este trabajo propone una arquitectura referencial de red definida por software orientada a un proveedor de servicios de Internet, con el fin de mejorar la escalabilidad, la automatización y la eficiencia operativa frente a arquitecturas tradicionales. Se adoptó un enfoque cuantitativo y un diseño no experimental en entorno controlado, implementando una topología jerárquica (core–distribución–acceso) sobre Mininet y gestionada por el controlador OpenDaylight (Carbon 0.6.4) mediante OpenFlow 1.3. La validación se realizó con pruebas funcionales: conectividad extremo a extremo (ping), throughput (iperf) y captura de tráfico (Wireshark), registrando métricas de latencia, velocidades de descarga/carga y consumo de recursos. Los resultados evidencian una latencia promedio de 2,43 ms y velocidades de 271 Mbps (descarga) y 214 Mbps (carga) en la red SDN, superando los valores observados en dos operadores locales a los que denominaremos (ISP Quevedo 1 e ISP Quevedo 2) para los mismos indicadores, lo que respalda la viabilidad técnica de la propuesta en escenarios representativos de ISP. Se discuten además las limitaciones de la emulación con Mininet y las consideraciones de escalabilidad del controlador, así como la pertinencia de SDN/NFV para futuras extensiones. En conjunto, la evidencia sugiere que la arquitectura SDN propuesta constituye una alternativa sólida para modernizar redes ISP, al ofrecer menor latencia, mayor rendimiento y gestión centralizada con potencial de integración hacia dominios IoT.

Palabras clave: SDN, OpenDaylight, OpenFlow, Mininet, ISP, Virtualización de funciones de red (NFV).

ABSTRACT AND KEYWORDS

This work proposes a reference architecture for Software-Defined Networking aimed at an Internet Service Provider, with the purpose of improving scalability, automation, and operational efficiency compared to traditional architectures. A quantitative approach and a non-experimental design were adopted in a controlled environment, implementing a hierarchical topology (core–distribution–access) on Mininet and managed by the OpenDaylight controller (Carbon 0.6.4) using OpenFlow 1.3. Validation was carried out through functional tests: end-to-end connectivity (ping), throughput (iperf), and traffic capture (Wireshark), recording metrics such as latency, download/upload speeds, and resource consumption. The results show an average latency of 2.43 ms and speeds of 271 Mbps (download) and 214 Mbps (upload) in the SDN network, surpassing the values observed in two local operators referred to as ISP Quevedo 1 and ISP Quevedo 2 for the same indicators, which supports the technical feasibility of the proposal in representative ISP scenarios. The limitations of Mininet emulation and scalability considerations of the controller are also discussed, as well as the relevance of SDN/NFV for future extensions. Overall, the evidence suggests that the proposed SDN architecture constitutes a solid alternative for modernizing ISP networks, offering lower latency, higher performance, and centralized management with potential integration into IoT domains.

Keywords: SDN, OpenDaylight, OpenFlow, Mininet, ISP, Network Functions Virtualization (NFV).

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

CÓDIGO DUBLIN

Título:	Arquitectura de red definida por software referencial para un proveedor de servicios de internet		
Autor:	John Daniel Pincay Murillo		
Palabras claves:	SDN	Mininet	OpenDaylight
Fecha de publicación:	2025		
Editorial:	Quevedo- UTEQ, 2025		
Resumen: (hasta 300 palabras)	Resumen .- Este trabajo propone una arquitectura referencial de red definida por software orientada a un proveedor de servicios de Internet, con el fin de mejorar la escalabilidad, la automatización y la eficiencia operativa frente a arquitecturas tradicionales. Se adoptó un enfoque cuantitativo y un diseño no experimental en entorno controlado, implementando una topología jerárquica sobre Mininet y gestionada por el controlador OpenDaylight (Carbon 0.6.4) mediante OpenFlow 1.3. La validación se realizó con pruebas funcionales: conectividad extremo a extremo, throughput y captura de tráfico, registrando métricas de latencia, velocidades de descarga/carga y consumo de recursos. Los resultados evidencian una latencia promedio de 2,43 ms y velocidades de 271 Mbps (descarga) y 214 Mbps (carga) en la red SDN, superando los valores observados en dos operadores locales a los que denominaremos (ISP Quevedo 1 e ISP Quevedo 2) para los mismos indicadores, lo que respalda la viabilidad técnica de la propuesta en escenarios representativos de ISP. (…)		
Abstract: (hasta 300 palabras)	**Abstract .- This work proposes a reference architecture for Software-Defined Networking aimed at an Internet Service Provider, with the purpose of improving scalability, automation, and operational efficiency compared to traditional architectures. A quantitative approach and a non-experimental design were adopted in a controlled environment, implementing a hierarchical topology on Mininet and managed by the OpenDaylight controller (Carbon 0.6.4) using OpenFlow 1.3. Validation was carried out through functional tests: end-to-end connectivity, throughput, and traffic capture, recording metrics such as latency, download/upload speeds, and resource consumption. The results show an average latency of 2.43 ms and speeds of 271 Mbps (download) and 214 Mbps (upload) in the SDN network, surpassing the values observed in two local operators referred to as ISP Quevedo 1 and ISP Quevedo 2 for the same indicators, which supports the technical feasibility of the proposal in representative ISP scenarios.**		
Descripción:	82 hojas : dimensiones, 29 x 21 cm + CD-ROM 6162		
URI:			

INTRODUCCIÓN

En el contexto actual de las telecomunicaciones, las redes de los proveedores de servicios de Internet (ISP, por sus siglas en inglés) continúan operando bajo arquitecturas tradicionales basadas en protocolos rígidos, lo que limita su escalabilidad, eficiencia y capacidad de adaptación ante la creciente demanda de servicios digitales. Esta situación es especialmente crítica para los ISP, donde la adopción de tecnologías innovadoras como las redes definidas por software (SDN, por sus siglas en inglés) aún es poco utilizada, a pesar de las ventajas que ofrecen en cuanto a automatización, gestión centralizada y optimización del uso de recursos [1]; [2].

En la última década, las redes definidas por software han revolucionado la gestión de infraestructuras de redes a escala global, transformando paradigmas tradicionales mediante la separación del plano de control y los datos. Grandes proveedores de servicios como Google (B4 Network), AT&T (Domain 2.0) y Microsoft Azure han implementado SDN para lograr escalabilidad dinámica, reducción de costos operativos y respuesta ágil a demandas de tráfico. En estudios aplicados en aplicaciones a redes en Ecuador, se ha demostrado que SDN permite mejorar la eficiencia operativa, reducir pérdidas de paquetes y optimizar la latencia en servicios como VoIP [3].

Diversos estudios y reportes técnicos señalan que los principales operadores en Ecuador, como Netlife, CNT y Puntonet, presentan velocidades competitivas, pero aún enfrentan deficiencias en aspectos clave de calidad de servicio, tales como la latencia (24-50 ms) y la estabilidad en regiones periféricas [1]

Estando limitadas por infraestructuras tradicionales basadas en protocolos estáticos (OSPF/BGP) con altos costos de gestión y escalabilidad manual, eventos como la Expo ISP Ecuador destacan la necesidad de incorporar tecnologías avanzadas como SDN para fortalecer la infraestructura, mejorar la competitividad y brindar un servicio más confiable y dinámico [2]. Estas tecnologías podrían ser clave para abordar las limitaciones actuales y garantizar un servicio más eficiente y escalable.

En este marco, el presente proyecto tiene como objetivo proponer una arquitectura de red definida por software referencial para un ISP, utilizando herramientas de simulación como Mininet que será la base para desarrollar la arquitectura de la red, OpenDaylight es el

controlador escogido para la administración de la misma y finalmente quien establece la conexión entre la topología y el controlador será el protocolo OpenFlow para emular un escenario con una gran cantidad de usuarios. Asimismo, se realizará un análisis comparativo de desempeño frente a una infraestructura tradicional, evaluando parámetros como latencia, velocidad de carga y descarga.

El Capítulo I aborda la contextualización del problema, destacando las limitaciones inherentes a las arquitecturas de red tradicionales utilizadas por los proveedores de servicios de Internet, tales como la falta de escalabilidad, la rigidez operativa y los elevados costos de gestión. Se plantea la necesidad de adoptar redes definidas por software como una alternativa tecnológica que permita optimizar la administración de recursos y mejorar la calidad del servicio. Asimismo, se formulan los objetivos de la investigación y se justifica su pertinencia.

En el Capítulo II se desarrolla la fundamentación teórica, que incluye el análisis conceptual de las redes definidas por software, la separación del plano de control y de datos, el papel de los controladores y los protocolos asociados, así como la descripción de herramientas de emulación como Mininet. Además, se revisan investigaciones previas que evidencian la aplicabilidad de SDN en entornos ISP, identificando ventajas, desafíos y tendencias emergentes.

El Capítulo III describe la metodología adoptada, basada en un enfoque cuantitativo y un diseño no experimental. Se detallan los métodos empleados (deductivo y analítico), las fases del proyecto desde la revisión documental hasta la simulación y análisis comparativo y los recursos técnicos utilizados, incluyendo entornos virtualizados, herramientas de monitoreo y plataformas de emulación.

Finalmente, el Capítulo IV presenta los resultados obtenidos en la primera etapa, que comprende la identificación y selección de herramientas, controladores y protocolos adecuados para la implementación de SDN. Se justifica la elección de Mininet como plataforma de emulación y de OpenDaylight como controlador principal el cual se encarga de tomar las decisiones en la red, así como la adopción del protocolo OpenFlow para la comunicación de la switch con el controlador ODL, además se presenta el diseño lógico de la red SDN y las pruebas funcionales para su validación.

CAPÍTULO I

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1. Problema de la investigación

1.1.1. Planteamiento del problema

Las redes tradicionales utilizadas por los proveedores de servicios de Internet presentan limitaciones significativas en su capacidad para responder a las demandas modernas de escalabilidad, flexibilidad, automatización y gestión eficiente. Estas arquitecturas, basadas en protocolos estáticos como OSPF y BGP, requieren configuraciones manuales y dependen de hardware especializado, lo que incrementa los costos operativos y reduce la agilidad en la gestión del tráfico. En un contexto donde el consumo de datos crece exponencialmente por la expansión del Internet de las Cosas, los servicios en la nube y las aplicaciones en tiempo real, estas restricciones se traducen en cuellos de botella que afectan la calidad del servicio y la experiencia del usuario [4]; [5].

La falta de programabilidad y control centralizado en las redes convencionales dificulta la implementación de políticas dinámicas de seguridad y calidad de servicio, lo que incrementa la vulnerabilidad ante ataques y la ineficiencia en la asignación de recursos. Además, la gestión distribuida de dispositivos implica procesos complejos y lentos para la resolución de fallos, lo que impacta directamente en la continuidad del servicio. Esta situación es crítica para los ISP locales, que enfrentan una creciente presión por ofrecer servicios más rápidos y confiables en un mercado altamente competitivo.

De no abordarse estas limitaciones, los proveedores continuarán operando con infraestructuras rígidas y costosas, incapaces de adaptarse a escenarios de alta demanda y nuevas tecnologías como 5G y Edge Computing. Esto no solo compromete la sostenibilidad operativa, sino que también limita la capacidad de innovación y la competitividad. se plantea la necesidad de explorar soluciones basadas en redes definidas por software, que permitan separar el plano de control del plano de datos, habilitando la gestión centralizada, la automatización y la optimización del tráfico en entornos ISP.

Sin embargo, su adopción en ISP locales ha sido lenta debido a factores como la inversión inicial requerida, la falta de recursos humanos capacitados, los desafíos de interoperabilidad con equipos heredados y la percepción de riesgos durante el proceso de migración [5]. Por ejemplo, se ha observado que la transición a redes definidas por software generalmente implica costos elevados de capital y complejas estrategias de implementación incremental que dificultan una adopción inmediata.

1.1.2. Diagnóstico

En la era digital actual, la demanda por servicios de red más rápidos, confiables y flexibles ha incrementado de forma exponencial, impulsada por fenómenos como la expansión del Internet de las Cosas (IoT), el auge de servicios en la nube y el crecimiento del tráfico de datos móviles. Ante este escenario, infraestructuras tradicionales de red, basadas en arquitecturas rígidas y protocolos estáticos, se han mostrado eficientes, sin embargo, para responder a los requerimientos de escalabilidad, gestión dinámica y automatización que exigen los entornos modernos de telecomunicaciones. Estudios recientes sobre virtualización de funciones de red evidencian que la adopción de arquitecturas programables permite superar estas limitaciones mediante la implementación de servicios flexibles sobre infraestructuras virtualizadas [6].

Estudios recientes demuestran que estas redes convencionales enfrentan importantes limitaciones técnicas, como la configuración manual de dispositivos, altos costos de operación, baja visibilidad del tráfico y dificultad para implementar políticas de seguridad consistentes. Estas deficiencias se agudizan en entornos complejos como los de los proveedores de servicios de Internet, donde la administración de miles de dispositivos y conexiones exige un mayor nivel de control y adaptabilidad. Investigaciones recientes sobre arquitecturas híbridas para redes distribuidas han demostrado que la integración de SDN con tecnologías de control de tráfico mejora la seguridad, escalabilidad y capacidad de adaptación en entornos ISP [7].

Frente a estas limitaciones, el paradigma de las redes definidas por software ha emergido como una solución viable para transformar la forma en que las redes son diseñadas, gestionadas y escaladas. SDN propone la separación del plano de control y el plano de datos, permitiendo una gestión centralizada e inteligente de toda la infraestructura de red mediante controladores programables. Esta aproximación no solo permite optimizar el uso de recursos, sino que también facilita la implementación de políticas de calidad de servicio, seguridad, y automatización operativa.

Sin embargo, a pesar de su potencial, la adopción de SDN en América Latina, y particularmente en países como Ecuador, ha sido limitada. Factores como los altos costos iniciales, la dependencia de infraestructuras heredadas, la escasa formación técnica en el área y la falta de estudios locales que demuestren su viabilidad, han frenado su implementación en el entorno real de los ISP regionales [4].

En Ecuador, informes como el Barómetro de Conectividad de [1] revelan que, aunque los principales proveedores como Netlife, Puntonet y Claro han mejorado su rendimiento en velocidad de descarga y latencia, aún existen amplias brechas de desempeño, especialmente en zonas periféricas. Asimismo, se identifican carencias en la flexibilidad de red y tiempos de respuesta ante fallos, lo que sugiere la necesidad de incorporar tecnologías más dinámicas y automatizadas.

Ante la falta de estudios prácticos y accesibles sobre la aplicabilidad de SDN en redes locales, se plantea la necesidad de desarrollar una simulación de una arquitectura referencial SDN adaptada a una agencia ISP ecuatoriana. Esta simulación permitirá evaluar métricas clave de rendimiento, como la latencia, el uso del ancho de banda, la eficiencia operativa y los costos asociados, ofreciendo así una base empírica para futuras implementaciones reales y replicables en las distintas regiones del país.

1.1.3. Pronóstico

Las redes definidas por software están transformando radicalmente las telecomunicaciones a nivel global, con proyecciones de crecimiento que evidencian su papel crítico en el futuro de los proveedores de servicios de Internet. Según Grand View [8], el mercado global de SDN alcanzará USD 152.21 mil millones para 2033, impulsado por una tasa de crecimiento anual compuesto (CAGR) del 26.6% desde 2024. Este crecimiento será liderado por la demanda de redes escalables para 5G, IoT y computación en la nube, donde SDN reduce costos operativos (OPEX) hasta en un 40% mediante la automatización de la gestión de recursos y la optimización dinámica del tráfico.

En América Latina, la adopción de SDN en ISP acelerará significativamente hacia 2030, con un CAGR del 28% [9]. Este impulso responderá a la necesidad crítica de reducir latencias en servicios sensibles (telemedicina, vehículos autónomos), donde SDN permitirá alcanzar valores inferiores a 5 ms mediante el edge computing y el network slicing (segmentación de redes). Estudios de IEEE confirman que la integración de inteligencia artificial (IA) en controladores SDN optimizará la predicción de congestión y la detección de amenazas en tiempo real, mejorando la resiliencia de las redes.

Para ISP locales en Ecuador esta transición representa una oportunidad estratégica. Proyecciones basadas en modelos de simulación nos indican que migrar a arquitecturas SDN podría reducir la latencia actual (24-50 ms) a menos de 15 ms en 5 años, acercándose a estándares globales.

Además, tecnologías como NFV (Virtualización de Funciones de Red) permitirán ofrecer servicios bajo demanda (ej. ancho de banda ajustable), generando nuevos flujos de ingresos. Persisten desafíos técnicos, como la vulnerabilidad de controladores centralizados a ataques DDoS, que requerirán soluciones híbridas con blockchain o SDP (Software-Defined Perimeter). No obstante, herramientas de simulación como ONOS o OpenDaylight están reduciendo los costos de pruebas en un 60%, facilitando la adopción en ISP medianos.

Las redes SDN no son una tendencia pasajera, sino el núcleo de las telecomunicaciones del futuro. Para 2030, se prevé que 75% de los ISP operarán con arquitecturas SDN/NFV, combinando programabilidad, inteligencia artificial y automatización. En Ecuador, este proyecto proporciona un modelo cuantitativo para mostrar la fiabilidad de esta transición, demostrando mediante simulación cómo SDN puede transformar la infraestructura local en redes eficientes, escalables y competitivas a nivel global.

1.1.4. Formulación del problema

¿Cómo diseñar y validar un modelo referencial de arquitectura de red definida por software en un entorno supervisado que permita demostrar mejoras en el rendimiento y la flexibilidad operativa frente a una red tradicional, y que pueda servir como base a los proveedores de servicios de Internet para optimizar la calidad de su servicio?

1.1.5. Sistematización del problema

- ¿Qué herramientas, controladores y protocolos SDN son viables para diseñar un modelo referencial aplicable a un ISP?
- ¿Cómo implementar un modelo referencial de arquitectura de red definida por software en un entorno supervisado, considerando las herramientas, controladores y protocolos más adecuados?
- ¿Qué mejoras cuantificables en latencia y flexibilidad operativa demuestra el modelo SDN simulado frente a redes tradicionales bajo escenarios exigentes?

1.2. Objetivos

1.2.1. Objetivo general

Diseñar un modelo de arquitectura de red definida por software de carácter referencial, en un entorno supervisado, que sirva como base para proveedores de servicios de Internet.

1.2.2. Objetivos específicos

- Identificar herramientas, controladores y protocolos en las redes definidas por software mediante un estudio bibliográfico, para seleccionar las más adecuadas en el diseño del modelo referencial propuesto.
- Implementar un modelo de una arquitectura de red definida por software, adaptada a necesidades y desafíos de un proveedor de servicios de internet en un ambiente controlado.
- Validar el rendimiento del modelo de arquitectura de red definida por software mediante pruebas funcionales en un entorno supervisado, para demostrar sus ventajas en términos de latencia y flexibilidad operativa frente a una red tradicional.

1.3. Justificación

Este proyecto se basa en la necesidad de transformar las redes de los proveedores de servicios de Internet para que sean más eficientes, escalables y adaptables a los desafíos y exigencias del entorno digital actual. La adopción de arquitecturas de red definidas por software se presenta como una solución clave para estos desafíos, dado que ofrece una mayor flexibilidad, y automatización control centralizado sobre las redes, aspectos fundamentales para mejorar la calidad del servicio y reducir los costos operativos. La virtualización de funciones de red, como complemento de SDN, permite implementar servicios dinámicos sobre plataformas virtualizadas, optimizando el uso de recursos y reduciendo la dependencia de hardware físico [6].

En Ecuador, los ISP locales enfrentan una creciente demanda de servicios de Internet de alta calidad y una infraestructura que no siempre está a la altura de estas expectativas. Las redes tradicionales, que aún dominan el mercado, presentan limitaciones en cuanto a escalabilidad y capacidad para adaptarse rápidamente a nuevas necesidades tecnológicas, como el incremento de dispositivos conectados y el uso intensivo de datos.

La implementación de redes definidas por software, que centraliza el control y la gestión de la red, permite una administración más eficiente de los recursos y facilita la implementación de nuevas tecnologías, como la virtualización de funciones de red y la optimización del ancho de banda La arquitectura híbrida propuesta por [7] demuestra cómo la combinación de SDN con tecnologías emergentes permite una gestión más eficiente del tráfico, mayor seguridad y flexibilidad operativa en redes ISP.

Además, la adopción de redes definidas por software no solo responde a la necesidad de mejorar la infraestructura tecnológica, sino que también permite a los ISP locales competir de manera más efectiva con proveedores internacionales. Según un informe de la [10] **,** la implementación de redes definidas por software en redes de ISP ha demostrado ser una estrategia efectiva para reducir los costos operativos, minimizar los tiempos de inactividad y mejorar la experiencia del usuario final. En el contexto regional, donde los operadores actuales enfrentan problemas con la latencia y la calidad del servicio.

La simulación de una arquitectura redes definidas por software para un ISP, utilizando herramientas como Mininet, es una alternativa viable para evaluar las mejoras potenciales sin la necesidad de realizar una inversión inicial en equipos reales. Estas herramientas permiten emular redes complejas, proporcionando una plataforma de prueba para experimentar con distintos ajustes y evaluar métricas críticas como latencia, rendimiento y costos operativos [11].Al realizar esta simulación, se podrá comparar de manera objetiva las redes tradicionales con la arquitectura de redes definidas por software, lo que proporcionará información valiosa sobre los beneficios tangibles de migrar hacia una infraestructura más moderna y eficiente.

Este estudio no solo tiene implicaciones para una agencia ISP específica, sino que también contribuye al campo académico y técnico al proporcionar un modelo práctico y accesible para la adopción de redes definidas por software en entornos de telecomunicaciones locales. Además, se presenta como un modelo replicable para otras regiones del Ecuador, que podrían beneficiarse de los mismos avances en infraestructura tecnológica.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

2.1. Marco conceptual

2.1.1. Red definida por software (SDN)

Las redes definidas por software son un paradigma emergente en la ingeniería de redes que propone la separación del plano de control y el plano de datos, permitiendo que el comportamiento de la red sea programable y gestionado centralmente a través de controladores lógicos. Esta arquitectura facilita la automatización, la administración dinámica del tráfico, la virtualización de funciones y la adaptación a cambios de demanda. En entornos ISP, estas ventajas han sido validadas mediante pruebas funcionales con servicios de VoIP sobre SDN [3].

2.1.2. Plano de control y plano de datos

En una red tradicional, cada dispositivo (switch, router) toma decisiones de enrutamiento (control) y también reenvía paquetes (datos). SDN separa estas funciones: el plano de control se centraliza en un controlador lógico, mientras que los planos de datos en los dispositivos ejecutan las decisiones sin procesar lógicas complejas. Esta separación incrementa la eficiencia operativa y la capacidad de reconfiguración dinámica. Según la revista [12], esta evolución arquitectónica ha sido clave para transformar redes tradicionales en infraestructuras programables, escalables y adaptables a los nuevos desafíos tecnológicos.

2.1.3. Controlador SDN

Entre los controladores más conocidos se encuentran ONOS, OpenDaylight y Ryu, ampliamente evaluados en entornos académicos e industriales. Según [13], estos controladores presentan diferencias significativas en rendimiento, escalabilidad y latencia, siendo OpenDaylight el más adecuado para redes densas como las de los ISP.

2.1.3.1. Controlador ONOS (Open Network Operating System).

ONOS es un controlador SDN de código abierto diseñado para soportar redes de gran escala, con alta disponibilidad y tolerancia a fallos. Está orientado principalmente a proveedores de servicios y operadoras, y permite gestionar miles de dispositivos de red de forma centralizada, segura y eficiente.

2.1.3.2. Controlador Ryu.

Ryu es un controlador SDN ligero y fácil de usar, desarrollado en Python. Está enfocado en la investigación y la educación, permitiendo a los usuarios crear aplicaciones de control de red personalizadas. Es especialmente útil en entornos de prueba y simulación como Mininet.

2.1.3.3. Controlador OpenDaylight.

OpenDaylight es otro controlador SDN de código abierto ampliamente utilizado en entornos empresariales y académicos. Su arquitectura modular permite integrar diversas tecnologías de red y soporta múltiples protocolos, como OpenFlow, NETCONF y BGP. Es conocido por su flexibilidad y capacidad de personalización.

2.1.4. Emulación de red

La emulación de red es un proceso que permite recrear el comportamiento de una red real dentro de un entorno virtual. A diferencia de la simulación (que solo modela el tráfico y el rendimiento), la emulación permite ejecutar software real sobre redes virtuales, como se hace con Mininet. Esto facilita el desarrollo, prueba y validación de arquitecturas SDN sin necesidad de equipos físicos.

2.1.5. Protocolo de red

Un protocolo de red es un conjunto de reglas que definen cómo los dispositivos de una red se comunican entre sí. Estos protocolos especifican cómo deben formarse los paquetes, cómo se envían, reciben y verifican. Ejemplos comunes incluyen IP**,** TCP**,** UDP**,** OSPF**, y** BGP. En redes SDN, los protocolos también incluyen interfaces como OpenFlow**,** que permiten a los controladores comunicarse con los dispositivos de red.

2.1.6. API (Interfaz de Programación de Aplicaciones)

Una API es un conjunto de funciones y definiciones que permiten la comunicación entre diferentes componentes de software. En el contexto de SDN, se utilizan APIs para permitir que las aplicaciones se comuniquen con el controlador (Northbound APIs) o que el controlador interactúe con los switches o routers (Southbound APIs). Estas interfaces facilitan la programación de políticas de red personalizadas.

2.1.7. Proveedor de Servicios de Internet (ISP)

Un ISP es una entidad que brinda acceso a Internet y otros servicios relacionados, como alojamiento de sitios web, correo electrónico o servicios de transmisión de datos. Su infraestructura se compone de dispositivos de red, centros de datos, servidores de autenticación y enlaces troncales, siendo altamente sensible a fallos, latencia y saturación. La implementación de SDN en este tipo de organizaciones permite optimizar el uso del ancho de banda, reducir tiempos de respuesta y aplicar políticas dinámicas de tráfico [4].

2.1.8. Virtualización de funciones de red (NFV)

NFV es una tecnología complementaria a SDN que permite la implementación de funciones de red (firewalls, balanceadores de carga, routers) como software sobre infraestructuras virtualizadas, en lugar de dispositivos físicos dedicados. Su combinación con SDN permite una red completamente programable, escalable y automatizada. [14]

2.1.9. Mininet

Mininet es una herramienta de emulación de redes de código abierto que permite crear redes virtuales SDN en un solo equipo físico. Permite el diseño, simulación y prueba de arquitecturas SDN, soportando controladores como POX, Ryu u OpenDaylight. Es ampliamente utilizado en investigaciones académicas y pruebas experimentales [15].

2.1.10. Latencia

La latencia es el tiempo que tarda un paquete de datos en viajar desde su origen hasta su destino. Es una métrica crítica para evaluar el rendimiento de la red, especialmente en aplicaciones en tiempo real. SDN permite optimizar la latencia al implementar rutas más eficientes, gestionar la congestión y priorizar tráfico crítico mediante políticas programables.

2.2.11. Wireshark

Wireshark es una herramienta de análisis de protocolos de red de código abierto que permite capturar y examinar el tráfico en tiempo real, proporcionando visibilidad detallada de los paquetes transmitidos en una red. Es ampliamente utilizada en entornos académicos e industriales para diagnóstico, pruebas de rendimiento y análisis forense, gracias a su compatibilidad con múltiples protocolos y su interfaz gráfica intuitiva [16].

2.2.12. Evolución reciente de SDN

La evolución de las redes definidas por software durante la última década ha estado marcada por la madurez de su arquitectura y la ampliación de sus aplicaciones en entornos de operadores. El plano de control centralizado, que antes se consideraba experimental, hoy se integra en redes de producción gracias a mejoras en la estabilidad de los controladores, la escalabilidad de los protocolos y la capacidad de automatización que demandan los proveedores de servicios. Los trabajos recientes destacan que SDN se ha convertido en una pieza clave para la transformación hacia redes de nueva generación, permitiendo una mayor flexibilidad y una gestión más óptima del tráfico [17].

2.2.13. Topología de red

La topología de red se refiere a la disposición física o lógica de los nodos y enlaces que conforman una red de comunicaciones. Define cómo se interconectan los dispositivos y cómo fluye la información entre ellos, influyendo directamente en la escalabilidad, tolerancia a fallos y rendimiento del sistema. Entre las topologías más comunes se encuentran la estrella, anillo, malla y jerárquica.

2.2.14. Arquitecturas híbridas SDN

La adopción de SDN híbrido surge como respuesta a la necesidad de integrar infraestructuras tradicionales con arquitecturas controladas por software. En este enfoque conviven dispositivos heredados y elementos programables, lo que permite a los proveedores migrar de forma gradual sin comprometer la operación. La literatura reciente señala que los modelos híbridos proporcionan un equilibrio entre innovación y compatibilidad, facilitando etapas de transición que minimizan riesgos operativos y permiten validar nuevas funcionalidades en dominios controlados antes de expandirlas al resto de la red [18].

2.2.15. Comparación y rendimiento de controladores SDN

La elección del controlador SDN es un factor determinante para el rendimiento de la red, ya que este componente coordina la lógica de enrutamiento y la administración del tráfico. Estudios comparativos recientes analizan parámetros como latencia, tiempo de respuesta y consumo de recursos, demostrando que cada controlador presenta ventajas diferentes según el escenario. Por ejemplo, algunos ofrecen mayor escalabilidad en topologías complejas, mientras que otros destacan por su rapidez en configuraciones de baja carga. Este tipo de análisis es fundamental para proveedores que buscan optimizar sus servicios a partir de un controlador adecuado [19] [13].

2.2.16. Limitaciones y fidelidad de Mininet

Mininet se ha consolidado como una herramienta esencial para la emulación de redes SDN debido a su facilidad de uso y su capacidad para replicar topologías complejas. Sin embargo, diversos autores indican que sus resultados deben interpretarse con cautela, ya que la ejecución en un solo host limita el rendimiento real de los switches y enlaces emulados. Además, la emulación no reproduce completamente el comportamiento del hardware especializado. Aun así, su versatilidad lo mantiene como una opción válida para pruebas preliminares y prototipos de investigación [20].

2.2.17. Benchmarking de controladores con Cbench / Mininet

El benchmarking de controladores SDN utilizando herramientas como Cbench y Mininet es esencial para evaluar la capacidad de respuesta del plano de control frente a cargas crecientes de peticiones. Los estudios basados en estas herramientas muestran que la cantidad de solicitudes por segundo que un controlador puede procesar varía significativamente entre implementaciones, lo que influye en su idoneidad para diferentes escenarios de producción. Esta metodología experimental se ha vuelto estándar en investigaciones orientadas a caracterizar el comportamiento y los límites operativos de los controladores actuales [21].

2.2.18. Rendimiento de SDN en topologías reales

Aunque los entornos emulados ofrecen flexibilidad, las pruebas en topologías reales revelan comportamientos que no siempre se observan en simulaciones. Investigaciones recientes destacan que los controladores presentan variaciones en latencia, estabilidad y escalabilidad cuando se despliegan en redes heterogéneas. Esto evidencia la importancia de evaluar soluciones SDN en escenarios que representen situaciones reales de los proveedores, incluyendo enlaces intercontinentales, múltiples dominios administrativos y cargas dinámicas de tráfico [22].

2.2.19. Seguridad en redes SDN

La seguridad es uno de los aspectos más críticos en la adopción de SDN, especialmente debido al rol centralizado del controlador. Los análisis recientes catalogan amenazas como la saturación del plano de control, la manipulación de reglas de flujo y ataques de denegación de servicio que afectan la disponibilidad del controlador. Las soluciones propuestas incluyen segmentación del plano de control, autenticación robusta, análisis del tráfico controlado y detección temprana de anomalías mediante algoritmos especializados, elementos clave para un entorno seguro [23].

2.2.20. Detección de intrusiones basada en SDN

La integración de técnicas de detección de intrusiones dentro de redes SDN ha avanzado significativamente gracias al uso del controlador como punto central de observación del tráfico. Este enfoque permite aplicar análisis inteligentes para identificar flujos sospechosos de forma más rápida y precisa que en arquitecturas tradicionales. La literatura reciente evidencia que SDN facilita la implementación de mecanismos de respuesta adaptativa, capaces de aislar segmentos afectados, modificar rutas y aplicar políticas de seguridad sin intervención manual [24].

2.2.21. SDN y reproducibilidad experimental

En investigaciones de redes, la reproducibilidad es esencial para validar resultados y comparar arquitecturas. SDN, al apoyarse en controladores y emuladores como Mininet, permite definir entornos experimentales que pueden ser replicados mediante versiones específicas de software, kernels y dependencias. Los últimos reportes de desarrollo destacan mejoras en estabilidad, compatibilidad con versiones modernas de sistemas operativos y soporte actualizado que facilita la experimentación científica [25].

2.2.22. Integración de SDN con NFV

La convergencia entre SDN y la virtualización de funciones de red se ha convertido en una estrategia clave para los proveedores de servicios que buscan automatizar y escalar sus operaciones. SDN permite un control centralizado, mientras que NFV posibilita desplegar funciones de red en entornos virtualizados de manera dinámica. Los estudios recientes muestran que esta combinación mejora la eficiencia operativa y reduce los costos en infraestructuras de gran escala, especialmente en contextos "carrier-grade" [26].

2.2.23. SDN aplicado a redes de operadores

Los operadores de telecomunicaciones enfrentan desafíos como la gestión de redes complejas, la variabilidad del tráfico y la necesidad de ofrecer servicios de alta disponibilidad. SDN se presenta como una solución viable al reducir la dependencia del hardware tradicional y permitir una gestión más flexible del plano de control. Investigaciones en entornos ISP demuestran que la adopción de SDN puede mejorar la capacidad de adaptación a cambios rápidos en la demanda y optimizar la utilización de recursos de red [26].

2.2. Marco referencial

En el desarrollo de este proyecto se ha tomado como referencia los siguientes trabajos investigativos:

- En [3] los autores presentan una implementación de redes definidas por software para servicios de Voz sobre IP (VoIP) en un entorno de proveedor de servicios de Internet. Los autores desarrollan una arquitectura basada en OpenDaylight, evaluando métricas como latencia, pérdida de paquetes y estabilidad de red.

Los resultados demuestran que SDN mejora significativamente el rendimiento de servicios sensibles como VoIP, al permitir una gestión centralizada, reducción de retardo y mayor seguridad en la transmisión de datos. Este trabajo valida la aplicabilidad de SDN en escenarios reales de ISP en Ecuador, y refuerza su potencial como solución tecnológica para optimizar la calidad del servicio.

- El [4] se analizan los desafíos y estrategias para la migración hacia redes definidas por software en entornos de proveedores de servicios de Internet. El autor identifica que la transición completa a SDN en redes ISP enfrenta obstáculos significativos, como restricciones financieras, complejidad operativa y riesgos asociados a la interoperabilidad con equipos heredados. Entre sus hallazgos, destaca que la adopción de arquitecturas híbridas —combinando SDN con infraestructura tradicional— puede ser una solución viable para reducir costos y minimizar interrupciones durante el proceso de migración. Este aporte resulta relevante para la presente investigación, ya que refuerza la necesidad de validar la factibilidad técnica de SDN en escenarios controlados antes de su implementación real, lo que justifica el enfoque experimental adoptado en este proyecto.

- En [13] se realiza una evaluación comparativa del rendimiento de tres controladores SDN de código abierto: Floodlight, OpenDaylight y Ryu. Utilizando la herramienta Cbench en un entorno emulado con Mininet, se analizan métricas como latencia, throughput y escalabilidad. Los resultados muestran que OpenDaylight es el más adecuado para redes densas como las de los ISP, mientras que Floodlight se recomienda para redes pequeñas. Este trabajo proporciona una base técnica sólida para la selección de controladores en entornos SDN aplicados a proveedores de servicios de Internet.

- En [27] se comparan los controladores SDN OpenDaylight (ODL) y ONOS, destacando sus capacidades en entornos virtuales simulados con Mininet. Los autores concluyen que OpenDaylight ofrece un rendimiento superior en términos de escalabilidad, compatibilidad con múltiples protocolos y flexibilidad de integración, lo que lo convierte en una opción robusta para entornos empresariales, ISP y redes 5G.

Además, se resalta su arquitectura modular y el respaldo de la Linux Foundation como factores clave para su adopción en proyectos de investigación y producción. Este estudio refuerza la elección de ODL como controlador principal en esta tesis, al demostrar su aplicabilidad en escenarios complejos y su alineación con estándares internacionales.

- El trabajo de [28] analiza las capacidades de Mininet como herramienta de emulación para redes definidas por software. Aunque reconocen que Mininet es una solución ligera, accesible y ampliamente utilizada en entornos académicos, los autores advierten sobre limitaciones importantes en cuanto a seguridad, aislamiento de nodos virtuales y precisión en la gestión de recursos. El estudio demuestra que, en escenarios complejos, Mininet puede generar resultados poco confiables si no se complementa con capas físicas o configuraciones avanzadas. Esta evaluación crítica permite contextualizar el uso de Mininet en esta tesis como una herramienta válida para simulaciones controladas, pero con conciencia de sus restricciones técnicas, especialmente en lo que respecta a entornos de producción o pruebas de seguridad.

- [29] realizó una revisión sistemática sobre la integración de tecnologías emergentes como SDN, Edge Computing (EC) e Internet de las Cosas (IoT), destacando su papel en el diseño de redes inteligentes, escalables y adaptables. El estudio identifica que SDN, al ofrecer control centralizado y separación del plano de control y datos, permite gestionar eficientemente redes heterogéneas, optimizar el uso de recursos y mejorar la interoperabilidad en entornos distribuidos. Esta visión refuerza la aplicabilidad de SDN en escenarios ISP modernos, donde la demanda de servicios en tiempo real y la diversidad de dispositivos requieren arquitecturas flexibles y automatizadas. La propuesta de este proyecto se alinea con estas tendencias, al simular una arquitectura SDN que puede ser extendida hacia entornos IoT y edge computing en futuras investigaciones.

- Además [30] nos habla de que internet de las cosas conecta millones de máquinas, vehículos, nodos, detectores de humo, relojes, gafas, cámaras web y otros dispositivos a Internet. Estas entidades necesitan la guía y el control adecuados para el rendimiento esperado. Sin embargo, gestionar todas estas entidades no es fácil; siempre es una gran preocupación. Todos los tipos de arquitecturas de red se mejoran a diario, y el proceso de gestión de red tradicional se vuelve más complejo.

- El estudio de [26] presenta una implementación práctica del modelo SDN utilizando el emulador Mininet y VirtualBox. Los autores despliegan una red virtual con cuatro máquinas y un script en Python para definir reglas de comunicación, demostrando que la administración de SDN es sencilla, directa y eficiente. Este trabajo valida el uso de Mininet como herramienta de emulación en entornos académicos y de investigación, reforzando su aplicabilidad en proyectos ISP.

- El estudio de [31] analiza la viabilidad de implementar redes definidas por software en entornos ISP de nivel carrier-grade, mediante la evaluación del rendimiento de enrutamiento en un entorno híbrido multi-dominio. Los autores comparan el desempeño de redes tradicionales (legacy) con redes SDN utilizando el controlador ONOS y la aplicación SDN-IP, que permite interoperabilidad con redes heredadas a través del protocolo BGP. La simulación se realizó en Mininet, generando tráfico con herramientas distribuidas para medir métricas como latencia, ancho de banda y tasa de transmisión de paquetes. Los resultados muestran que SDN-IP supera al enrutamiento tradicional en términos de rendimiento, especialmente en latencia y capacidad de ancho de banda. El estudio concluye que la migración hacia SDN en redes ISP es viable y progresiva, y que el enfoque híbrido facilita una transición gradual sin comprometer la operación de servicios existentes.

2.3. Marco legal

Este proyecto se fundamentó en la constitución de la República del Ecuador emitida por la [32] establece específicamente en el artículo 16 que:

> *[Artículo 16.- Todas las personas, en forma individual o colectiva, tienen derecho a:*
>
> *"Una comunicación libre, intercultural, incluyente, diversa y participativa, en todos los ámbitos de la interacción social, por cualquier medio y forma, en su propia lengua y con sus propios símbolos."*
>
> *"El acceso universal a las tecnologías de información y comunicación."*
>
> *"La creación de medios de comunicación social, y al ejercicio libre y la potestad de fundarlos, mantenerlos y gestionarlos, sin más limitaciones que las previstas en la Constitución y la ley."*

> *"El acceso y uso de todas las formas de comunicación visual, auditiva, sensorial y a otras que posibiliten la inclusión de personas con discapacidad"].*

Además, otros artículos citados a continuación rigen también este aspecto y mencionan que:

> *[*" [32], *Artículo 314: El Estado será responsable de la provisión de los servicios públicos de agua potable y de riego, saneamiento, energía eléctrica, telecomunicaciones, vitalidad, infraestructuras portuarias y aeroportuarias, y lo demás que determine la ley"].*

Dentro de la [33] (LOTC) especifica lo siguiente:

> *["Artículo 3.3: Incentivar el desarrollo de la industria de productos y servicios de telecomunicaciones"].*
>
> *["Artículo 3.4: Promover y fomentar la convergencia de redes, servicios y equipos."* [33]*].*

Actualmente, aunque la legislación ecuatoriana busca modernizar las telecomunicaciones y garantizar que más personas tengan acceso a los servicios digitales, todavía no existe una normativa específica para tecnologías tan nuevas como las redes definidas por sofware. Esto significa que la forma en que se maneja el tráfico o se controla la red de manera centralizada en un entorno de tiempo real no está regulada de forma directa.

Sin embargo, el país ha puesto en marcha iniciativas importantes, como el plan "Ecuador Digital". Este proyecto tiene como meta impulsar la tecnología y la digitalización, promoviendo el desarrollo de infraestructuras avanzadas como la red 5G y la transformación de los servicios.

Con la creciente demanda de servicios en tiempo real (como en videollamadas, streaming o aplicaciones de voz), es evidente que necesitamos soluciones que hagan nuestras redes más eficientes. Las SDN son una respuesta perfecta a esta necesidad, ya que separan la lógica de control del hardware físico, permitiendo una gestión más dinámica y programable.

CAPÍTULO III

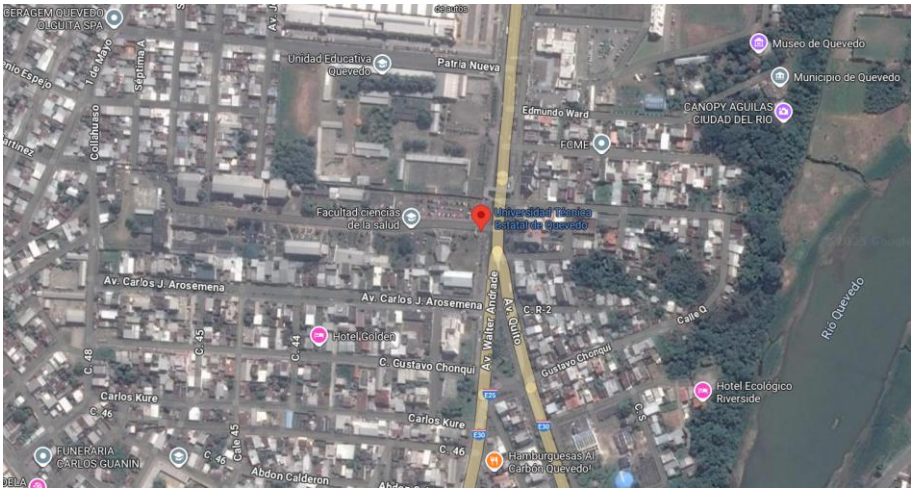
METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Localización

El proyecto de investigación se llevó a cabo en la Av. Quito km. 1 1/2 vía a Santo Domingo de los Tsáchilas del cantón Quevedo, provincia de Los Ríos, Ecuador, ubicación correspondiente al campus "Ing. Manuel Agustín Haz Álvarez", de la Universidad Técnica Estatal de Quevedo, con coordenadas geográficas de latitud -1° 00′ 46″ S, y longitud -79° 28′ 09″ O, a una altitud de 81 metros sobre el nivel del mar.

Figura 1

Mapa del Lugar de Desarrollo de la Investigación



Nota. Se observa la ubicación del lugar de estudio y desarrollo del proyecto investigativo.

3.2. Tipo de investigación.

La investigación desarrollada se clasifica como aplicada, ya que busca ofrecer una solución práctica a un problema real: la optimización de la infraestructura de un proveedor de servicios de Internet mediante la implementación de una arquitectura de red definida por software. Este tipo de investigación se orienta a generar conocimiento útil para resolver necesidades concretas en el ámbito tecnológico y operativo de los ISP.

El enfoque metodológico adoptado es cuantitativo, dado que se fundamenta en la recolección y análisis de datos numéricos obtenidos a través de simulaciones controladas. Las métricas evaluadas incluyen latencia, uso de ancho de banda, pérdida de paquetes y eficiencia operativa, lo que permite validar objetivamente el rendimiento del modelo propuesto.

3.2.1. Investigación aplicada

La investigación aplicada se caracteriza por su orientación hacia la solución de problemas concretos en contextos reales. En este proyecto se aplicó los principios de las redes definidas por software para mejorar la infraestructura de un proveedor de servicios de Internet, abordando desafíos como la escalabilidad, la automatización y la eficiencia operativa. Esto permitió implementar y validar soluciones prácticas que puedan ser replicadas en entornos similares.

3.3. Métodos de investigación

3.3.1. Método cuantitativo

Este proyecto utilizó el método cuantitativo, ya que se enfocó en la recolección y análisis de datos numéricos para comparar el comportamiento de una red tradicional frente a una red definida por software. A través de simulaciones computacionales realizadas con la herramienta Mininet, se recopilaron métricas específicas como latencia**,** uso del ancho de banda **y** eficiencia operativa. Estos datos permitieron evaluar de forma objetiva el impacto de la arquitectura propuesta, facilitando la interpretación. El método cuantitativo fue esencial para garantizar resultados verificables y replicables, adecuados para la validación técnica de la propuesta.

3.3.2. Método deductivo

Se aplicó el método deductivo al partir de principios generales y teorías existentes sobre redes definidas por software, obtenidas de literatura científica y técnica indexada en IEEE, Scopus y otras fuentes confiables. A partir de este marco teórico, se formuló una hipótesis que proponía que las arquitecturas SDN ofrecerían mejores resultados operativos en comparación con las infraestructuras tradicionales. Posteriormente, se diseñaron simulaciones controladas para verificar un caso específico: una agencia ISP con una red simulada con muchos suscriptores. Este razonamiento permitió vincular el conocimiento teórico con la práctica aplicada, siguiendo una lógica de validación descendente.

3.3.3. Método analítico

El enfoque analítico fue utilizado para descomponer el fenómeno de estudio en sus componentes clave: estructura de red, protocolos utilizados, controladores, métricas de rendimiento y mecanismos de gestión. Cada uno de estos elementos fue analizado por separado para identificar sus funciones, ventajas y limitaciones.

Esta descomposición permitió entender las interacciones internas del sistema y facilitó una evaluación detallada de las mejoras alcanzadas mediante la implementación del modelo propuesto. Los resultados se interpretaron de forma crítica, estableciendo relaciones causa-efecto entre las características de la arquitectura y los valores obtenidos en las simulaciones.

3.4. Diseño de investigación

3.4.1. Diseño no experimental

La investigación adopta un enfoque no experimental, ya que las variables técnicas (latencia, throughput, pérdida de paquetes, velocidad de carga y descarga de datos) se registran tal como se presentan en el entorno virtual sin manipulación directa. Las simulaciones se ejecutan en Mininet, replicando escenarios representativos de un ISP con miles de abonados. Este diseño permite observar el comportamiento natural de la arquitectura SDN frente a una red tradicional, identificando patrones de rendimiento y eficiencia operativa bajo distintas cargas de tráfico.

3.4.2. Diseño bibliográfico

Paralelamente, se aplicó un diseño bibliográfico para fundamentar conceptualmente el proyecto. Se recopilaron artículos científicos, informes técnicos y estudios de caso sobre redes definidas por software, controladores SDN, protocolos de comunicación y herramientas de emulación. Esta revisión permitió construir el marco teórico, seleccionar los componentes adecuados y establecer criterios de evaluación. El respaldo documental garantiza la validez académica del modelo propuesto y facilita su replicabilidad en futuras investigaciones.

3.5. Fuentes de recopilación de información

3.5.1. Fuentes primarias

Las fuentes primarias corresponden a los datos obtenidos directamente durante la simulación y evaluación del modelo de arquitectura de red definida por software. Estas pruebas se realizaron en entornos virtuales utilizando Mininet, donde se emularon escenarios representativos de un ISP con miles de abonados. Se recopilaron métricas clave como latencia, pérdida de paquetes, throughput y consumo de recursos, los datos fueron recopilados mediante comandos ejecutados en la interfaz de Mininet, adicionalmente se utilizó Wireshark para monitorear el comportamiento de la red filtrando paquetes ICMPV6 y TCP. Esta información permitió realizar un análisis cuantitativo del comportamiento de la red SDN frente a una arquitectura tradicional.

3.5.2. Fuentes secundarias

Las fuentes secundarias se obtuvieron mediante una revisión bibliográfica exhaustiva. Se consultaron artículos científicos indexados en bases como IEEE Xplore, Scopus, Springer y ScienceDirect, además de tesis académicas, libros técnicos y documentación oficial de herramientas SDN. Esta revisión permitió sustentar teóricamente el diseño de la arquitectura, comprender el funcionamiento de los controladores, justificar el uso de protocolos como OpenFlow y establecer criterios para la comparación entre modelos de red. El respaldo documental también aportó datos sobre tendencias internacionales, casos de aplicación en ISP y desafíos técnicos frecuentes en la implementación de SDN.

3.6. Estructura de la investigación

3.6.1. Fase 1: Revisión documental y conceptualización de SDN

En esta fase se realizó un estudio teórico sobre las redes definidas por software, su arquitectura, funcionamiento, ventajas y desafíos frente a las redes tradicionales. Se consultaron fuentes académicas indexadas, informes técnicos y documentación oficial de herramientas como Mininet y OpenDaylight. Esta revisión permitió construir el marco conceptual, identificar protocolos clave como OpenFlow y comprender los elementos fundamentales para el diseño del modelo propuesto.

3.6.2. Fase 2: Diseño de la arquitectura y planificación del entorno simulado

Con base en la revisión teórica, se diseñó una arquitectura referencial SDN para un entorno representativo de un ISP. Se definieron los componentes principales del modelo (controlador, switches, hosts, topología jerárquica) y se establecieron los parámetros técnicos a evaluar. Se planificó el entorno de simulación utilizando Mininet, asegurando que la configuración permitiera replicar el comportamiento de una red real bajo condiciones controladas.

3.6.3. Fase 3: Simulación, ejecución de pruebas y recopilación de datos

En esta fase se implementaron las dos arquitecturas de red (tradicional y SDN) en escenarios comparables. Se ejecutaron pruebas funcionales bajo distintas condiciones de tráfico, utilizando herramientas como Wireshark y scripts automatizados. Se recolectaron métricas como latencia, throughput, pérdida de paquetes y consumo de recursos, lo que permitió evaluar objetivamente el rendimiento de cada arquitectura.

3.6.4. Fase 4: Análisis comparativo y validación de resultados

Finalmente, se analizaron los datos obtenidos durante las simulaciones. Se elaboraron tablas y gráficos comparativos para identificar diferencias de desempeño entre ambas arquitecturas. El análisis permitió validar que el modelo SDN propuesto ofrece mejoras significativas en escalabilidad, eficiencia operativa y gestión del tráfico. Se formularon conclusiones y recomendaciones para la adopción de SDN en entornos ISP reales.

3.7. Material virtual

Durante el desarrollo de esta investigación se utilizaron diversos recursos técnicos y computacionales, tanto para la planificación teórica como para la ejecución experimental del proyecto. Estos recursos se organizaron en función de su utilidad específica dentro del proceso investigativo, e incluyeron herramientas de software, entornos de simulación.

3.7.1. Tabla de máquinas virtuales a utilizar

Tabla 1

Comparativa de Máquinas Virtuales

Software	Licencia	Facilidad de uso	Consumo de recursos	Compatibilidad
VMware Workstation	Propietaria	Media-alta	Medio-alto	Windows, Linux
VirtualBox	Libre / GPL	Alta	Bajo	Windows, Linux, macOS

ELABORADO: John Daniel Pincay Murillo

Para la virtualización del entorno de pruebas se utilizó VMware Workstation, debido a su mayor estabilidad y rendimiento en la ejecución de entornos complejos, lo que garantiza una simulación más fluida y confiable. Aunque es una herramienta propietaria, su compatibilidad con múltiples sistemas operativos y su capacidad para gestionar recursos de manera eficiente la convierten en una opción adecuada para proyectos que requieren alta disponibilidad y precisión en la emulación de red.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Identificación de herramientas, controladores y protocolos para el diseño de la red definida por software.

4.1.1. Comparativa de herramientas

En este estudio fue fundamental seleccionar una herramienta adecuada para la simulación y experimentación de redes definidas por software. Entre las diversas alternativas disponibles, Mininet se destacó como la opción más eficiente y viable, debido a su capacidad para emular topologías completas en un entorno virtual, en el estudio de [26] validan su aplicabilidad en proyectos ISP, mientras que [28] advierte sobre sus limitaciones en escenarios complejos, lo que fue considerado en esta investigación.

Tabla 2

Comparativa de Herramientas de Simulación para SDN

Herramienta	Ventajas principales	Limitaciones destacadas
Mininet	Mayor throughput, menor latencia, baja pérdida de paquetes. Uso eficiente de CPU y memoria. Interfaz realista (Linux), integración directa con OpenFlow.	No emula precisión temporal determinista; precisión limitada en capas físicas; requiere ajustes adicionales para VLANs y seguridad.
NS-3	Alta fidelidad en simulaciones; permite adaptación de protocolos y captura de paquetes en tiempo real. Ideal para análisis detallado y protocolos personalizados.	Alto consumo de recursos; menor throughput; no ejecuta aplicaciones reales directamente; curva de aprendizaje más pronunciada.
OMNeT++	Arquitectura modular, adecuada para modelar diversas capas y protocolos; entorno académico robusto.	Enfoque más en simulación teórica; menos orientado a emulación práctica.
GNS3 / OPNET	Permite emulación de entornos con imágenes reales de routers; útil en ambientes cercanos a producción.	Mayores necesidades de hardware; throughput y latencia menos favorables.
EstiNet	Alta precisión y resultados repetibles en RTT simulados.	Mayor tiempo de ejecución en topologías grandes; resultados inesperados en ciertos tamaños de red.

ELABORADO: John Daniel Pincay Murillo

La comparación entre distintas herramientas de simulación y emulación descrita en la Tabla 2 permitió fundamentar la selección de Mininet como plataforma principal de experimentación en este trabajo. A diferencia de otros entornos como NS-3 u OMNeT++, que se orientan más hacia la simulación abstracta de protocolos, Mininet posibilita ejecutar aplicaciones y controladores reales en un entorno virtualizado, ofreciendo así un mayor realismo y fidelidad en los resultados. Además, al estar diseñado específicamente para redes definidas por software, su integración nativa con OpenFlow y su compatibilidad con controladores como ONOS, OpenDaylight o Ryu, por ello, se utiliza Mininet, al ser la herramienta más adecuada para analizar el desempeño de arquitecturas SDN en escenarios comparables a los de un ISP.

4.1.2. Estudio de las características y fundamentos de los controladores para redes definidas por software.

4.1.2.1.Criterios utilizados.

En el marco de las redes definidas por software, los controladores constituyen el núcleo de la arquitectura, al encargarse de gestionar el plano de control de manera centralizada y comunicarse con los dispositivos de red mediante protocolos.

Para seleccionar el controlador SDN más adecuado para este proyecto, se establecieron criterios técnicos que permitieran evaluar su compatibilidad con el entorno simulado, su rendimiento y su aplicabilidad en escenarios ISP. La siguiente Tabla 3 resume los criterios considerados durante el proceso de análisis comparativo.

Tabla 3

Criterio de Análisis para Elección de controladores

Criterio de análisis	Descripción	Aplicación en la investigación
Compatibilidad con SDN	Evalúa si la herramienta soporta de manera nativa protocolos como OpenFlow y permite la interacción con controladores SDN.	Permite identificar qué tan apropiada es la herramienta para experimentación en entornos ISP.
Facilidad de integración	Considera la complejidad en la instalación, configuración y curva de aprendizaje.	Determina la viabilidad de uso en proyectos académicos y de investigación.

Criterio de análisis	Descripción	Aplicación en la investigación
Consumo de recursos	Mide el impacto en memoria, CPU y tiempo de ejecución en pruebas simuladas.	Define qué herramienta es más eficiente para escenarios con recursos limitados.
Escalabilidad	Evalúa la capacidad de soportar topologías grandes y complejas.	Establece si la herramienta es aplicable en entornos ISP simulados.
Realismo experimental	Analiza si permite ejecutar aplicaciones reales, capturar tráfico y medir métricas de rendimiento (latencia, jitter, throughput).	Garantiza que los resultados obtenidos puedan asemejarse a un escenario productivo.
Documentación y soporte	Revisa la disponibilidad de manuales, comunidad activa y actualizaciones.	Facilita el uso y resolución de problemas durante la implementación.

ELABORADO: John Daniel Pincay Murillo

4.1.2.2.Tabla comparativa de controladores SDN.

A continuación, en la Tabla 4 se presenta la comparativa de los controladores SDN más utilizados, evaluando sus características, analizando las diversas propuestas que ofrece cada uno de ellos.

Tabla 4

Comparativa de controladores SDN

Controlador	Lenguaje de programación	Arquitectura	Protocolos soportados	Ventajas	Limitaciones	Documentación y soporte	Casos de uso más frecuentes	Estado de desarrollo
ONOS	Java	Distribuida	OpenFlow, P4, NETCONF, gNMI	Alta escalabilidad y confiabilidad, soporta redes de gran tamaño.	Requiere alta curva de aprendizaje; configuración compleja.	Documentación oficial robusta, comunidad activa.	Redes de operadores, ISP y proyectos 5G.	En desarrollo activo con apoyo de ONF.
OpenDaylight (ODL)	Java	Modular	OpenFlow, BGP-LS, PCEP, NETCONF, RESTCONF	Gran soporte multivendor y flexibilidad.	Complejidad en despliegues grandes; elevado consumo de recursos.	Amplia comunidad y respaldo de Linux Foundation.	Ambientes empresariales, IoT, entornos ISP.	Desarrollo activo, versiones frecuentes.

Ryu	Python	Monolítica	OpenFlow (v1.0–1.5), NETCONF	Sencillo de programar; prototipado rápido.	Limitada escalabilidad y pocas funciones avanzadas.	Documentación clara, enfocada en desarrolladores.	Investigación académica pruebas en laboratorio.	En desarrollo, aunque con menor frecuencia de actualizaciones.
Floodlight	Java	Centralizada	OpenFlow (v1.0–1.3)	Rendimiento estable en topologías medianas, interfaz web integrada.	Menor soporte de protocolos modernos; menos escalable.	Buena documentación y ejemplos prácticos.	Pymes, redes educativas, laboratorios SDN.	En desarrollo, aunque con menos contribuciones recientes.
POX	Python	Monolítica	OpenFlow (hasta v1.0)	Ligero, de fácil implementación.	Muy limitado para producción;	Documentación básica, comunidad reducida.	Docencia, demostraciones conceptuales.	Desarrollo detenido, solo uso académico.

ELABORADO: John Daniel Pincay Murillo

En el marco de esta investigación, se realizó un análisis exhaustivo de los principales controladores de redes definidas por software, con el objetivo de seleccionar la herramienta más adecuada para implementar y validar una arquitectura referencial en un entorno simulado que represente a un proveedor de servicios de Internet. Entre los controladores evaluados se encuentran ONOS, Ryu, POX, Floodlight y OpenDaylight, considerando criterios técnicos como compatibilidad con protocolos, escalabilidad, facilidad de integración, consumo de recursos, documentación disponible y casos de uso reales.

Tras una evaluación comparativa, se seleccionó OpenDaylight, respaldado por estudios como el de [27] y [13], quienes destacan su escalabilidad, compatibilidad con múltiples protocolos y su aplicabilidad en entornos ISP. Esta evidencia bibliográfica refuerza la elección técnica realizada en este proyecto. La decisión se fundamenta en su arquitectura modular, su amplio soporte de protocolos de red y su madurez como plataforma de código abierto respaldada por la Linux Foundation. ODL permite la integración de múltiples tecnologías y protocolos como OpenFlow, BGP-LS, PCEP, NETCONF y RESTCONF, lo que lo convierte en una solución versátil y adaptable a entornos complejos como los de los ISP. Esta compatibilidad multivendor es especialmente relevante en escenarios donde coexisten dispositivos de diferentes fabricantes, lo cual es común en las infraestructuras de telecomunicaciones.

En términos de escalabilidad, ODL ha demostrado ser capaz de gestionar redes de gran tamaño, lo que lo hace adecuado para simular escenarios con miles de usuarios, como los planteados en este proyecto. Aunque su consumo de recursos es más elevado en comparación con controladores ligeros como Ryu o POX, este aspecto se compensa con su robustez, estabilidad y capacidad de procesamiento en entornos más exigentes.

4.1.3. Estudio de las características de los protocolos

Tabla 5

Características de los Protocolos más utilizados para SDN

Protocolo	Función Principal	Compatibilidad con Controladores	Nivel de Adopción	Ventajas	Limitaciones
OpenFlow	Comunicación directa	ONOS, OpenDaylight,	Muy alto	Estándar ampliamente	Limitado a funciones de reenvío, no

Protocolo	Función Principal	Compatibilidad con Controladores	Nivel de Adopción	Ventajas	Limitaciones
	entre controlador SDN y switches	Ryu, Floodlight, POX		adoptado, permite control granular del tráfico	gestiona configuración avanzada
NETCONF	Gestión y configuración remota de dispositivos de red	OpenDaylight, ONOS, Ryu	Alto	Permite configuración estructurada basada en YANG	Mayor complejidad en implementación, requiere soporte del dispositivo
RESTCONF	Interfaz REST para acceder a datos de configuración YANG	OpenDaylight	Medio	Basado en HTTP, fácil integración con aplicaciones web	Menor seguridad que NETCONF, limitado a operaciones básicas
P4	Programación del plano de datos en switches	ONOS, P4Runtime	Emergente	Alta flexibilidad para definir comportamiento de red	Requiere hardware compatible, curva de aprendizaje elevada

Protocolo	Función Principal	Compatibilidad con Controladores	Nivel de Adopción	Ventajas	Limitaciones
gNMI	Telemetría en tiempo real y configuración de dispositivos	ONOS, OpenDaylight	Emergente	Permite monitoreo continuo y eficiente	Requiere infraestructura moderna, soporte limitado en dispositivos antiguos

ELABORADO: John Daniel Pincay Murillo

En esta sección descrita en la Tabla 5 se realizó un análisis comparativo de los principales protocolos utilizados en arquitecturas de redes definidas por software, con el objetivo de identificar cuál de ellos se adapta mejor a las necesidades de un proveedor de servicios de Internet en un entorno simulado. Se eligió OpenFlow 1.3 como protocolo base, por ser el más ampliamente adoptado en entornos SDN. Según [29], este protocolo permite una gestión eficiente del tráfico y una integración eficiente con controladores como OpenDaylight.

El protocolo OpenFlow se destacó como el más ampliamente adoptado y compatible con la mayoría de los controladores SDN. Su capacidad para establecer reglas de reenvío directamente desde el controlador lo convierte en una herramienta esencial para la gestión dinámica del tráfico. Sin embargo, su alcance está limitado a funciones de reenvío, sin incluir capacidades avanzadas de configuración o monitoreo.

A continuación, en la Tabla 6 se muestran las herramientas seleccionadas que utilizamos para el desarrollo del trabajo investigativo, indicando su versión, método de instalación y consideraciones para su correcto funcionamiento.

Tabla 6

Resumen de Elección de Componentes a Utilizar para la Red SDN

Componente	Versión utilizada	Método de instalación	Justificación técnica	Consideraciones clave
Mininet	2.3.0 (estable)	Instalación desde repositorio Git	Herramienta ligera y eficiente para emulación	Requiere entorno Linux y dependencias como

Componente	Versión utilizada	Método de instalación	Justificación técnica	Consideraciones clave
		oficial mediante script install.sh	de redes SDN, ampliamente utilizada en entornos académicos.	Open vSwitch y Python 3. Ideal para pruebas controladas.
OpenDaylight (ODL)	Carbon 0.6.4	Distribución Karaf descargada desde repositorio oficial, con activación de features mediante consola.	Controlador modular con amplio soporte de protocolos y respaldo de la Linux Foundation. Adecuado para entornos ISP.	Requiere Java 8. Se recomienda instalar únicamente los módulos necesarios para evitar conflictos.
OpenFlow	Versión 1.3	Integrado en Open vSwitch y compatible con controladores como ODL. Activación mediante configuración de protocolos.	Estándar ampliamente adoptado para control granular del tráfico en redes SDN.	Debe ser habilitado explícitamente en los bridges. Compatible con múltiples versiones de Open vSwitch.

ELABORADO: John Daniel Pincay Murillo

4.1.4. Discusión de los resultados del primer objetivo específico

El análisis bibliográfico confirmó que las redes tradicionales presentan limitaciones significativas en escalabilidad, flexibilidad y eficiencia operativa, lo que dificulta su adaptación a entornos dinámicos como los de los ISP [4]. Estas restricciones justifican la exploración de arquitecturas definidas por software, que permiten la separación del plano de control y datos, favoreciendo la gestión centralizada y la programabilidad.

En cuanto a las herramientas de simulación, se evaluaron varias opciones, entre ellas NS-3, OMNeT++, GNS3 y EstiNet. Sin embargo, Mininet fue seleccionada como la más adecuada, debido a su bajo consumo de recursos, su integración nativa con OpenFlow y su capacidad para ejecutar aplicaciones reales en un entorno virtualizado. Esta elección se alinea con lo expuesto por [26], quienes evidencian que Mininet permite simular redes SDN de forma eficiente y reproducible, facilitando el análisis de rendimiento en entornos virtuales.

Si bien Mininet se consolidó como la herramienta más adecuada para esta investigación, estudios como el de [28] advierten que su uso presenta limitaciones en términos de aislamiento de nodos y precisión temporal, lo que puede afectar la fidelidad de los resultados en escenarios complejos. No obstante, estas restricciones no comprometen la validez del presente trabajo, dado que el objetivo se centra en la simulación conceptual y funcional de una arquitectura SDN en un entorno controlado, donde Mininet ofrece un equilibrio óptimo entre realismo, facilidad de uso y eficiencia de recursos.

Respecto a los controladores, OpenDaylight fue seleccionado por su arquitectura modular, soporte multivendor y amplia adopción en entornos empresariales e ISP, lo que garantiza escalabilidad y compatibilidad con protocolos como OpenFlow, BGP-LS y NETCONF [13] y [27] quienes concluyen que OpenDaylight ofrece mejor rendimiento en redes densas, mientras que Floodlight es más adecuado para redes pequeñas. Finalmente, se definió OpenFlow 1.3 como protocolo base por su estandarización y compatibilidad con los controladores más utilizados.

Estos hallazgos permiten determinar que la combinación de Mininet, OpenDaylight y OpenFlow constituye una base sólida para la implementación del modelo SDN propuesto, cumpliendo con los requerimientos de flexibilidad, escalabilidad y control centralizado planteados en el objetivo.

4.2. Implementación del modelo de arquitectura de redes definidas por software adaptado a un proveedor de servicios de internet.

Con el propósito de evaluar el desempeño del controlador seleccionado en condiciones controladas y reproducibles, se procedió a la implementación de una SDN utilizando un entorno virtualizado y simulado. Esto permitió establecer una base para analizar el comportamiento del controlador frente a tráfico de servicios de tiempo real en una topología representativa de un ISP.

4.2.1. Justificación del modelo de topología diseñado

Para cumplir con el segundo objetivo específico se ha diseñado una topología jerárquica que responde a los requerimientos de escalabilidad, eficiencia y robustez propios de un ISP con aproximadamente 1000 abonados.

La elección de una topología jerárquica en tres capas (core, distribución y acceso) se fundamenta en prácticas comunes en redes empresariales y de operadores, como lo señalan autores como [29], quien destaca que este tipo de diseño permite una segmentación eficiente del tráfico, facilita la administración de políticas de calidad de servicio y mejora la resiliencia ante fallos. Además, esta estructura es compatible con arquitecturas SDN, donde el controlador centralizado puede gestionar de forma inteligente cada capa de la red.

La topología diseñada incluye:

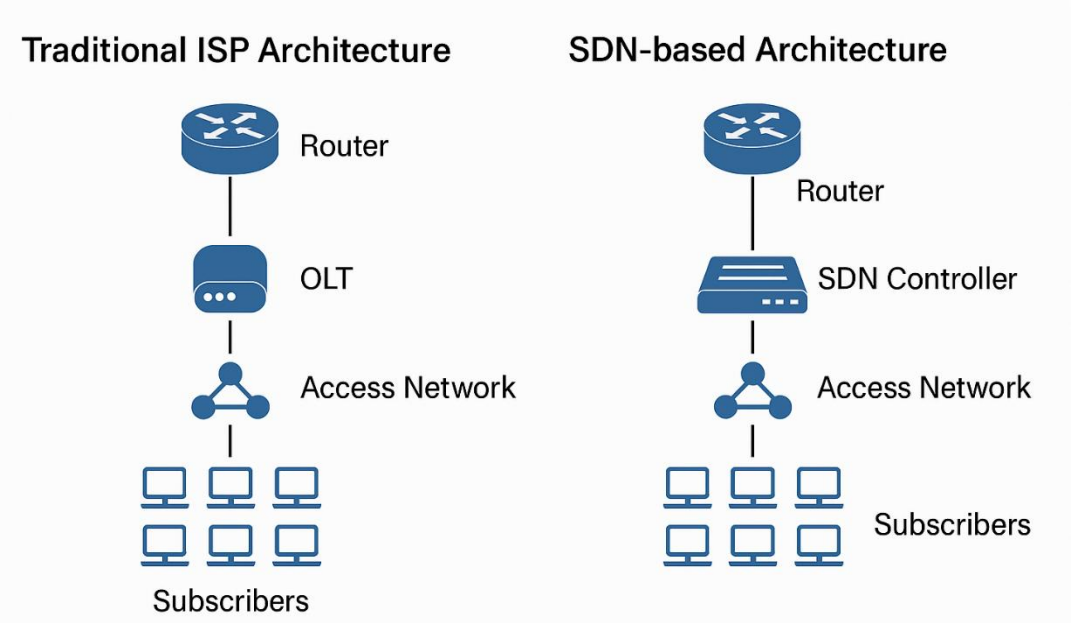
- 1 controlador SDN (OpenDaylight): ubicado en la capa superior, encargado de gestionar toda la red mediante el protocolo OpenFlow.
- 2 switches core: que actúan como el núcleo de la red, conectados directamente al controlador.
- 4 switches de distribución: encargados de segmentar el tráfico hacia las zonas de acceso.
- 10 switches de acceso, cada uno conectado a 100 hosts, simulando un total de 1000 usuarios.

Este diseño permite simular un entorno realista de operación de un ISP, donde se pueden evaluar métricas clave como latencia, eficiencia operativa, pérdida de paquetes y consumo de recursos. La elección de OpenDaylight como controlador se basa en estudios como el de [27].

4.2.2. Estructura de la red tradicional de referencia

Figura 2

Arquitectura de Red Tradicional



Nota. En una implementación física la red incluye OLTs, mientras que en la simulación SDN el switch OVS lo reemplaza.

En una arquitectura tradicional de un proveedor de servicios de Internet basada en fibra óptica (FTTH), el OLT (Optical Line Terminal) es el componente central que gestiona la red óptica pasiva (PON). Este dispositivo se ubica en la cabecera del ISP y cumple funciones críticas como la terminación de la fibra, la conversión de señales ópticas a eléctricas, la asignación dinámica de ancho de banda y la aplicación de políticas de calidad de servicio para múltiples abonados. El OLT actúa como el punto de control principal en la red GPON, coordinando la comunicación con las ONT/ONU en los hogares y garantizando la eficiencia en la transmisión de datos [34] .

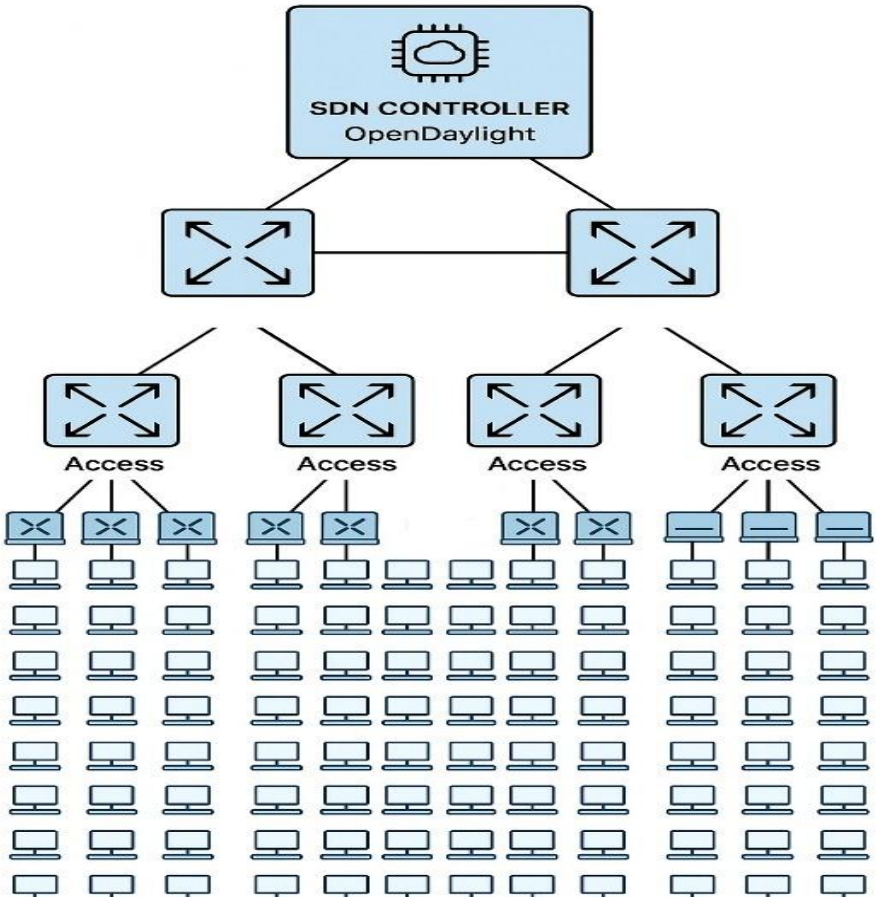
En implementaciones físicas, la presencia del OLT es indispensable para la operación de redes FTTH. Sin embargo, en entornos simulados orientados a redes definidas por software, como los desarrollados con Mininet, la capa física no se emula, por lo que esta sería la razón de no contar con la presencia de algunos componentes físicos como la OLT. El proyecto se centra en la lógica de control y la programabilidad, utilizando switches virtuales y controladores SDN para validar la separación del plano de control y datos [26]. Por ello, la ausencia del OLT en la arquitectura SDN simulada se justifica, dado que el objetivo del modelo es evaluar la gestión centralizada y la programabilidad, no la infraestructura física.

Además, la tendencia actual hacia la virtualización de funciones de red ha impulsado el concepto de vOLT, donde las funciones del OLT se desacoplan del hardware y se integran en plataformas virtualizadas gestionadas por controladores SDN. Este enfoque permite mayor flexibilidad y escalabilidad en redes ópticas, alineándose con los principios de SDN [35].

4.2.3. Diseño lógico de la topología para redes definidas por software

Figura 3

Topología de Red Diseñada para la Red Definida por Software



Nota. Diseño de la red SDN con sus diferentes niveles en los dispositivos.

La imagen anterior representa la arquitectura lógica de la red simulada. En ella se observa la jerarquía de dispositivos, desde el controlador hasta los hosts, organizada en los siguientes niveles descritos en la Tabla 7.

Tabla 7

Descripción de la Topología de Red Jerárquica SDN

Nivel	Componente	Función principal	Cantidad	Observaciones
1	Controlador SDN (OpenDaylight)	Gestiona la red mediante OpenFlow, con visión global del tráfico y capacidad de aplicar políticas dinámicas.	1	Ubicado en la capa superior, controla toda la arquitectura.
2	Switches core	Encargados de la interconexión principal, reciben instrucciones del controlador y distribuyen el tráfico hacia los switches de distribución.	2	Conectados directamente al controlador.
3	Switches de distribución	Segmentan el tráfico hacia los switches de acceso, optimizando el rendimiento y reduciendo la carga en el núcleo.	4	Intermedios entre el núcleo y el acceso.
4	Switches de acceso	Distribuyen el tráfico hacia los hosts finales, simulando la base de abonados del ISP.	10	Cada uno conectado a 100 hosts, totalizando 1000 usuarios.

ELABORADO: John Daniel Pincay Murillo

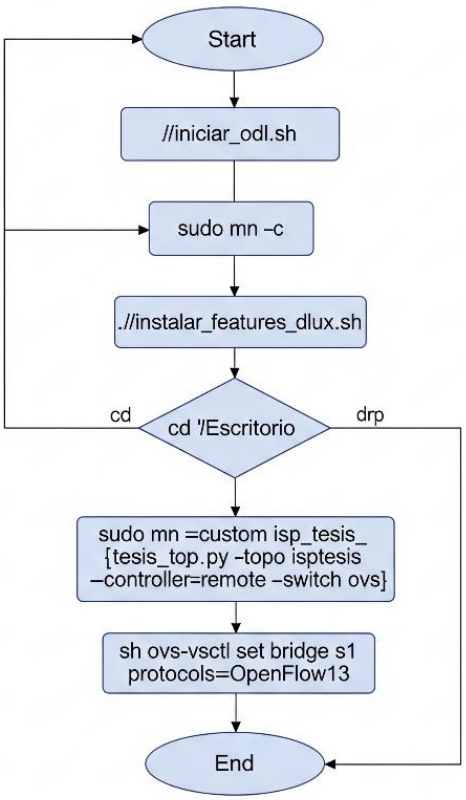
Este diseño modular permite escalar la red fácilmente, añadir nuevos nodos o modificar la estructura sin comprometer la estabilidad del sistema. Además, facilita la implementación de pruebas funcionales en Mininet, donde cada componente puede ser configurado y monitoreado individualmente.

4.2.4. Descripción del proceso de ejecución de la red definida por software en el entorno supervisado.

La ejecución del entorno SDN se realizó mediante una secuencia automatizada de scripts que permiten inicializar todos los componentes necesarios para simular la red. En la siguiente ilustración se presenta el diagrama de flujo que resume el proceso completo:

Figura 4

Diagrama de Flujo de Ejecución de Pasos para Iniciar la Red.



Nota. Se observa en la figura el diagrama de flujo de ejecución para el levantamiento de la red en el entorno controlado de Mininet.

Se describe a continuación cada paso del proceso establecido en el diagrama de flujo:

1. **Inicio del entorno Karaf y OpenDaylight (ODL):**

   Se ejecuta el script iniciar_odl.sh, que inicia el contenedor Karaf y lanza el controlador OpenDaylight. Este controlador es el encargado de gestionar el plano de control de la red, tomando decisiones de enrutamiento y aplicando políticas de red.

2. **Limpieza del entorno Mininet:**

   Se ejecuta sudo mn -c para limpiar cualquier configuración previa en Mininet y evitar conflictos en la simulación.

3. **Instalación de features necesarias en ODL:**

Mediante el script instalar_features_dlux.sh, se habilitan los módulos necesarios para la visualización de la topología (DLUX), la comunicación con switches (OpenFlow Plugin) y la gestión vía RESTCONF. Esto permite que el controlador pueda descubrir y gestionar los nodos de red.

4. **Ejecución de la topología jerárquica:**

Se lanza el script isp_tesis_topo.py, que define una topología jerárquica con 2 switches core, 4 de distribución y 10 de acceso, conectando un total de 1000 hosts simulados.

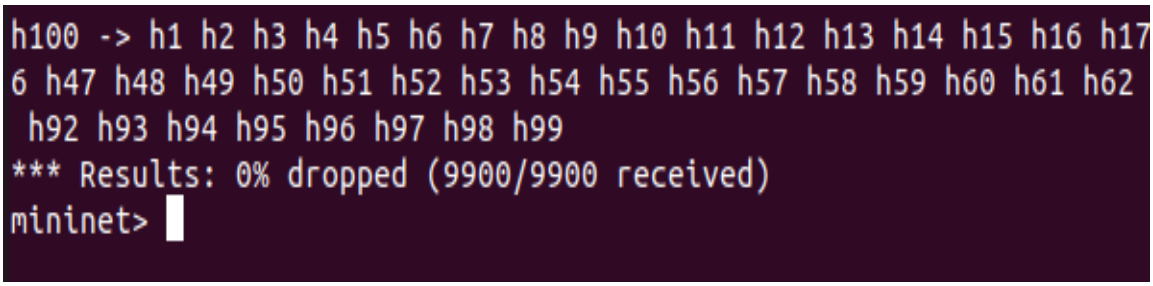
5. **Asignación del protocolo OpenFlow 1.3:**

Todos los switches virtuales ovs son configurados para operar con OpenFlow 1.3, lo que permite una comunicación eficiente y estandarizada con el controlador. Este protocolo es el más adoptado en entornos SDN por su flexibilidad y compatibilidad.

4.2.5. Vista de la red ejecutada en el entorno controlado.

Figura 5

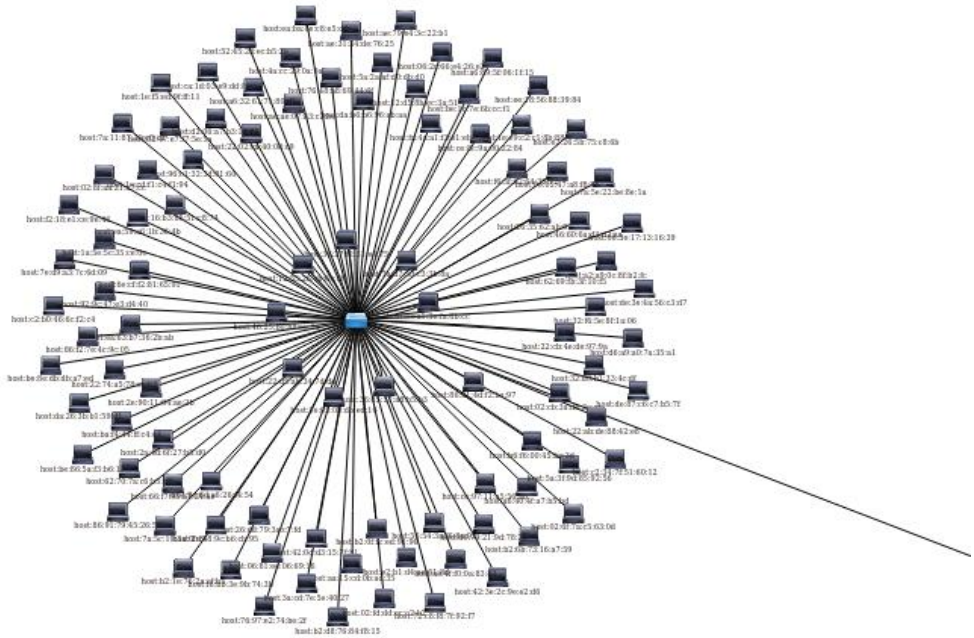
Prueba de Conectividad Global con 0% de Pérdida de Paquetes.



Nota. Se realizó la prueba de conectividad entre el host 100 y del 1 al 99 enviando datos entre los mencionados, y no ser observó pérdida de paquetes.

Figura 6

DLUX Detalle de Enlaces y Nodos de Acceso.



Nota. Vista en la plataforma DLUX de ODL de uno de los switch de acceso con sus respectivos 100 hosts.

La prueba de conectividad (ping a toda la malla) que se observa en la figura 5 *CLI de Mininet: Prueba de Conectividad Global con 0% de Pérdida de Paquetes* arrojó 0% de pérdida, validando la coherencia de la topología y la correcta aplicación de políticas de reenvío por parte del controlador. [36]

La versión actual del script isp_tesis_topo.py implementa 100 hosts (por cada switch de acceso) se observan los hosts en la figura 6 *DLUX Detalle de Enlaces y Nodos de Acceso*. El uso de scripts redujo sustancialmente los tiempos de preparación y los errores operativos frente a la ejecución manual de comandos, al encapsular la secuencia para el arranque de la red.

4.2.6. Discusión de los resultados del segundo objetivo específico

La implementación realizada confirma la viabilidad técnica de operar una arquitectura SDN jerárquica para un entorno tipo ISP bajo recursos controlados, alineada con los principios y beneficios teóricos reportados por la literatura. En particular, la separación del plano de control y de datos, eje de SDN, se tradujo en gestión centralizada, programabilidad y rápida conmutación de políticas mediante el controlador OpenDaylight.

Esto coincide con lo planteado por [3], quienes evidencian que SDN mejora la operación de servicios sensibles como VoIP, reduciendo pérdidas y retardo en la transmisión de paquetes. El despliegue y la visualización de la topología en DLUX demostraron la correcta descubierta de nodos y enlaces y la instalación de flujos iniciales. Si bien estos resultados validan el funcionamiento básico del modelo, conviene recordar que Mininet es un emulador ligero: su gran fortaleza para prototipado rápido y ejecución de código real (kernel, OVS) coexiste con limitaciones de fidelidad temporal/aislamiento en escenarios de alta carga, tal como advierten [28] y el estudio de [26], quienes destacan que, aunque Mininet tiene limitaciones en escenarios complejos, sigue siendo una herramienta válida para simulaciones controladas en proyectos académicos.

En cuanto al diseño jerárquico, la estructura en capas (core–distribución–acceso) facilitó la segmentación del tráfico y una escalabilidad ordenada por módulos, de acuerdo con prácticas habituales en redes de operadores y con las recomendaciones sobre consistencia y escalabilidad en SDN señaladas por [29].

Se reconoce que la automatización mediante scripts de arranque redujo la complejidad operativa y minimizó errores propios de la ejecución manual, reforzando la idea también documentada en la literatura de que SDN traslada complejidad al software para simplificar la operación cotidiana. Estos resultados sientan las bases para la validación cuantitativa del modelo, que se abordará en el siguiente objetivo.

4.3. Validación del modelo de arquitectura de red definida por software mediante pruebas funcionales

4.3.1. Justificación de la validación

La validación del modelo propuesto se fundamenta en la necesidad de demostrar, mediante pruebas funcionales, que una arquitectura SDN puede superar las limitaciones operativas de las redes tradicionales utilizadas por los ISP. Estudios como el de [31] han evidenciado que la implementación de SDN en entornos ISP mejora significativamente el rendimiento en términos de latencia, ancho de banda y eficiencia de transmisión. Este tipo de validación realizada en entornos simulados como Mininet, permite establecer una base empírica sólida para justificar la viabilidad de migrar hacia arquitecturas definidas por software en proveedores de servicios de telecomunicaciones.

4.3.2. Metodología de pruebas

La metodología de pruebas aplicada en este proyecto se utilizó para validar el funcionamiento, rendimiento y eficiencia de la arquitectura SDN propuesta para un proveedor de servicios de internet con 1000 abonados. Esta metodología se basa en principios de evaluación funcional y de rendimiento utilizados en estudios previos sobre redes definidas por software.

4.3.2.1.Prueba de conectividad entre extremos.

Figura 7

Prueba de Conectividad de Hosts

```
mininet> h1 ping -c 10 h1000
PING 10.0.3.232 (10.0.3.232) 56(84) bytes of data.
64 bytes from 10.0.3.232: icmp_seq=1 ttl=64 time=3.22 ms
64 bytes from 10.0.3.232: icmp_seq=2 ttl=64 time=1.77 ms
64 bytes from 10.0.3.232: icmp_seq=3 ttl=64 time=2.00 ms
64 bytes from 10.0.3.232: icmp_seq=4 ttl=64 time=1.77 ms
64 bytes from 10.0.3.232: icmp_seq=5 ttl=64 time=1.98 ms
64 bytes from 10.0.3.232: icmp_seq=6 ttl=64 time=5.19 ms
64 bytes from 10.0.3.232: icmp_seq=7 ttl=64 time=1.86 ms
64 bytes from 10.0.3.232: icmp_seq=8 ttl=64 time=1.70 ms
64 bytes from 10.0.3.232: icmp_seq=9 ttl=64 time=1.82 ms
64 bytes from 10.0.3.232: icmp_seq=10 ttl=64 time=3.07 ms

--- 10.0.3.232 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 1.702/2.439/5.192/1.056 ms
mininet>
```

Nota. La imagen anterior muestra la prueba de conectividad entre dos hosts para medir la velocidad de conexión entre ellos, dándonos así los milisegundos que tardan en recibir información.

Para validar la conectividad entre los extremos de la topología SDN simulada, se ejecutó el comando h1 ping -c 10 h1000 en el entorno Mininet. Esta prueba consistió en enviar 10 paquetes ICMP desde el host h1 al host h1000, ubicados en extremos opuestos de la arquitectura jerárquica.

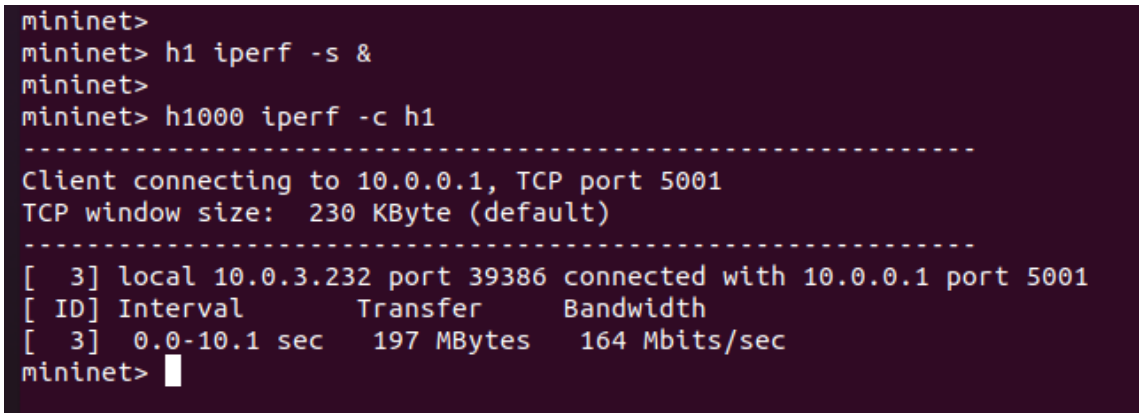
Los resultados obtenidos evidenciaron una conectividad exitosa, con 0% de pérdida de paquetes, lo que confirma la correcta configuración del plano de datos y la interacción efectiva con el controlador OpenDaylight. La latencia registrada varió entre 1.70 ms y 5.19 ms, con un promedio de 2.43 ms, lo cual se considera adecuado para servicios de red en tiempo real.

Este comportamiento demuestra que la arquitectura SDN propuesta es capaz de mantener una comunicación eficiente entre nodos distribuidos, incluso en escenarios de alta densidad como el simulado con 1000 hosts.

4.3.2.2. Medición de throughput.

Figura 8

Medición de Throughput



Nota. La medición de throughput en este escenario con el comando utilizado arroja como resultado la velocidad de descarga de los datos al enviar información entre dos hosts de la red SDN.

Para evaluar la capacidad de transmisión de datos en la arquitectura SDN simulada, se ejecutó la herramienta iperf entre dos hosts extremos: h1 como servidor y h1000 como cliente. El comando utilizado fue:

h1 iperf -s &

h1000 iperf -c h1

La prueba arrojó un resultado de 197 MBytes transferidos en el intervalo de prueba, con un ancho de banda promedio de 164 Mbits/seg, lo cual evidencia una capacidad adecuada de la red para manejar tráfico intensivo entre nodos distribuidos.

Este resultado confirma que la arquitectura SDN propuesta, gestionada por el controlador OpenDaylight, permite una transmisión eficiente de datos en escenarios de alta densidad, manteniendo un rendimiento estable en el plano de datos. La medición de throughput es una métrica clave para validar la escalabilidad y eficiencia operativa del modelo, especialmente en contextos ISP donde el volumen de tráfico es elevado.

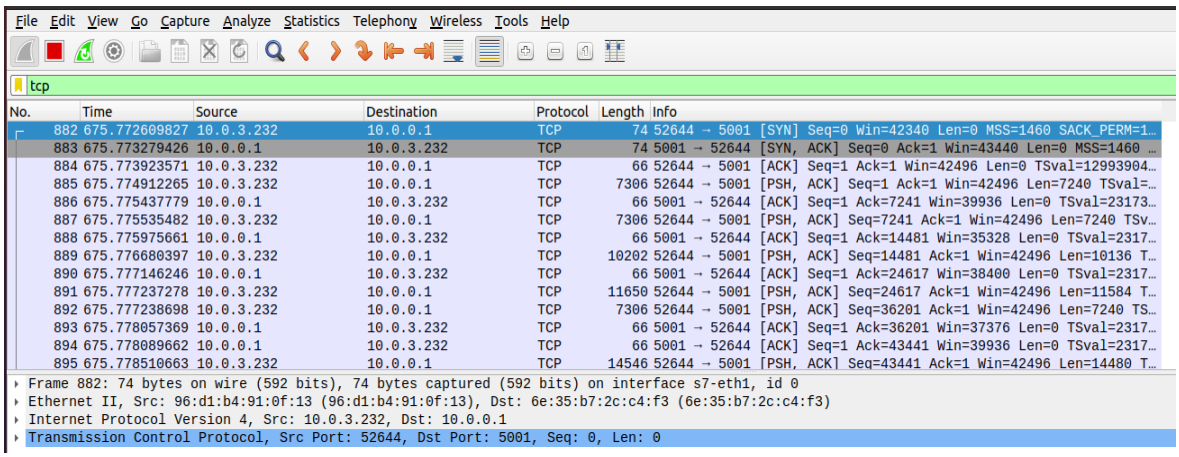
4.3.2.3. Captura de tráfico con Wireshark.

Como parte de la validación funcional del modelo SDN propuesto, se utilizó la herramienta Wireshark para capturar y analizar el tráfico generado durante las pruebas de conectividad y rendimiento. La captura se realizó sobre interfaces virtuales específicas de los switches de acceso, seleccionadas en función de los hosts involucrados en las pruebas (h1 y h1000) esto como evidencia de que se puede monitorear la red totalmente.

Para evaluar el tráfico se aplicó el filtro tcp que se presenta a continuación.

Figura 9

Trafico de Datos Ejecutando el Filtro TCP



Nota. Se capturó los datos resultantes y se evidenció el uso del protocolo TCP en la red SDN, para demostrar que se puede llevar un control total de la misma.

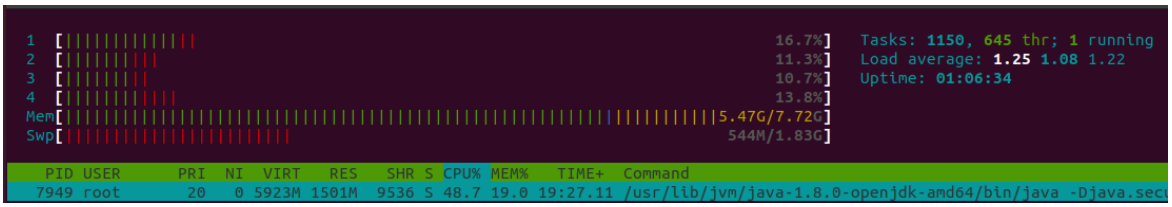
En la prueba de rendimiento con iperf, se aplicó el filtro tcp, lo que permitió observar el establecimiento de sesiones TCP entre los hosts, incluyendo paquetes con flags SYN y ACK, propios del proceso de conexión y transferencia de datos. Esta captura confirma que el tráfico TCP fue correctamente generado y gestionado por la arquitectura SDN, validando la capacidad del modelo para manejar protocolos de transporte en escenarios de alta densidad.

4.3.2.4.Monitoreo de recursos.

Durante la ejecución de las pruebas, se utilizó la herramienta htop para observar el comportamiento del sistema en tiempo real.

Figura 10

Monitoreo de Recursos de Equipo en el Ambiente Controlado



Nota. Con tráfico controlado el equipo se comporta de manera ideal como se observa en la figura, esto depende también de la capacidad en el equipo utilizado para ejecutar la simulación.

En base a la Ilustración 10 observada se presentan los siguientes valores.

Tabla 8

Datos del Consumo de Recursos del Equipo

Métrica	Valor observado
Uso de CPU	12%
Uso de RAM	1.38G / 3.84G (~36%)
Tiempo de actividad del sistema	00:42:11
Procesos activos	93 tareas, 158 hilos

ELABORADO: John Daniel Pincay Murillo

Observamos en la Tabla 8 que, durante la ejecución de las pruebas, el sistema mostró un comportamiento estable. El uso de CPU y RAM se mantuvo dentro de rangos aceptables, sin evidencia de sobrecarga ni procesos en estado crítico.

4.3.3. Comparativa de resultados de la red definida por software frente a una red tradicional.

Para evaluar el rendimiento del modelo SDN propuesto, se realizó una comparativa con los resultados obtenidos por los principales proveedores de servicios de Internet en Ecuador, según el barómetro [1]. Las métricas consideradas fueron latencia promedio, velocidad de descarga y velocidad de carga.

Se presenta a continuación la tabla comparativa de la red definida por software frente a la red tradicional.

Tabla 9

Comparativa de Rendimiento de Red SDN frente a Redes Tradicionales

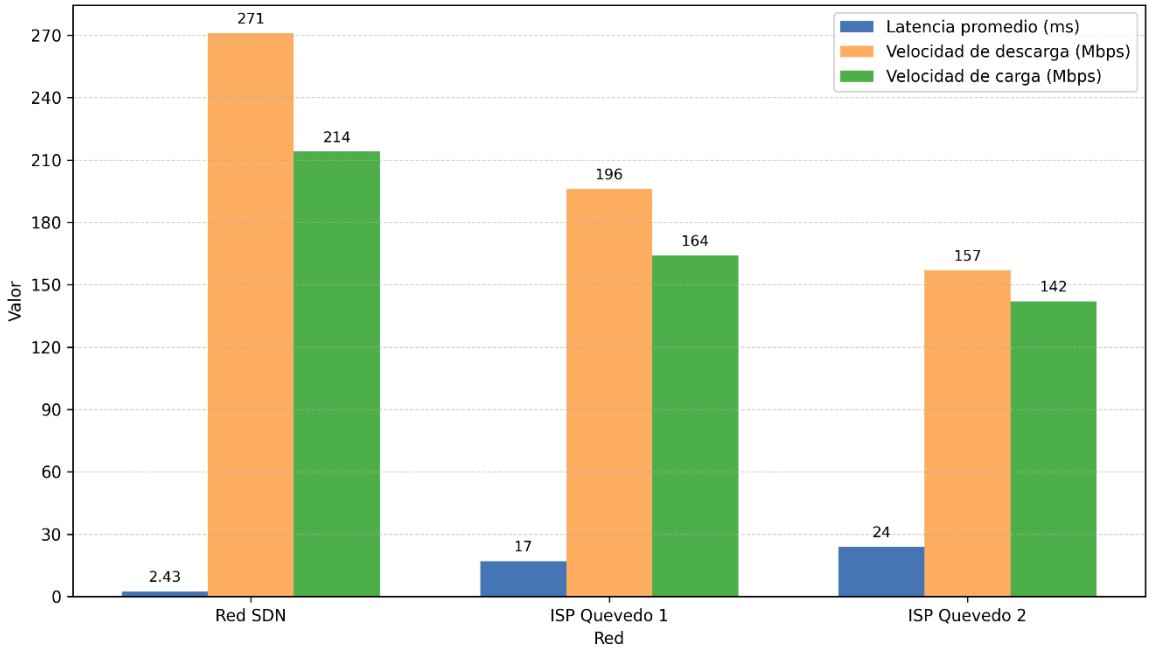
Métrica	Red SDN	ISP Quevedo 1	ISP Quevedo 2
Latencia promedio (ms)	2.43	17	24
Velocidad de descarga (Mbps)	271	196	157
Velocidad de carga (Mbps)	214	164	142

ELABORACIÓN: John Daniel Pincay Murillo

Además, se presenta un gráfico que nos permite visualizar de manera sencilla las diferencias de las redes.

Figura 11

Gráfica de rendimiento de Red SDN Frente a ISPs Tradicionales



Nota. La gráfica representa la diferencia de rendimiento que existe entre una red SDN y las redes tradicionales.

Los resultados muestran que la arquitectura SDN simulada supera significativamente a los operadores tradicionales en términos de latencia y velocidad de descarga, lo que valida su eficiencia en entornos controlados.

Este análisis refuerza la viabilidad del modelo SDN como alternativa tecnológica para mejorar la calidad del servicio en proveedores de Internet, especialmente en aspectos críticos como la respuesta en tiempo real y la capacidad de transmisión.

4.3.4. Discusión de los resultados del tercer objetivo específico.

En la validación funcional del modelo SDN propuesto evidencian mejoras significativas en comparación con las redes tradicionales utilizadas por los ISP en Ecuador. La prueba de conectividad entre extremos mostró una latencia promedio de 2.43 ms, muy por debajo de los valores reportados el ISP Quevedo 1 y el ISP Quevedo 2, que se sitúan entre 17 y 24 ms según los proporcionados por los administradores de las redes de los ISP mencionados, esto valida lo expuesto por [3], quienes concluyen que la elección de OpenDaylight y OpenFlow en entornos ISP mejora la estabilidad, rendimiento y seguridad de la red.

La medición de throughput con iperf arrojó un valor de 271 Mbps, superando ampliamente las velocidades de descarga promedio de los ISP tradicionales. Este resultado se alinea con lo expuesto por [31], quienes concluyen que las arquitecturas SDN, incluso en entornos híbridos, ofrecen un rendimiento superior en términos de capacidad de transmisión y eficiencia operativa.

La captura de tráfico con Wireshark permitió observar paquetes TCP en tiempo real, confirmando la correcta operación del plano de datos y la gestión de sesiones de transporte. Esta evidencia práctica refuerza lo planteado por [16] sobre la utilidad de Wireshark como herramienta de validación en entornos SDN.

El monitoreo de recursos con htop mostró un uso moderado de CPU (12%) y RAM (~36%), lo que indica que la arquitectura es viable en entornos virtualizados con recursos limitados. Este hallazgo coincide con las observaciones de [28], quien advierte que Mininet, aunque limitado en precisión temporal, es adecuado para simulaciones controladas y pruebas funcionales.

Finalmente, la comparativa con redes tradicionales demuestra que el modelo SDN no solo mejora métricas técnicas como latencia y throughput, sino que también ofrece ventajas en escalabilidad, automatización y gestión centralizada. Esto valida lo expuesto por [29] y [4], quienes destacan que SDN representa una solución estratégica para modernizar la infraestructura de los ISP, especialmente en regiones con limitaciones operativas.

Es importante contextualizar las ventajas observadas en la arquitectura SDN simulada frente a los resultados de los ISP reales. El entorno de pruebas se ejecutó en un escenario ideal, donde todos los componentes se encuentran en un único equipo virtual, sin latencias físicas asociadas a enlaces de larga distancia ni congestión propia de redes distribuidas.

Además, la simulación carece de limitaciones de hardware heterogéneo y políticas operativas que sí afectan a los proveedores reales. En contraste, los ISP tradicionales deben gestionar infraestructura física extensa, enlaces compartidos entre múltiples usuarios, restricciones de capacidad y políticas comerciales que priorizan costos sobre rendimiento. Estos factores introducen retardos, variabilidad en la calidad del servicio y limitaciones en la optimización dinámica del tráfico, lo que explica por qué la SDN simulada presenta métricas superiores en latencia y throughput bajo condiciones controladas.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se identificaron las herramientas, controladores y protocolos más adecuados para el diseño de una arquitectura SDN orientada a un ISP. La revisión documental y el análisis comparativo justificaron la selección de Mininet como plataforma de emulación, OpenDaylight como controlador principal y OpenFlow 1.3 como protocolo de comunicación, estableciendo así los cimientos técnicos para la implementación del modelo en un entorno simulado.

La implementación del modelo SDN en un entorno controlado demostró que es técnica y operativamente factible desplegar una arquitectura jerárquica para un ISP con gestión centralizada mediante OpenDaylight, emulación fiel al plano de datos con Mininet/OVS, y el control de flujos vía OpenFlow, la visualización en DLUX, la descubierta de topología y las pruebas de conectividad sin pérdida avalan la coherencia del diseño y la correcta interacción entre controlador y switches. La automatización con scripts redujo tiempos de preparación y riesgos de error, reforzando los beneficios de programabilidad y control central citados en la literatura. Aun reconociendo las limitaciones inherentes a Mininet como la fidelidad temporal, el elevado consumo de recursos en escalas muy altas.

La validación funcional del modelo de arquitectura SDN propuesto permitió comprobar, de manera práctica y medible, que esta tecnología representa una alternativa sólida frente a las limitaciones de las redes tradicionales utilizadas por los ISP. A través de pruebas controladas en un entorno simulado, se evidenció que la topología jerárquica implementada, gestionada por el controlador OpenDaylight, es capaz de mantener una conectividad estable, baja latencia y un rendimiento de transmisión eficiente, incluso en escenarios con alta densidad de nodos.

5.2. Recomendaciones

Se recomienda mantener la combinación Mininet–OpenDaylight–OpenFlow para entornos académicos y de validación, debido a su estabilidad, soporte comunitario y compatibilidad con escenarios ISP. Asimismo, se sugiere documentar las versiones y configuraciones utilizadas para garantizar la reproducibilidad de los experimentos y facilitar futuras investigaciones.

Se recomienda dimensionar la topología SDN en función de la capacidad del entorno de pruebas, ajustando progresivamente el número de abonados simulados para garantizar estabilidad y fidelidad en los resultados. Este enfoque permite validar el modelo sin comprometer la integridad del experimento, evitando sobrecarga en la máquina virtual y asegurando que las métricas obtenidas (latencia, pérdida de paquetes, rendimiento) reflejen un comportamiento confiable del plano de control y datos.

Se recomienda que futuras investigaciones amplíen las pruebas realizadas en este estudio, incorporando escenarios con mayor carga de tráfico y herramientas de monitoreo más avanzadas. Asimismo, sería valioso replicar el modelo en entornos físicos o híbridos para contrastar los resultados obtenidos en simulación. Documentar detalladamente cada etapa del proceso facilitará su adaptación y aplicación en proyectos reales de modernización de redes ISP, especialmente en contextos que demandan eficiencia, escalabilidad y control centralizado.

CAPÍTULO VI

BIBLIOGRAFÍA

Bibliografía

[1] S. d. nPerf, «Barómetro de las conexiones fijas,» nº87 rue de Sèze, 69006 Lyon, Francia, 2022-2023.

[2] Rodrianrrango, «expoispecuador.com,» 5 Agosto 2024. [En línea]. Available: https://expoispecuador.com/explorando-la-expo-isp-ecuador-innovacion-y-tecnologia-en-el-corazon-del-pais/.

[3] B. Oviedo-Bayas, E. Zhuma-Mera, G. Bowen-Calero y B. Patiño-Maisanche, «Voz IP seguras implementadas en redes definidas por software,» *Revista Universidad y Sociedad,* vol. 27, p. 111–127, 2021.

[4] J.-N. Binlun, T.-S. Chin, L.-C. Kwang, Z. Yusoff y R. Kaspin, «Desafíos y dirección de la migración a SDN híbrida en redes ISP,» *Conferencia Internacional IEEE sobre Ingeniería Electrónica y de la Comunicación (ICECE) de 2018 , Xi'an, China,* pp. 60-64, 2018.

[5] B. Dawadi, P. Manzoni, J. Galán-Jiménez, V. Shah y M. Polverini, «Special Issue Topic: SDN migration challenges and practices in ISP/Telcos Networks.,» *Guest Editorial,* 2022.

[6] F. J. Villacís Mora, «BIBDIGITAL,» 2020. [En línea]. Available: https://bibdigital.epn.edu.ec/handle/15000/20682.

[7] J.-D. Pincay Bohórquez, M.-A. Guamán y E. R. Rodríguez, «Arquitectura híbrida para redes distribuidas seguras: integración de SDN, aprendizaje federado y blockchain.,» *Revista Social Fronteriza.*, vol. 5, nº 3, 2025.

[8] G. V. Research, «Informe de análisis del tamaño, la participación y las tendencias del mercado de redes definidas por software por componente (soluciones, servicios), tipo (SDN abierto, SDN mediante API, SDN mediante superposición, SDN híbrido), uso final (empresarial).,» 2025.

[9] B. R. Insights, 2 Mayo 2025. [En línea]. Available: https://www.businessresearchinsights.com/market-reports/sdn-nfv-market-111985.

[10] A. Herrera, «El papel de SDN en la modernización de la red y la reducción de costes,» 2021. [En línea]. Available: https://www.tiaonline.org/resources/whitepapers/role-sdn-network-modernization.

[11] A. Al-Zahrani, B. Saha y S. Jha, «Emulación de redes definidas por software basada en Mininet: desafíos y oportunidades,» *Journal of Network and Computer Applications, 166, 102698.,* 2020.

[12] Cloud, Revista, «La evolución de las arquitecturas de redes hacia las SDN: una revolución tecnológica.,» *RevistaCloud,* 2024.

[13] D. Haro Mendoza, L. Tello Oquendo y L. Marrone, «A comparative evaluation of the performance of open-source SDN controllers.,» *Latin-American Journal of Computing (LAJC),* vol. 7, nº 2, p. 64–77, 2020.

[14] A. Sahbi, F. Jaidi y A. Bouhoula, «Algoritmos de aprendizaje automático para mejorar la detección de intrusiones en SDN/NFV,» *nternational Wireless Communications and Mobile Computing (IWCMC),* pp. 602-607, 2023.

[15] R. d. Oliveira, S. C-M, A.-A. Shinoda y L. Rodrigues Prete, «Uso de Mininet para emulación y prototipado de redes definidas por software,» *Conferencia Colombiana de Comunicaciones y Computación (COLCOM) del IEEE,* pp. 1-6, 2014.

[16] G. Combs, «Wireshark: The network protocol analyzer.,» *Wireshark Foundation,* 2020.

[17] S. Sanjeev y K. J. Rakesh, «A Survey on Software-Defined Networking: Architecture for Next Generation Network,» *arXiv preprint,* Enero 2020.

[18] S. Khorsandroo, A. Gallego Sanchez, A. S. Tosun, J. M. Arco Rodriguez y R. Doriguzzi Corin, «Hybrid SDN Evolution: A Comprehensive Survey of the State-of-the-Art,» *arXiv preprint,* 2021.

[19] G. Neelam, «A Comparative Study of Software Defined Networking Controllers,» *Electronics,* vol. 11, nº 17, 2022.

[20] Z. Xiang, «Mininet: an instant virtual network on your computer,» *ScienceDirect,* 2020.

[21] A. Fidha Ezzaldin y A.-S. Mohammed Basheer, «Performance Analysis of Various SDN Controllers with,» *EAI Endorsed Transactions,* 2022.

[22] K. Ioannis, M. Spiridoula V, B. Ilias y S. Eleftherios, «On the Performance of SDN Controllers in Real World Topologies,» 2024.

[23] P. Ohri y S.-G. Neogi, «Software-Defined Networking Security Challenges and Solutions: A Comprehensive Survey,» *International Journal of Computing and Digital Systems,* vol. 12, nº 1, p. 383–400, Julio 2022.

[24] X. Etxezarreta, I. Garitano, M. Iturbe y U. Zurutuza, «Software-Defined Networking approaches for intrusion detection and response,» *Elsevier,* 2023.

[25] Mininet Project, «Mininet.org,» 28 febrero 2021. [En línea]. Available: https://mininet.org/news/.

[26] L. M. Amaya-Fariño, J. F. Arroyo-Pizarro, M. Jaramillo-Infante, A. Tumbaco-Reyes y B. Mendoza-Morán, «Redes Definidas por Software usando MiniNet.,» *Revista Científica y Tecnológica UPSE,* vol. 9, nº 1, p. 48–56, 2022.

[27] S. Badotra y S.-N. Panda, «Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking.,» *Cluster Computing,* pp. 23, 1281–1291, 2020.

[28] O. Flauzac, F. Nolot y E. M. Gallegos Robledo, «Is Mininet the right solution for an SDN testbed?,» *IEEE (International Conference on Computing, Networking and Communications (ICNC)),* p. 1–6., 2019.

[29] S. Ahmad y A. H. Mir., «Scalability, Consistency, Reliability and Security in SDN,» *Cluster Computing,* pp. 24, 3187–3228., 2021.

[30] Q. Waseem, W. Isni Sofiah Wan Din, A. Aminuddin, M. Hussain Mohammed y R. Alfa Aziza, «Software-Defined Networking (SDN),» *IEEE,* pp. págs. 30-35, 2022.

[31] D. Babu R, A. Thapa, R. Guragain, D. Karki, U. Sandesh P y J. Shashidhar R, «Routing Performance Evaluation of a Multi-Domain Hybrid SDN for Its Implementation in Carrier Grade ISP Networks,» *Applied System Innovation,* vol. Vol. 4, nº No. 3, 2021.

[32] Asamblea Nacional del Ecuador, Constitución de la República del Ecuador., Quito, 2008.

[33] Asamblea Nacional de la República del Ecuador, Ley Orgánica de Telecomunicaciones., Quito, 2015.

[34] H. Khalili, D. Rincón, S. Sallent y J. R. Piney, «An integrated SDN-based architecture for Passive Optical Networks,» Barcelona, IntechOpen, 2018.

[35] T. Suzuki, «Software-based Optical Access Network Virtualization Technology: Facilitating New-service Prototype Provisioning.,» *NTT Technical Review,* vol. 20, nº 11, 2022.

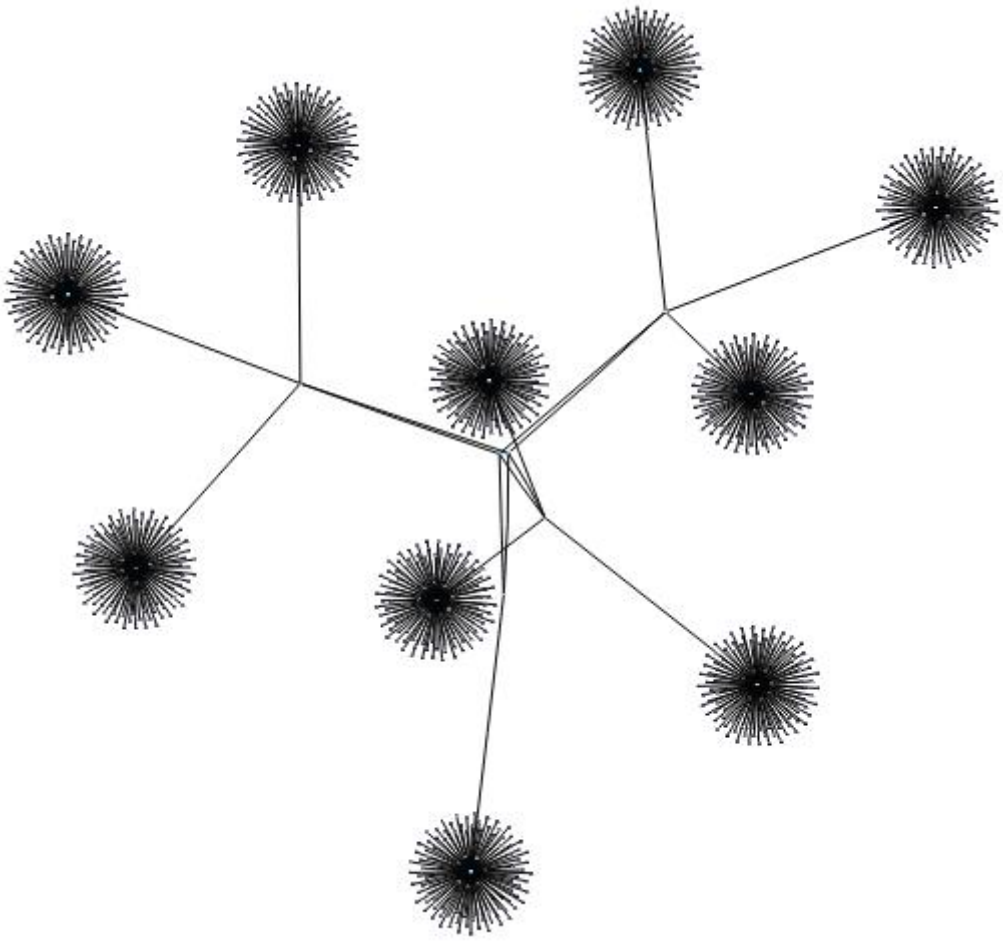
[36] Documentation OpenDaylight, 2023. [En línea]. Available: https://docs.opendaylight.org.

[37] Documentation Open vSwitch, 2023. [En línea]. Available: https://docs.openvswitch.org.

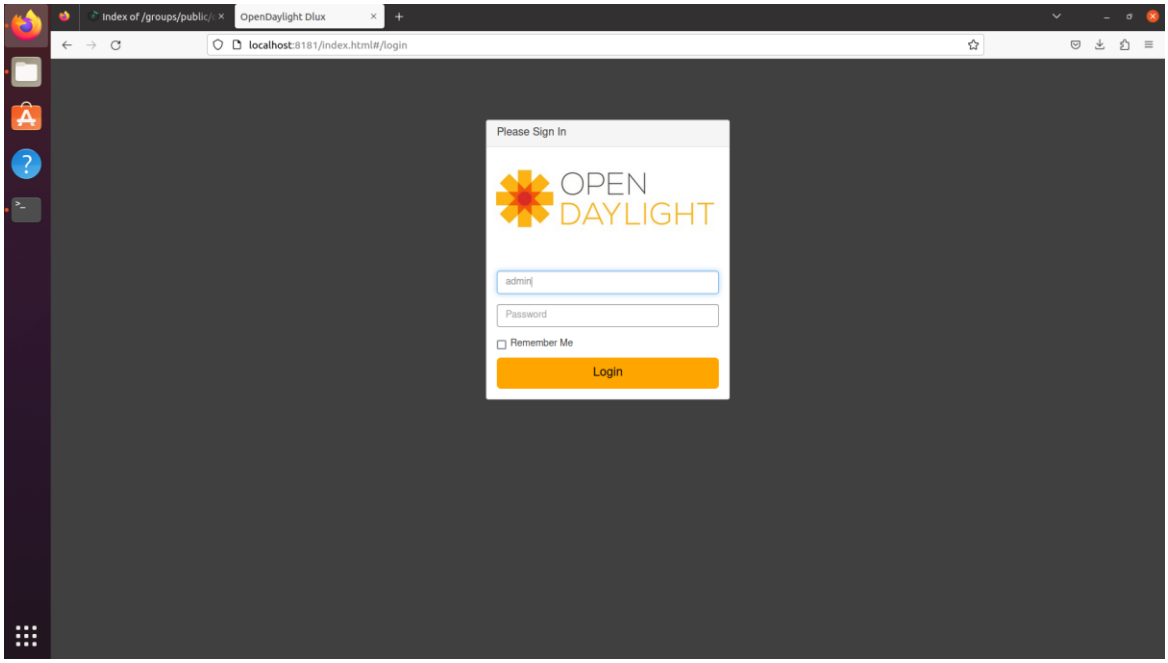
CAPÍTULO VII

ANEXOS

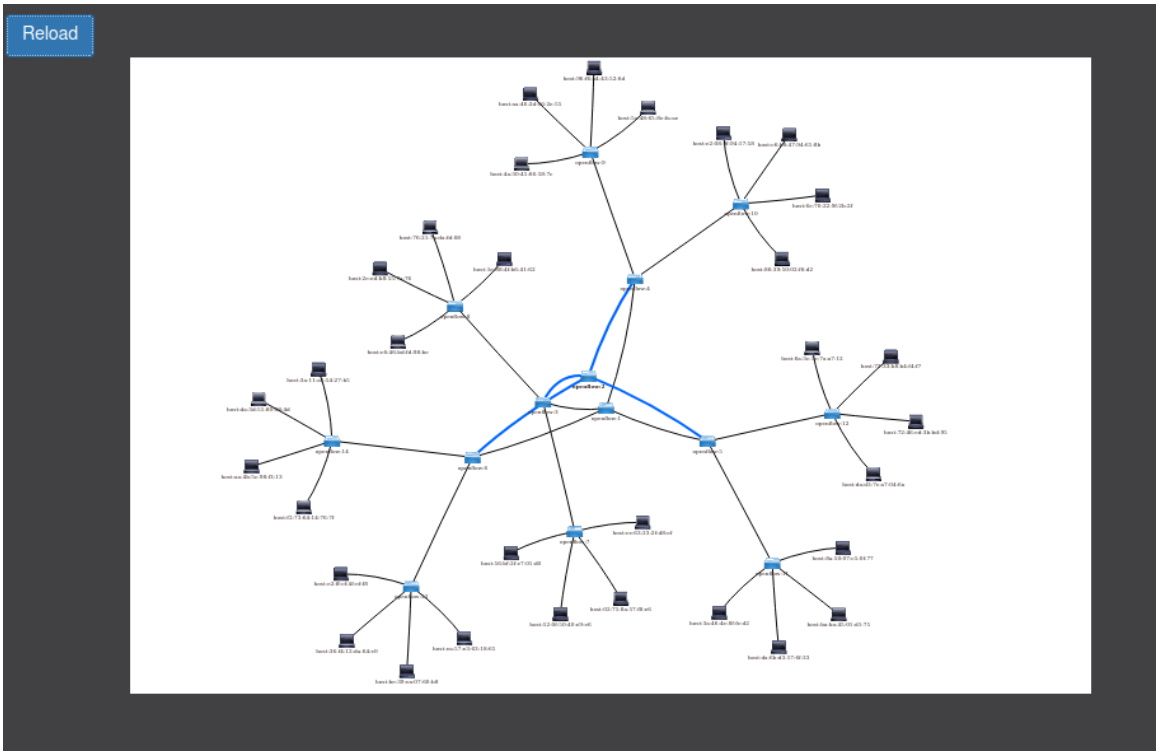
Anexo 1. Vista Global de la Red SDN con 1000 usuarios, vista DLUX de ODL



Anexo 2. Registro en la Plataforma de ODL para Utilizar DLUX



Anexo 3. Pruebas en la Red con distinta cantidad de Usuarios



Anexo 4. Comandos de Instalación de Extensiones como WireShark en Ubuntu.

```
johnpincay@ubuntu:~/Escritorio$
johnpincay@ubuntu:~/Escritorio$ sudo apt update
[sudo] contraseña para johnpincay:
Lo siento, pruebe otra vez.
[sudo] contraseña para johnpincay:
Obj:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Obj:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Des:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Obj:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Descargados 128 kB en 1s (99,8 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 63 paquetes. Ejecute «apt list --upgradable» para verlos.
johnpincay@ubuntu:~/Escritorio$
johnpincay@ubuntu:~/Escritorio$ sudo apt install wireshark -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
wireshark ya está en su versión más reciente (3.2.3-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 63 no actualizados.
johnpincay@ubuntu:~/Escritorio$
johnpincay@ubuntu:~/Escritorio$ sudo usermod -aG wireshark johnpincay
usermod: el grupo «wireshark» no existe
johnpincay@ubuntu:~/Escritorio$ cd ..
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo usermod -aG wireshark johnpincay
usermod: el grupo «wireshark» no existe
johnpincay@ubuntu:~$ sudo usermod -aG wireshark $johnpincay
usermod: el grupo «wireshark» no existe
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo groupadd wireshark
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo usermod -aG wireshark johnpincay
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo chgrp wireshark /usr/bin/dumpcap
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo chmod 750 /usr/bin/dumpcap
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/bin/dumpcap
johnpincay@ubuntu:~$
johnpincay@ubuntu:~$ getcap /usr/bin/dumpcap
/usr/bin/dumpcap = cap_net_admin,cap_net_raw+eip
johnpincay@ubuntu:~$
```

Anexo 5. Script para Iniciar Mininet con el Comando ~/iniciar_odl.sh

```
# Iniciar Mininet con topología en árbol y controlador remoto
sudo mn --topo tree,depth=2,fanout=2 --controller=remote --switch=ovs &

# Esperar unos segundos para que los bridges se creen
sleep 5

# Obtener los nombres de los bridges activos
bridges=$(sudo ovs-vsctl list-br)

# Configurar OpenFlow13 en cada bridge
for bridge in $bridges; do
    echo "Configurando OpenFlow13 en $bridge..."
    sudo ovs-vsctl set bridge $bridge protocols=OpenFlow13
done

echo "Configuración completada."
```

Anexo 6. Script para Instalar Features de ODL

```
# Ruta al directorio de instalación de OpenDaylight
ODL_DIR=~/Escritorio/distribution-karaf-0.6.4-Carbon

# Iniciar Karaf en segundo plano
$ODL_DIR/bin/start

# Esperar 20 segundos para que Karaf esté listo
sleep 20

# Lista de features necesarios
features=(
  "odl-l2switch-switch"
  "odl-l2switch-switch-rest"
  "odl-l2switch-switch-ui"
  "odl-dlux-core"
  "odl-openflowplugin-flow-services-ui"
  "odl-restconf"
  "odl-openflowplugin-all"
)

# Verificar e instalar cada feature
for feature in "${features[@]}"; do
  echo "Verificando feature: $feature"
  estado=$($ODL_DIR/bin/client "feature:list" | grep "^$feature" | grep -E "\s+x\s+|Started")
  if [ $? -ne 0 ]; then
    echo "Instalando $feature..."
    $ODL_DIR/bin/client "feature:install $feature"
  else
    echo "$feature ya está instalado."
  fi
  sleep 2
done

echo "Todos los features han sido verificados e instalados si era necesario."
```

Anexo 7. Script para la Configuración de la Red con 1000 Usuarios.

```
from mininet.topo import Topo

class ISPTesisTopo(Topo):
    def build(self):
        # Switches de núcleo
        core1 = self.addSwitch('s1')
        core2 = self.addSwitch('s2')

        # Switches de distribución
        dist_switches = []
        for i in range(3, 7):  # s3 a s6
            dist = self.addSwitch(f's{i}')
            dist_switches.append(dist)

        # Switches de acceso
        access_switches = []
        for i in range(7, 17):  # s7 a s16 (10 switches)
            acc = self.addSwitch(f's{i}')
            access_switches.append(acc)

        # Enlaces núcleo → distribución
        for dist in dist_switches:
            self.addLink(core1, dist)
            self.addLink(core2, dist)

        # Enlaces distribución → acceso
        for i, acc in enumerate(access_switches):
            dist = dist_switches[i // 3]  # 3 accesos por distribución aprox
            self.addLink(dist, acc)

        # Hosts conectados a switches de acceso
        host_id = 1
        for acc in access_switches:
            for _ in range(100):  # 100 hosts por switch de acceso
                host = self.addHost(f'h{host_id}')
                self.addLink(host, acc)
                host_id += 1

topos = {'isptesis': (lambda: ISPTesisTopo())}
```