



Autónoma
Universidad Autónoma del Perú

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE
ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR
SOFTWARE

PARA OBTENER EL TÍTULO DE

INGENIERO DE SISTEMAS

AUTOR

ANDRES JUNIOR APARCANA TASAYCO

ORCID: 0000-0001-6136-7701

ASESOR

MG. MICHAEL ALEJANDRO CABANILLAS CARBONELL

ORCID: 0000-0001-9675-0970

LÍNEA DE INVESTIGACIÓN DEL PROGRAMA

GESTIÓN ESTRATÉGICA DE TECNOLOGÍAS Y/O SISTEMAS DE INFORMACIÓN

LIMA, PERÚ, OCTUBRE DE 2023



CC BY

<https://creativecommons.org/licenses/by/4.0/>

Esta licencia permite a otros distribuir, mezclar, ajustar y construir a partir de su obra, incluso con fines comerciales, siempre que le sea reconocida la autoría de la creación original. Esta es la licencia más servicial de las ofrecidas. Recomendada para una máxima difusión y utilización de los materiales sujetos a la licencia

Referencia bibliográfica

Aparcana Tasayco, A. J. (2024). *Sistema de monitoreo de red-NMS de detección de fallos de enlace en el proceso de monitoreo de redes definidas por software* [Tesis de pregrado, Universidad Autónoma del Perú]. Repositorio de la Universidad Autónoma del Perú.

HOJA DE METADATOS

Datos del autor	
Nombres y apellidos	Andres Junior Aparcana Tasayco
Tipo de documento de identidad	DNI
Número de documento de identidad	72527217
URL de ORCID	https://orcid.org/0000-0001-6136-7701
Datos del asesor	
Nombres y apellidos	Michael Alejandro Cabanillas Carbonell
Tipo de documento de identidad	DNI
Número de documento de identidad	43426369
URL de ORCID	https://orcid.org/0000-0001-9675-0970
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Wilyam David Torres Meza
Tipo de documento	DNI
Número de documento de identidad	08150222
Secretario del jurado	
Nombres y apellidos	Ana Julieta Gonzalez Garcia
Tipo de documento	Carnet de extranjería
Número de documento de identidad	003020400
Vocal del jurado	
Nombres y apellidos	Ivonne Sadith Musayon Oblitas
Tipo de documento	DNI
Número de documento de identidad	09606289
Datos de la investigación	
Título de la investigación	Sistema de monitoreo de red-NMS de detección de fallos de enlace en el proceso de monitoreo de redes definidas por software
Línea de investigación Institucional	Ciencia, Tecnología e Innovación
Línea de investigación del Programa	Gestión Estratégica de Tecnologías y/o Sistemas de Información
URL de disciplinas OCDE	https://purl.org/pe-repo/ocde/ford#2.02.04

FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
ACTA DE SUSTENTACIÓN DE TESIS

En la ciudad de Lima, el Jurado de Sustentación de Tesis conformado por: DR. WILYAM DAVID TORRES MEZA como presidente, la MG. ANA JULIETA GONZALEZ GARCIA como secretaria y la DRA. IVONNE SADITH MUSAYON OBLITAS como vocal, reunidos en acto público para dictaminar la tesis titulada:

**SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE
EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE**

Presentada por el bachiller:

ANDRES JUNIOR APARCANA TASAYCO

Para obtener el **Título Profesional de Ingeniero de Sistemas**; luego de escuchar la sustentación de la misma y resueltas las preguntas del jurado se procedió a la calificación individual, obteniendo el dictamen de **Aprobado-Excelente** con una calificación de **DIECINUEVE(19)**.

En fe de lo cual firman los miembros del jurado, el 27 de octubre del 2023.



PRESIDENTE
DR. WILYAM DAVID
TORRES MEZA



SECRETARIO
MG. ANA JULIETA
GONZALEZ GARCIA



VOCAL
DRA. IVONNE SADITH
MUSAYÓN OBLITAS


ACTA DE APROBACIÓN DE ORIGINALIDAD

Yo Michael Alejandro Cabanillas Carbonell docente de la Facultad de Ingeniería y Arquitectura de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Autónoma del Perú, en mi condición de asesor de la tesis titulada:

SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE Del bachiller Andres Junior Aparcana Tasayco, certifico que la tesis tiene un índice de similitud de 12% verificable en el reporte de similitud del software Turnitin que se adjunta.

El suscrito revisó y analizó dicho reporte a lo que concluyó que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad Autónoma del Perú.

Lima, 12 de enero del 2024



Michael Alejandro Cabanillas Carbonell

DNI: 43426369

DEDICATORIA

A mi familia que con el esfuerzo y dedicación han ayudado a cumplir la meta que se trazó hace 5 años, su apoyo y resiliencia han constituido parte fundamental de este desarrollo académico.

AGRADECIMIENTOS

Agradezco a los Mg. Daniel Díaz Ataucuri, Mg. Michael Cabanillas Carbonell, Dr. Javier Gamboa Cruzado, Mg. Johny Pretell Cruzado e Mg. Celis Henry Ochoa Jayo, por brindar el apoyo teórico y científico pertinente para la presente investigación, a los compañeros y a los docentes de la Facultad de Ingeniería y Arquitectura por sus valiosos aportes.

ÍNDICE

DEDICATORIA	2
AGRADECIMIENTOS.....	3
LISTA DE TABLAS	5
LISTA DE FIGURAS.....	6
RESUMEN	9
ABSTRACT	10
CAPÍTULO I: INTRODUCCIÓN	11
CAPÍTULO II: METODOLOGÍA.....	14
2.1. Tipo y diseño de la investigación.....	69
2.2. Población, muestra y muestreo	69
2.3. Hipótesis.....	71
2.4. Variables y operacionalización	71
2.5. Técnicas e instrumentos de recolección de datos, validez y confiabilidad	72
2.6. Procedimientos.....	79
2.7. Análisis de datos	85
2.8. Aspectos éticos	86
CAPÍTULO III: RESULTADOS	119
CAPÍTULO IV: DISCUSIÓN	135
CAPÍTULO V: CONCLUSIONES.....	138
CAPÍTULO VI: RECOMENDACIONES	141
REFERENCIAS	
ANEXOS	

LISTA DE TABLAS

Tabla 1	Indicadores en preprueba
Tabla 2	Tabla comparativa de Modelos de Redes Tradicionales y sus respectivos Protocolos de Gestión
Tabla 3	Comparativa de controladores
Tabla 4	Cantidad de pruebas realizadas
Tabla 5	Pruebas y características de cada proceso de control
Tabla 6	Técnicas, instrumentos y herramientas de la investigación
Tabla 7	Coeficiente de Relación de Pearson
Tabla 8	Prueba de normalidad del indicador: tiempo de visualización de la topología
Tabla 9	Prueba de confiabilidad del indicador: tiempo de visualización de la topología
Tabla 10	Prueba de normalidad del indicador: cantidad de uso de memoria
Tabla 11	Prueba de confiabilidad del indicador: cantidad de uso de memoria
Tabla 12	Prueba de normalidad del indicador: cantidad de uso de CPU
Tabla 13	Prueba de confiabilidad del indicador: cantidad de uso de CPU
Tabla 14	Lista de especialistas que avalaron la validez del instrumento
Tabla 15	Recursos informáticos
Tabla 16	Recursos de la investigación
Tabla 17	Frecuencia del indicador de tiempo de visualización de topología
Tabla 18	Frecuencia del indicador de cantidad de uso de memoria
Tabla 19	Frecuencia del indicador de cantidad de uso de CPU
Tabla 20	Prueba de normalidad del indicador de tiempo de visualización de topología
Tabla 21	Prueba de normalidad del indicador de cantidad de uso de memoria
Tabla 22	Prueba de normalidad del indicador de cantidad de uso de CPU
Tabla 23	Resumen del modelo del indicador 1: Tiempo de Visualización de Topología
Tabla 24	Resumen del modelo del indicador 2: Cantidad de Uso de Memoria
Tabla 25	Resumen del modelo del indicador 1: Cantidad de Uso de CPU

LISTA DE FIGURAS

- Figura 1 Número de publicaciones sobre SDN entre los años 2016-2020
- Figura 2 Proceso actual de Monitoreo de Redes Definidas por Software (AS-IS)
- Figura 3 Proceso propuesto de Monitoreo de Redes Definidas por Software (TO-BE)
- Figura 4 Capas del Modelo OSI
- Figura 5 Comparativa de las Capas del Modelo OSI con la Arquitectura TCP/IP
- Figura 6 Comunicación simple entre un Gestor y un agente, y sus MIB
- Figura 7 Estructura de Información de Gestión OBJECT-TYPE
- Figura 8 Modelo SNMP
- Figura 9 Arquitectura SDN
- Figura 10 Diagrama de perspectiva arquitectónica de los controladores
- Figura 11 Enfoques arquitectónicos de los controladores
- Figura 12 Estructura tradicional de un Sistema de Gestión de Red-NMS
- Figura 13 Proceso de Monitoreo de Red
- Figura 14 Arquitectura de referencia para el acceso remoto de laboratorios
- Figura 15 Fotografía de referencia de laboratorio SDN de la FIEE-UNI
- Figura 16 Comandos para controlador SDN OpenDayLight
- Figura 17 Ejecución del sistema de monitoreo de red (NMS)
- Figura 18 Ejecución de herramienta de red: Cbench
- Figura 19 Ejecución de herramienta para memoria y CPU
- Figura 20 Pantallas de tiempo de visualización de componentes topología
- Figura 21 Aplicación SDN en el paradigma de Redes Definidas por Software (SDN)
- Figura 22 Arquitectura de Software del Sistema de Monitoreo de Red
- Figura 23 EPIC-1 Detección de topología y estado de enlace, alcance de la tesis
- Figura 24 EPIC-2 Monitoreo de estadísticas de la red, alcance de la tesis
- Figura 25 Elaboración de la Característica del Sistema 1
- Figura 26 Elaboración de la Característica del Sistema 2
- Figura 27 Elaboración de la Característica del Sistema 3
- Figura 28 Historia de Usuario 1
- Figura 29 Historia de Usuario 2
- Figura 30 Historia de Usuario 3

- Figura 31 Historia de Usuario 4
- Figura 32 Historia de Usuario 5
- Figura 33 Historia de Usuario 6
- Figura 34 Historia de Usuario 7
- Figura 35 Historia de Usuario 8
- Figura 36 Historia de Usuario 9
- Figura 37 Historia de Usuario 10
- Figura 38 Historia de Usuario 11
- Figura 39 Scrumboard Sprint 1-1
- Figura 40 Scrumboard Sprint 1-2
- Figura 41 Diagramas de explicación de protocolo LLDP y monitoreo en Redes Legadas y Redes Definidas por Software
- Figura 42 Diagrama de actividades del monitoreo de red en controlador POX
- Figura 43 Scrumboard Sprint 2-1
- Figura 44 Scrumboard Sprint 2-2
- Figura 45 Arquitectura del Sistema simplificada bajo el paradigma SDN
- Figura 46 Scrumboard Sprint 3-1
- Figura 47 Scrumboard Sprint 3-2
- Figura 48 Modelo de base de datos no relacional
- Figura 49 Diagrama de Actividad entre aplicación Servidor y Controlador SDN
- Figura 50 Diagrama de Paquetes de la Aplicación Back-end
- Figura 51 Segundo Prototipo Funcional del Sistema de Monitoreo de Red SDN
- Figura 52 Burndown Chart del Sprint 3
- Figura 53 Scrumboard Sprint 4-1
- Figura 54 Scrumboard Sprint 4-2
- Figura 55 Burndown Chart del Sprint 4
- Figura 56 Pantalla del Sistema de Monitoreo de Red SDN para el Sprint 4
- Figura 57 Scrumboard Sprint 5-1
- Figura 58 Scrumboard Sprint 5-2
- Figura 59 Scrumboard Sprint 5-3
- Figura 60 Pantalla final del sistema de monitoreo de red-NMS. Vista principal y detección de enlaces caídos
- Figura 61 Pantalla final del sistema de monitoreo de red-NMS. Vista de switches

- Figura 62 Pantalla final del sistema de monitoreo de red-NMS. Vista del Switch 5
- Figura 63 Pantalla final del sistema de monitoreo de red-NMS. Vista del puerto Openflow:5 del Switch 5
- Figura 64 Pantalla final del sistema de monitoreo de red-NMS. Vista de las tablas de flujos del Switch 5
- Figura 65 Burndown Chart del Sprint 5
- Figura 66 Tiempo de respuesta del componente topología con respecto al número de switches
- Figura 67 Cantidad de uso de memoria RAM con diferentes configuraciones de switches
- Figura 68 Porcentaje de uso de CPU con diferentes configuraciones de switches
- Figura 69 Prueba de normalidad del tiempo de visualización de topología en la preprueba
- Figura 70 Prueba de normalidad del tiempo de visualización de topología en la postprueba
- Figura 71 Prueba de normalidad de la cantidad de uso de memoria en la preprueba
- Figura 72 Prueba de normalidad de la cantidad de uso de memoria en la postprueba
- Figura 73 Prueba de normalidad de cantidad de uso de CPU en la preprueba
- Figura 74 Prueba de normalidad de cantidad de uso CPU en la postprueba

SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE

ANDRES JUNIOR APARCANA TASAYCO

UNIVERSIDAD AUTÓNOMA DEL PERÚ

RESUMEN

Las Redes Definidas por Software (SDN) son una tecnología que emplea el paradigma de la programabilidad y es crucial para preparar a futuros especialistas en telecomunicaciones para comprender otras tecnologías de vanguardia como 5G, el Internet de las cosas y SD-WAN. Los estudiantes universitarios volverán a los laboratorios de redes de datos en un entorno semipresencial bajo la nueva normalidad creada por COVID-19. Para la educación de futuros profesionales de las telecomunicaciones, es crucial contar con una aplicación de sistema de monitoreo de red (NMS) que muestre los escenarios implementados de manera completa, en tiempo real y sea adaptable a nuevos requisitos. El NMS propuesto utiliza una base de datos no relacional, OpenDayLight como controlador SDN y una arquitectura cliente-servidor para mostrar de manera asíncrona los resultados en una aplicación web utilizando NeXt-UI. El objetivo fue proporcionar un marco que facilite a los estudiantes comprender cómo funcionan estas redes SDN y permita una mejor representación de los escenarios SDN. Los resultados revelaron que el NMS influye en los indicadores de tiempos de visualización de la topología, la cantidad de uso de Memoria y uso de CPU en todos los casos aumentando en promedio 79%, 70% y 76.35%, respectivamente con un nivel de confianza del 95%.

Palabras clave: sistema de monitoreo de red (nms), redes definidas por software (sdn), monitoreo de redes, scrum

NETWORK MONITORING SYSTEM-NMS FOR LINK FAULT DETECTION IN MONITORING PROCESS OF SOFTWARE-DEFINED NETWORKS

ANDRES JUNIOR APARCANA TASAYCO

UNIVERSIDAD AUTÓNOMA DEL PERÚ

ABSTRACT

Software Defined Networking (SDN) is a technology that employs the programmability paradigm, and is crucial for preparing future telecom specialists to comprehend other cutting-edge technologies like 5G, the Internet of Things, and SD-WAN. Under the new circumstances established by the COVID-19 pandemic, university students will resume their participation in data networking laboratories with a semi-presential approach. For the education of future telecom professionals, it is crucial to have a network monitoring system (NMS) application that displays implemented scenarios completely, in real time, and is adaptable to new requirements. The proposed NMS uses a non-relational database, OpenDayLight as an SDN controller, and a server-client architecture to asynchronously display the results on a web app using NeXt-UI. The objective was to provide a framework that makes it easier for students to grasp how these SDN networks function and allows for a better depiction of the SDN scenarios. The results showed that the NMS influences the indicators of topology visualization times, Memory usage and CPU usage in all cases increasing on average 79%, 70% and 76.35%, respectively with a confidence level of 95%.

Keywords: network monitoring system (nms), software-defined networks (sdn), network monitoring, scrum

CAPÍTULO I
INTRODUCCIÓN

Hoy en día, debido al Covid-19 declarada oficialmente por la OMS, el mundo vive un problema de salud pública que ha originado cambios en las relaciones entre los seres humanos. En la nueva normalidad originada a causa de la pandemia, los alumnos de las universidades retornan a los laboratorios de redes de datos en un escenario presencial o semipresencial, en particular, el laboratorio de redes definidas por software (SDN) de la Universidad Nacional de Ingeniería.

Este laboratorio es trascendental para los estudiantes de telecomunicaciones, puesto que permite entender de manera práctica las nuevas tecnologías emergentes como SD-WAN, conceptos como la programabilidad y su relación con otras que ya empiezan a ser planificada en nuestro país como 5G.

El laboratorio con tecnología SDN tiene la necesidad de un sistema de monitoreo que permita visualizar el estado de la red de manera inmediata; permitiendo realizar un seguimiento de las acciones que los alumnos realicen, en la nueva normalidad, cuando hagan uso de este laboratorio tanto local como remotamente.

En el presente informe de Tesis, el sistema de monitoreo propuesto hace uso de base de datos no relacional, OpenDayLight como controlador SDN y una arquitectura que utiliza un servidor que asincrónicamente muestra los resultados en un cliente con el uso de NeXt-UI. El principal aporte es disponer de un sistema que permite una mejor visualización de los escenarios de SDN desarrollados y que facilite a los alumnos realizar mejoras para comprender el funcionamiento de SDN.

Esta investigación ha planeado la hipótesis general que un Sistema de Monitoreo de Red-NMS de detección fallos de enlace influye en el proceso de monitoreo de redes definidas por software. Para abordar esta cuestión, se ha adoptado el marco de trabajo "Scrum", que se destaca por su capacidad de desarrollo ágil y su adaptabilidad a los cambios en los requisitos de investigación.

Los resultados obtenidos revelan que el NMS influye significativamente en indicadores clave de rendimiento. Tanto los tiempos de visualización de la topología como la cantidad uso de memoria y el uso de CPU han experimentado un aumento promedio del 79%, 70% y 76.35%, respectivamente. Estos hallazgos son respaldados con un nivel de confianza del 95%.

La estructura de este documento es la siguiente: El capítulo I aborda detalladamente la realidad problemática, los objetivos planteados, la justificación e importancia del estudio, así como también se exponen las limitaciones identificadas. El capítulo II proporciona un análisis de los antecedentes, se presenta el marco teórico pertinente y se definen los conceptos clave. En el capítulo III, se brinda una descripción minuciosa del tipo y diseño de la investigación, incluyendo detalles acerca de la población y muestra de estudio, las hipótesis formuladas, las variables consideradas, así como los métodos y técnicas empleados para la investigación. Además, se detallan las estrategias de procesamiento y análisis de los datos recopilados. El capítulo IV abarca el estudio de factibilidad, el modelado del NMS y la metodología empleada. En el capítulo V, se realiza un análisis de la confiabilidad de las variables examinadas, se presentan e interpretan los resultados obtenidos y se lleva a cabo la contrastación de las hipótesis planteadas. Finalmente, en el capítulo VI, se lleva a cabo la discusión de los resultados y se presentan las conclusiones y recomendaciones correspondientes.

CAPÍTULO II

METODOLOGÍA

Realidad problemática

Descripción de la realidad problemática

Como consecuencia de la pandemia del Covid-19 declarada oficialmente por la OMS, las relaciones entre los seres humanos ha cambiado en el mundo. En el ámbito educativo ha cambiado sus actividades de presenciales a virtuales; más del 80% de países han experimentado cierres en sus centros de estudios (WORLD BANK, 2020b).

Esta pandemia ha puesto en la discusión pública el término “nueva normalidad”, esta se refiere principalmente a una convivencia de actividades de la anterior normalidad junto a protocolos de bioseguridad que cambian nuestras conductas con las lecciones aprendidas (UNESCO, 2020). Por ende, el sistema educativo en diferentes niveles cambiará con esta “nueva normalidad” en los próximos años (Inter-American Development Bank, 2020). Las telecomunicaciones están permitiendo que los servicios públicos de salud, transporte y educación se acondicionen a esta “nueva normalidad”. Según Bafoutsou et al. (2020), los volúmenes de tráfico persistentemente aumentaron como resultado del mayor uso de herramientas de colaboración en línea para el trabajo y la educación remota.

Esto lleva a que cobren relevancia tecnologías como SDN; donde la flexibilidad y escalabilidad de SDN/NFV proporciona a los proveedores de telecomunicaciones una forma rápida y dinámica de responder a los cambios para cumplir con las demandas (NSTAC, 2020).

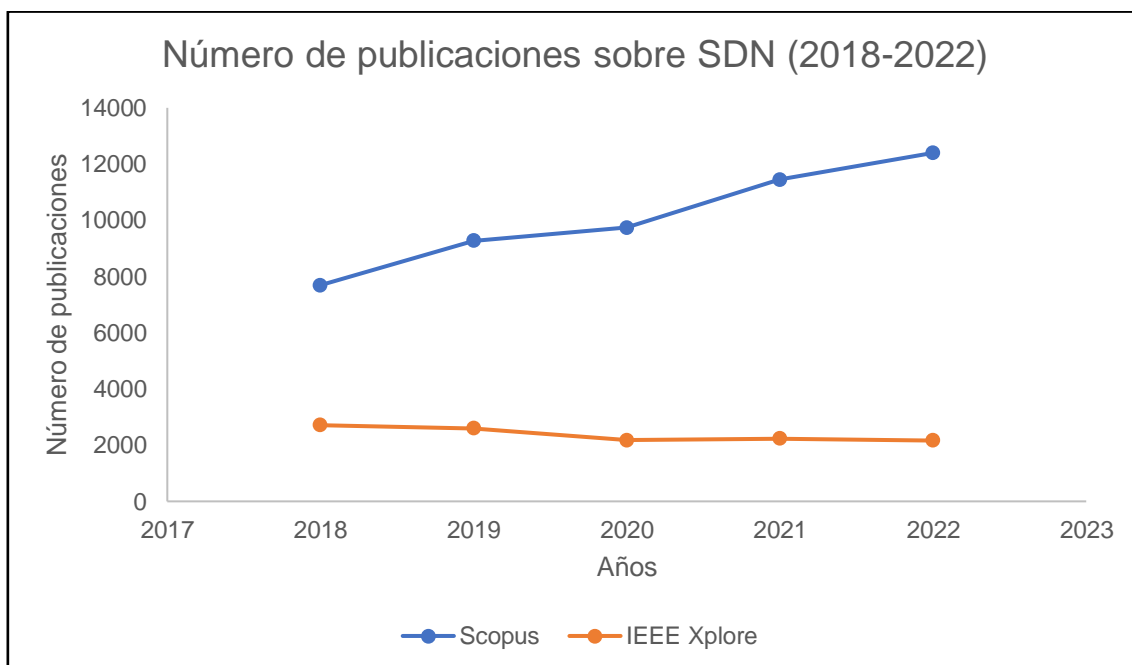
Las Redes Definidas por Software (Software Defined Networking-SDN) surgen como un nuevo paradigma en las redes que promete simplificar la gestión de redes y permitir su innovación (Bonfim et al., 2019). La evolución de SDN, denominada red WAN definidas por Software, SD-WAN, tomará un rol protagónico en las redes

basadas en servicios en las implementaciones de 5G debido a la necesidad de disponer de redes más inteligentes, control centralizado y la evolución hacia la programabilidad. En el dominio empresarial, “la adopción de SDN se justifica por su alto rendimiento a bajo costo y comercialmente viable acceso a Internet” (NSTAC, 2020, p. 32). Marcado por este incremento, de acuerdo a (Cisco, 2020), la automatización de red (25%), SDN (23%), y Redes basadas en Intenciones-IBN (16%) serán las tecnologías que más impactarán en los próximos 5 años.

Las principales universidades del mundo están investigando en SDN, prueba de ello están los artículos indexados y tesis que se han generado en las principales bases de datos científicas como Scopus y IEEE Xplore, detallado en la Figura 1 se muestra la búsqueda de artículos con "sdn" o "software-defined network" o "software defined network".

Figura 1

Número de publicaciones sobre SDN entre los años 2018-2022



Nota: Estadísticas de las bases de datos: Scopus e IEEE Xplore

El Perú no ha sido ajeno a esta problemática de la pandemia. El gobierno peruano anunció la distribución de más de 80 0000 tablets para alumnos y 97 000 para profesores (WORLD BANK, 2020a) y ha tenido que replantear la forma de educación hacia una remota en el año 2020 (MINEDU, 2020) con políticas que flexibilicen pero que aún mantengan la mayoría de instituciones de educación cerradas para el año 2021 (MINEDU, 2021).

De acuerdo a la tendencia internacional, el gobierno peruano ha comenzado a priorizar y desplegar las tecnologías de telecomunicaciones necesarias para los sectores económicos y educativos como 5G, siendo el país líder en la región en adoptar esta tecnología (MTC, 2021).

Sin embargo, durante este período de pandemia muchas actividades académicas involucradas con el despliegue de material de enseñanza en la educación superior se paralizaron, una de estas actividades es la de investigación y desarrollo con la utilización de los laboratorios que no estaban pensados inicialmente para el acceso remoto y que se vieron obligados a cerrar o permitir un número limitado de investigadores (PCM, 2020).

Actualmente, el Instituto Nacional de Investigación y Capacitación de Telecomunicaciones (INICTEL-UNI) como parte de la Universidad Nacional de Ingeniería-UNI viene desarrollando actividades de investigación conjuntamente con la Facultad de Ingeniería Eléctrica y Electrónica-FIEE. Esta facultad cuenta desde enero de 2020 con un laboratorio de desarrollo en SDN con el objetivo de fortalecer el aprendizaje de la especialidad de telecomunicaciones y realizar investigación en esta área. Este laboratorio de SDN está conformado por un controlador con el sistema operativo de red OpenDayLight y 5 switches OpenFlow: laboratorio que es utilizado en la nueva normalidad tanto por los alumnos como profesores Esta tecnología SDN

va ampliar los campos de investigación hacia la ciberseguridad en el país; como ya se viene iniciando en colaboración del Universidad Nacional de Ciencia y Tecnología de Seoul (SeoulTech, 2020).

El uso de este laboratorio SDN va complementar la formación de los estudiantes de telecomunicaciones y entender de manera práctica las nuevas tecnologías emergentes como SDN, SD-WAN, conceptos como la programabilidad y su relación con otras que ya empiezan a ser planificada en nuestro país como 5G.

Un tema pendiente en este laboratorio con tecnología SDN es la necesidad de un sistema de monitoreo con el cual visualizar el estado de la red del laboratorio SDN de manera inmediata; permitiendo realizar un seguimiento de las acciones que los alumnos realicen, en la nueva normalidad, cuando hagan uso de este laboratorio SDN tanto local como remotamente.

Definición del problema

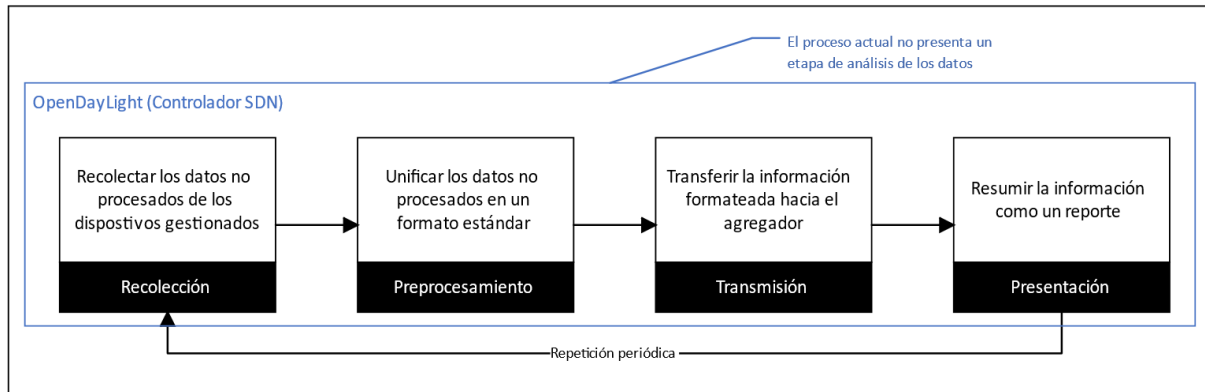
Actualmente, el INICTEL-UNI, como parte de la UNI tiene acceso al laboratorio moderno de Redes Definidas por Software de la FIEE-UNI para la investigación y desarrollo, y aun no se dispone de un sistema de monitoreo que permita conocer el estado de los dispositivos y los enlaces que forman este laboratorio.

De allí, es necesario disponer de un sistema de monitoreo del estado de red a modo de brindar una visualización remota de la red del laboratorio SDN. El proceso actual de monitoreo de redes definidas por software en modelo AS-IS se puede ver en la

Figura 2. Este modelo muestra las etapas implementadas actualmente donde se resalta la ausencia de una etapa de análisis que permita la detección de enlaces caídos como un problema dentro del proceso.

Figura 2

Proceso actual de Monitoreo de Redes Definidas por Software (AS-IS)



La Tabla 1 muestra los valores medidos en la preprueba para los indicadores definidos a continuación.

Tabla 1

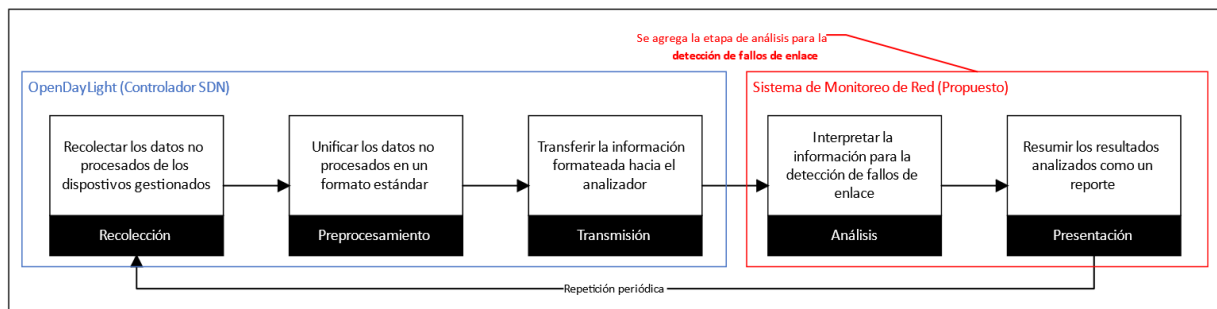
Indicadores en preprueba

Indicadores	Preprueba
Tiempo de visualización de la topología.	1021 milisegundos
Cantidad de uso de memoria.	2542892 KiloBytes
Cantidad de uso de CPU.	3,59 %

Con el objetivo de resolver el problema mencionado se propone un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace para el proceso de monitoreo de redes definidas por software. En la Figura 3 se presenta el proceso propuesto en modelo TO-BE.

Figura 3

Proceso propuesto de Monitoreo de Redes Definidas por Software (TO-BE)



Formulación del problema

Problema general

¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de Redes Definidas por Software?

Problemas específicos

- ¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología?
- ¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria?
- ¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de CPU?

Justificación e importancia de la investigación

Justificación teórica

Las Redes Definidas por Software (SDN, Software-Defined Networking) contiene tres componentes: plano de control que consiste de uno o más controladores, plano de datos que contiene los dispositivos de red y el plano de aplicación donde se ejecutan las aplicaciones de gestión de la red (Hamdan et al., 2021). SDN surgió como respuesta a las necesidades y naturaleza dinámica de las

aplicaciones modernas, como una arquitectura para la escalabilidad y flexibilidad en la configuración de la red. Siendo SDN basado en flujos permite la adquisición de información en tiempo real por medio del protocolo OpenFlow y la separación de los planos de datos y de control permite mayor control sobre los flujos de datos (Tang et al., 2018).

La importancia de la gestión de la red en SDN se asienta en el plano de aplicación o, también denominado, plano de gestión. El plano de gestión incluye diferentes servicios y aplicaciones de red que gestionan el comportamiento de la red como el balanceo de carga, los firewalls, etc. El plano de gestión configura y monitorea los dispositivos de red (Keshari et al., 2021). Las funciones importantes para la gestión de la red como el monitoreo son más eficientes mediante una lógica centralizada (Zhao et al., 2019).

La implementación de SDN está haciendo más fácil la administración de redes y permitiendo configuraciones de red eficientes para optimizar el rendimiento y el monitoreo a través de la programabilidad de la red. SDN tiene la capacidad de soportar la naturaleza dinámica de las funciones de red y las aplicaciones inteligentes del futuro, lo que se traduce en costos operativos más bajos gracias a la simplificación del hardware, software y gestión (Zhao et al., 2019).

De acuerdo a Tsai et al. (2018) dice que el proceso del monitoreo de red es un concepto importante en la gestión de la red que ayuda a los administradores de la red para determinar las conductas de la red y sus componentes. El monitoreo de red provee las bases para posteriores investigaciones como el balanceo de tráfico, quality of service (QoS) y detección de fallas y anomalías.

El monitoreo provee funciones para la descubrimiento y recuperación de fallos. Según Yu et al. (2019), dentro de la red, una **falla** es la imposibilidad de que la red o

uno de sus componentes ejerzan correctamente sus funciones; un **error** es una acción humana u otro factor que produzca un resultado erróneo; y un **defecto** es la manifestación de un error en forma de una condición incorrecta o defecto que puede causar que la red se comporte de manera diferente a la prevista. Por ende, el resultado de un error es un fallo, y un fallo puede llevar a una avería. La ocurrencia de fallos, errores, y averías son las más comunes y directas formas por la cuáles la fiabilidad de la red es puesta en duda. Con lo expuesto, el diseño de soluciones de manejos de fallos es indispensable para satisfacer la necesidad de fiabilidad en las redes (Yu et al., 2019).

Justificación tecnológica

Hoy en día los administradores de red se enfrentan a la mejora y el rediseño de la red, por lo tanto, es necesario la información del estado de la red, lo cual es beneficiado por la característica de SDN al tener una lógica centralizada (Keshari et al., 2021). En las redes tradicionales a gran escala es bien conocida la alta complejidad de la gestión de redes. Por lo tanto, es imperativo que se establezca un sistema de gestión de redes (NMS, *Network Management System*) efectivo capaz de monitorear y controlar la red de manera centralizada, distribuida o jerárquica. El NMS en SDN aprovecha la simplicidad de los switches SDN para obtener el estado de la red en “tiempo real” o “casi real”.

Las tecnologías utilizadas en la investigación tienen gran aceptación en el mundo académico y comercial. SDN se alza como una prometedora arquitectura de centralización de las redes puesto que facilita la adecuación de la red a las aplicación que requieren cada vez más recursos y servicios especializados (Tang et al., 2018). Despierta interés en los sectores comerciales puesto que ofrece diferentes soluciones

de mejoras de procesos y oportunidades de negocio para las organización cada vez más demandantes e innovadoras (Jiménez et al., 2021).

Justificación práctica

Según Yu et al. (2019) en la gestión de fallas en las redes SDN se requieren nuevas técnicas que aprovechen la arquitectura centralizada de flujo de datos. El contexto temporal en los flujos permitiría utilizar herramientas y algoritmos en la detección de fallos para aumentar su fiabilidad en las redes.

Dicho lo anterior, existen pocas herramientas que aprovechen modelos secuenciales para el manejo de fallos en la red. Esto hace práctico la utilización de este para medir su utilidad en este proceso de tanta importancia para las redes de arquitectura SDN.

Objetivos de la investigación

Objetivo general

Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de Redes Definidas por Software.

Objetivos específicos

- Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye el tiempo de visualización de la topología.
- Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye la cantidad de uso de memoria.
- Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye la cantidad de uso de CPU.

Limitaciones

Temporal: La investigación tiene como meta realizarse desde noviembre del año 2020 y a darse por concluido a diciembre del año 2021.

Espacial: La investigación se realizó en un entorno de redes SDN especializado con la plataforma Mininet en el laboratorio de ciberseguridad y redes del INICTEL-UNI.

Conceptual: El informe de tesis tiene como base conceptual la metodología SCRUM, enfocándola al desarrollo de un Sistema de Monitoreo de Red-NMS para la detección de fallos de enlace en SDN.

Antecedentes de estudios

Antecedentes internacionales

Montoya-Munoz et al. (2021) presentan un estudio sobre la falta de una especificación del Plano de Gestión en SDN y proponen un enfoque agnóstico a la tecnología utilizada para abordar este problema. El enfoque se sustenta en una colección de modelos de datos YANG y sus relaciones, que brindan soporte para las áreas de Falla, Configuración, Contabilidad, Rendimiento y Seguridad (FCAPS) desde un Plano de Gestión especializado en SDN. La evaluación del enfoque muestra la eficiencia de su solución en términos de tiempo de respuesta para ejecutar la operación SetController. Los resultados indican que tanto la Solución Integrada YANG logra tiempos de respuesta admisibles (≤ 10 segundos), siendo la Solución Integrada YANG ligeramente más lenta a sus competidoras, pero adecuada para redes pequeñas y medianas. El estudio aporta a la presente investigación en la importancia de considerar el tiempo de respuesta como una métrica crítica en el desarrollo de herramientas de monitoreo para redes definidas por software.

AlZoman y Alenazi (2020) planteó que el concepto de ciudades inteligentes trae consigo la necesidad de tecnologías de la información y la comunicación (TIC) para la mejora de diversos aspectos de la vida urbana. La SDN representó una solución prometedora para simplificar la administración de la red al separar los planos de control y reenvío de paquetes. Esto permitió que los administradores de red manejaran los elementos de la red mediante software, sin importar las especificaciones del vendedor. Las ciudades inteligentes consisten en dispositivos heterogéneos que generan una enorme cantidad de datos en tiempo real, lo que demanda una gestión más dinámica y eficiente; por lo tanto, la integración de SDN y ciudades inteligentes permiten un sistema de gestión de redes simplificado, dinámico

y flexible. Los fallos de enlace debidos a desastres naturales suponen un gran riesgo para el rendimiento de la comunicación de la ciudad inteligente. Debido a esto, se propuso un sistema diseñado para redes de ciudades inteligentes que utiliza SDN para proporcionar calidad de servicio (QoS) frente a fallos de enlace. El sistema propuesto fue evaluado con el tráfico de datos típico de la ciudad inteligente (AlZoman y Alenazi, 2020). Este trabajo aporta a la investigación en la introducción de un sistema SDN de detección de fallos de enlace en ciudades inteligentes.

Yang et al. (2019) afirmó que la virtualización de las funciones de red (NFV) amplía la funcionalidad proporcionada por las redes definidas por software (SDN). Se trata de una tecnología de virtualización que pretende sustituir la funcionalidad proporcionada por el hardware de red tradicional mediante soluciones de software. De este modo, permite un despliegue y una gestión de la red más barata y eficiente. Se previó que el uso de NFV y SDN mejore el rendimiento de las nubes de infraestructura como servicio (IaaS). Sin embargo, debido a la presencia de un gran número de dispositivos de red en las nubes IaaS que ofrecen una plétora de servicios en red, es necesario desarrollar un sistema de monitorización del tráfico para la red eficiente. Este artículo propuso un sistema extensible de monitorización del tráfico de red habilitado por SDN y NFV. Los autores concluyeron que el sistema propuesto puede igualar el rendimiento de las redes tradicionales con costes más bajos y añadiendo más flexibilidad a las tareas de gestión de la red. Este artículo aportó el esquema de una aplicación de monitorización de tráfico de una red SDN-NFV,

Wang et al. (2019) planteó que las estadísticas de red oportunas y precisas son esenciales para las tareas críticas de gestión de redes, como la detección de anomalías y el balanceo de tráfico de datos. La red definida por software (SDN) administra la red mediante el controlador centralizado que puede acceder a los datos

del dispositivo subyacente de forma flexible. Los autores implementaron una herramienta de monitoreo de redes definidas por software SCSCDaylight basada en OpenDayLight, que es un controlador SDN open-source. La herramienta admite la partición de topología de red, la colaboración de múltiples controladores, la velocidad de enlace bidireccional y la medición de la tasa de pérdida de paquetes. La evaluación se hizo mediante la medición de la pérdida de paquetes con la herramienta *iperf* (Wang et al., 2019). Este artículo aporta a la investigación en un sistema de monitoreo que utiliza el controlador SDN código abierto: OpenDayLight, en la presente tesis se utiliza también dicho controlador SDN.

Usman et al. (2019) aseveró que la infraestructura de tecnologías de la información y comunicaciones (TIC) es vulnerable a diversos fenómenos naturales, esto supone un reto para los operadores que deseen visualizar el estado de los dispositivos TIC. Para hacer frente a estos retos, es esencial contar con una supervisión precisa con visualizaciones interactivas. Por lo tanto, en este artículo, mediante la visualización interactiva de los dispositivos y de sus interacciones se abordó los problemas del mundo real en las plataformas experimentales de emulación habilitada para SDN, denominada como OF@TEIN y OF@KOREN Playgrounds. Haciendo uso del Framework SmartX MultiView Visibility, primero el sistema recopila y valida los datos generados. A continuación, se integra con el sistema de almacenamiento basado en DataLake, que incluye: MongoDB, ElasticSearch y InfluxDB. Con OF@TEIN y OF@KOREN, se presentaron los resultados de la verificación de las implementaciones de los prototipos con las métricas de tiempos de almacenamiento y tiempo de carga de visualización de los componentes web. Este artículo de investigación aportó la integración de bases de datos no relacionales para el desarrollo de los sistemas de monitoreo de redes definidas por software.

Tran et al. (2019) presentó la implementación de un sistema de gestión con tecnologías web para controladores SDN. Esta implementación permite controlar y supervisar rápidamente las funciones de red virtuales (VNF) proporcionadas por las redes SDN. La implementación no sólo aprovechó las ventajas del software de código abierto, como los marcos Flask y Ryu, OpenvSwitch y Mininet, sino que también fue compatible con las funciones de red virtuales. El sistema emuló una topología de red con switches y enlaces sencillos y también actúa como centro de gestión para consultar los comandos de la aplicación de red Ryu mediante la API REST con el fin de supervisar los cambios en el tráfico de la red. Este artículo aportó a la investigación en utilizar tecnologías web para la implementación de un sistema de monitoreo de la red mediante las API REST ofrecidos por los controladores SDN, así también, esta investigación utilizó OpenvSwitch y Mininet.

Lin et al. (2019) publicó un artículo científico, el cual provee un método integrado de monitorización y análisis en SDN VPN (Virtual Private Network). Este método establece un proceso coherente para supervisar el estado de los elementos de red (NE), e integra la información de alarma y salud de los NE para la correlación de alarmas y el análisis del impacto del servicio. El personal de mantenimiento puede supervisar el estado de todos los NE con un único panel. El proceso coherente propuesto consta de varias partes, incluido el mecanismo de recopilación de alarmas y estado de los recursos de los NE desde Resource Orchestrator (RO), el análisis de correlación de alarmas y el análisis del impacto del servicio. Con esta propuesta, podemos conocer la causa de los eventos de fallo en SDN VPN, la naturaleza del fallo y el alcance del impacto del servicio, mejorando la calidad del servicio. Este artículo aportó a la investigación en la generación de alertas para los eventos de fallo en el servicio de red. Además, el artículo presentó el diseño de un sistema de monitoreo

integrado SDN con el motivo de brindar una mejor respuesta a los administradores de red.

Kavitha et al. (2019) consideró que el monitoreo de la red desempeña un papel vital en SDN, ya que la información completa de la red está disponible en un controlador centralizado. Ellos propusieron un Sistema de Monitorización de Red (NMS), que se asienta sobre el controlador SDN OpenDayLight (ODL), lo cual la define como una aplicación northbound, y que envía peticiones periódicas en forma de solicitud HTTP. Esta aplicación enviará varias peticiones hacia el ODL utilizando REST API y procesará la respuesta para cada petición que estará en forma de formato JSON. Cada respuesta será entonces alimentada al módulo individual y los datos serán extraídos, procesados y serán actualizados en consecuencia en las páginas de visualización individuales destinadas a un propósito específico. Este artículo aportó en la utilización de un NMS junto al controlador SDN OpenDayLight. El artículo muestra una arquitectura a alto nivel de la solución, sin embargo, no evalúa la solución ni presenta resultados.

Vela et al. (2018) presentó la herramienta CASTOR MDA que permite recoger grandes cantidades de mediciones de los dispositivos a través del controlador ONOS. Estos datos deben ser preprocesados y analizados para descubrir y resumir el conocimiento que puede ser consumido por los operadores humanos. Por ende, la herramienta permite visualizar la topología, tráfico de red y estadísticas para la localización de fallos a los administradores de la red. Este artículo aportó a la investigación en una arquitectura de un componente software para la presentación de datos de la red. También presentó las interfaces visuales de la herramienta, la cual fue integrada al controlador SDN ONOS.

Finalmente, el estudio de Lange et al. (2018) presentó un aporte importante al utilizar métricas de CPU y memoria para evaluar la eficiencia del monitoreo en redes definidas por software. Al registrar el uso de estos recursos en el controlador ONOS, se pudo medir el impacto del procesamiento adicional de información externa del NMS en la carga computacional. Los resultados indicaron que, si bien la integración del NMS pudo mejorar el rendimiento del balanceo de carga, también generó un aumento en el uso de CPU y Memoria. Este hallazgo resaltó el hecho de que toda herramienta añadida al entorno de red puede tener una pequeña carga adicional en los recursos computacionales en CPU y Memoria. El uso de métricas de CPU y memoria proporcionaron una visión más completa y detallada de cómo el NMS influye en el rendimiento global del sistema, lo que es crucial para entender el impacto real de la integración de herramientas adicionales en una red definida por software.

Marco Teórico

Redes de datos tradicionales

En el contexto de las redes de datos convencionales, resulta fundamental examinar el modelo de referencia OSI y la arquitectura TCP/IP. Ambos conceptos deben ser comprendidos, ya que posibilitan la conectividad entre diversos tipos de redes.

Modelo OSI

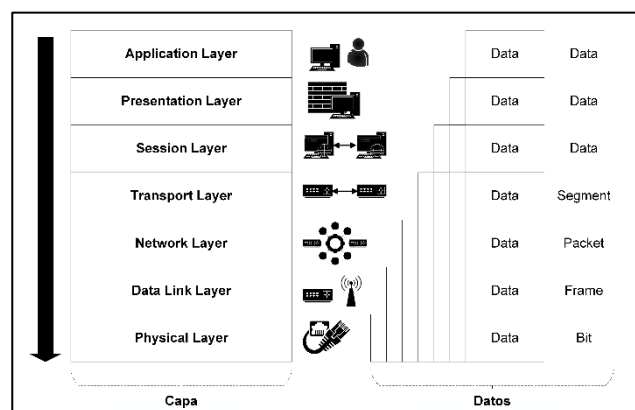
El modelo OSI fue presentado en un standard ISO en 1979, principalmente impulsado por Mike Canepa y su equipo; quienes empezaron a tratar el problema de falta de estandarización a mediados de los años setenta y propusieron las siete capas que actualmente la define. El modelo OSI fue adaptado para cubrir las necesidades crecientes de las redes de computadores (Srivastava et al., 2023).

Las Capas OSI

El modelo OSI tiene siete capas; como se observa en Figura 4, y funciona en una estrategia peer-layer, esto significa que la información de control añadida al PDU (Protocol Data Unit) por una capa es solo útil para la capa de la entidad que recibe la información e insignificante a las otras capas.

Figura 4

Capas del Modelo OSI



La **capa física** (Physical Layer) básicamente maneja datos como bits en bruto. Los protocolos de la capa física dependiente de medio físico y tipo de señal que es llevada en él. La señal puede ser voltaje eléctrico por un cable, una señal de luz a través de una conexión de fibra, o incluso una señal electromagnética en el aire o en el espacio exterior (Panek, 2019).

La **capa de conexión de datos** (Data-Link Layer) proporciona las comunicaciones entre dos nodos conectados a un mismo enlace, la Unidad de Datos de Protocolo (PDU) de esta capa es la *trama*. Estas *tramas* pueden ser de un rango de cientos de bytes a miles de bytes (Panek, 2019).

La **capa de red** (Network Layer) tiene de PDU al *paquete* y se encarga de direccionar los datos de una red a otra y de supervisar las subredes. El proceso de direccionamiento puede ser intrincado, ya que diversos factores pueden influir en la determinación de la ruta óptima para los *paquetes* desde su origen hasta su destino (Panek, 2019).

La **capa de transporte** (Transport Layer) tiene como PDU a los *segmentos*. Existen dos tipos de servicios que hacen comportar a esta capa de diferentes maneras, estos son: orientados a conexión y no orientados a conexión (Panek, 2019).

La **capa de sesión** (Session Layer) maneja los datos de la forma que ellos vienen, sin dividir o concatenar nada. Tiene el propósito de proveer a las entidades de presentación organizar la comunicación para múltiples sesiones que toman lugar en un mismo tiempo (Panek, 2019).

La **capa de presentación** (Presentation Layer) desempeña el papel crucial de determinar cómo los datos son presentados a la aplicación. Su función implica negociar la estructura de los datos a ser enviados con la entidad que recibe el dato, es decir, la sintaxis de la transferencia. Una vez finalizada esta negociación, la capa

puede ofrecer una serie de servicios adicionales, como compresión, encriptación y traducción. Cabe destacar que la elección de los servicios a utilizar recae en la propia aplicación (Panek, 2019).

La **capa de aplicación** es responsable de especificar los parámetros aceptables de calidad de servicio así también características de seguridad, como la autenticación y el control de acceso. Por último, la sincronización de las comunicaciones entre las aplicaciones en servicios orientados a conexión (Panek, 2019).

Arquitectura TCP/IP

TCP/IP es el sucesor del antiguo proyecto ARPANET del Departamento de Defensa de los Estados Unidos (DoD). Esta red fue diseñada para sobrevivir a la caída de los enlaces de tal manera que la comunicación continúe siempre y cuando la fuente y el destino de la conversación existan. Las tecnologías inalámbricas hicieron que sea necesario un nuevo modelo y por ello en 1974 fue desarrollado la arquitectura TCP/IP. En comparación al modelo OSI, que fue creado de tal manera que los protocolos sean desarrollados en base a este, la arquitectura TCP/IP estuvo basada en protocolos ya existentes. Esto da al modelo OSI mayor flexibilidad sobre la arquitectura TCP/IP. Aun así, TCP/IP fue adoptado porque ARPANET ya estaba ahí y ARPANET la adoptó (Srivastava et al., 2023).

Capas de TCP/IP

A diferencia del modelo OSI, TCP/IP consiste en 4 capas (aun cuando también se puede definir 5 capas). En la Figura 5 se puede apreciar que hay sustracciones y agrupaciones de capas a comparación del modelo OSI.

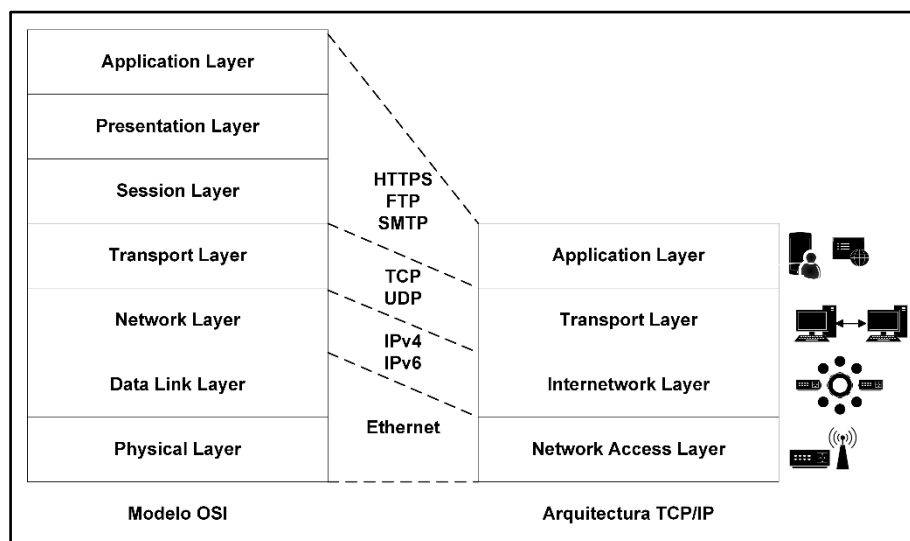
La **capa de acceso a la red** (Network Access Layer) tiene como función principal, asegurar la transferencia de los paquetes originados en la capa de internet

a través de una conexión física al otro extremo del enlace y viceversa. El PDU de esta capa es la trama, entre las más conocidas tenemos de la tecnología *Ethernet* (Srivastava et al., 2023).

La **capa de internet** (Internetwork Layer) tiene como función principal de elegir la ruta más óptima para los datos entre el emisor y destino. El protocolo que lidera esta capa es IP; sin embargo, existen protocolos de soporte como ICMP (Srivastava et al., 2023).

Figura 5

Comparativa de las Capas del Modelo OSI con la Arquitectura TCP/IP



La **capa de transporte** (Transport Layer) tiene similares propósitos que la del modelo OSI. Esta capa está diseñada para dar a la fuente y al destino la habilidad de tener una conversación end-to-end. Son dos protocolos que operan en esta capa, TCP y UDP. Tomando en cuenta que IP no es confiable en la entrega, el uso de TCP provee la confiabilidad para asegurar que los datos lleguen seguros e íntegros. UDP, sin embargo, no posee mecanismo de confiabilidad lo que lo hace mejor para transportar datos más rápido que el primer protocolo. Este último, por ende, es

importante para la transmisión de datos de gestión como lo hace con el protocolo SNMP (Srivastava et al., 2023).

La **capa de aplicación** (Application Layer) es la de más alto nivel en TCP/IP; la cual no tiene capas de sesión ni presentación. Estos no son necesario puesto que la capa de aplicación maneja el control del dialogo, codificación y representación de los datos. Aquí se ubican protocolos de alto nivel como HTTP, HTTPS, telnet, DNS, SMTP y SNMP (Srivastava et al., 2023).

Protocolo IP, TCP y UDP

a. Protocolo IP

El **protocolo de Internet** (IP, *Internet Protocol*) es el más importante protocolo dentro de la capa de internet. IP recibe datos desde un protocolo de la capa de transporte, lo empaqueta en un *datagram*, y entonces lo transporta a y desde un conjunto de nodos. IP es no orientado a conexión, por lo que no establece una línea de comunicación fiable. IP es también responsable por el direccionamiento IP para los nodos de red, usualmente routers. Tomando en cuenta que IP es de no conexión, son las capas superiores responsables para el chequeo de error. Lo que hará IP es desechar un *paquete* y entonces enviar un mensaje ICMP a la dirección IP de origen un mensaje que el *paquete* no llego a donde se suponía (Srivastava et al., 2023).

El protocolo IP provee fragmentación de *paquetes largos* en unos más pequeños. El protocolo trata cada *paquete* como una entidad independiente y no relacionada a otro *paquete* (Brooks et al., 2018).

b. Protocolo TCP

El protocolo TCP está orientado a conexión. Es confiable porque tiene funciones construidas que proveen varios chequeos y balanceos para asegurar la integridad de que la data está siendo transmitida.

TCP es capaz de dividir los datos en segmentos para que los trozos más pequeños se pierdan si hay problemas con la transmisión. TCP también soporta el reensamblaje de *datagram*, asegurando que todos vuelvan al orden del cómo fue enviado (Panek, 2019).

c. Protocolo UDP

El protocolo UDP fue diseñado para proveer a las aplicaciones, la habilidad de enviar datos con un mínimo número de bytes en el tamaño de cabecera. UDP es definido como ‘orientado a transacción’, y protección en la entrega y la duplicación no están garantizadas (Srivastava et al., 2023).

Es por este sentido que UDP no está orientado a conexión. A su vez, el carecer de los mecanismos de mantenimiento de TCP lo hace ligero y una elegible opción para las aplicaciones que necesitan la transferencia en “tiempo real” como VoIP, videoconferencia, y streaming de video y audio. También por ser no orientado a conexión, puede soportar *broadcasting* (enviar mensajes a todos los nodos dentro un dominio *broadcast*) y *multicasting* (enviar mensajes a todos los nodos que están suscriptos) (Srivastava et al., 2023).

Dado el hecho que los segmentos UDP pueden ser perdidos a lo largo del camino y ellos pueden ser recibidos fuera de la secuencia, entonces las aplicaciones tendrán que manejar esta tarea por sí misma. UDP usa también un *checksum*, el cual es un método para detectar errores de transmisión (Srivastava et al., 2023). Una utilidad de UDP es para la gestión de redes. UDP es utilizado por SNMP para enviar los mensajes necesarios entre el agente SNMP y el gestor SNMP por sus características antes mencionadas (Panek, 2019).

Gestión en redes IP tradicionales

Las redes y los sistemas distribuidos son cada vez más importantes y críticos en la actualidad. La tendencia es avanzar hacia redes más sofisticadas que puedan soportar un mayor número de aplicaciones y usuarios. La gestión de redes implica llevar a cabo operaciones como el monitoreo de dispositivos, la gestión del enrutamiento y la gestión de la seguridad, con el objetivo de asegurar un buen rendimiento de la red, como una baja latencia, un bajo consumo de energía y una baja pérdida de paquetes, entre otros (Aboubakar et al., 2022).

SNMP

SNMP (SNMP, *Simple Network Management Protocol*) es un protocolo de red desarrollado por el IETF (Internet Engineering Task Force) que permite el monitoreo remoto de dispositivos sobre redes TCP/IP. Ofrece una variedad de operaciones, como el monitoreo, la reconfiguración de parámetros de dispositivos de red. SNMP involucra los tres elementos mencionados anteriormente para la gestión de dispositivos de red: agentes, nodos y gestor. Este protocolo se basa en la Estructura de Información de Gestión (SMI) y en la Base de Información de Gestión (MIB). La MIB es una base de datos utilizada para gestionar los dispositivos de red, mientras que la SMI define la estructura y los tipos de objetos almacenados en la MIB (Aboubakar et al., 2022).

Conceptos básicos

El enfoque de gestión de red utilizado en SNMP comprende los siguientes componentes fundamentales:

a. Gestor o entidad gestora

Actúa como un medio de comunicación que conecta el gestor de la red (humano) con el sistema de gestión de red. El **gestor** es el puesto central de la gestión y su función es de recopilar, procesar, analizar y/o visualizar la información de gestión.

Aquí es donde se realizan las acciones para controlar el funcionamiento de la red, y donde el administrador de la red interactúa con los dispositivos de red (Aboubakar et al., 2022).

b. Agente o agente de gestión

Los dispositivos esenciales, como computadoras, routers y switches, pueden estar equipados con software de agente. Este software actúa como un proceso residente que se ejecuta en cada **dispositivo gestionado**, permitiendo que sean controlados desde el **gestor**. El agente responde a las solicitudes de información y ejecuta acciones realizadas desde el **gestor**, además puede proporcionar información no solicitada de manera asíncrona al **gestor** (Aboubakar et al., 2022).

c. Protocolo de gestión de red

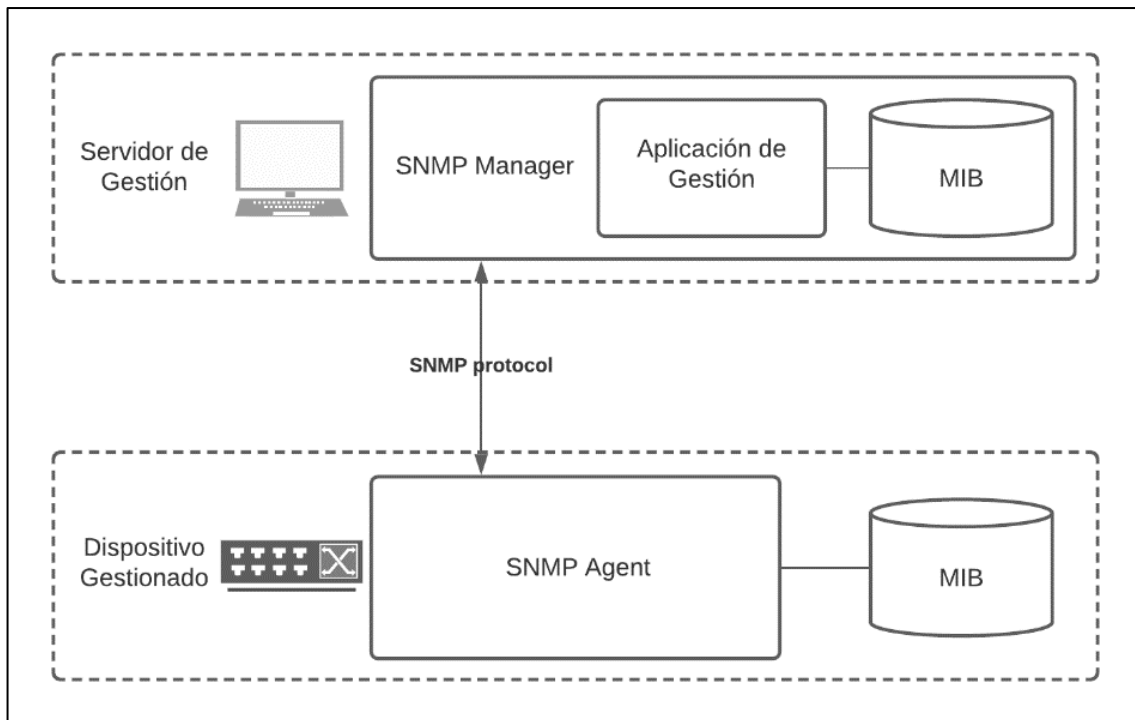
De acuerdo con Aboubakar et al. (2022), existe una conexión entre el **gestor** y los **agentes** a través de un **protocolo de gestión de red**, usualmente, el Protocolo Simple de Gestión de Red (SNMP). Las operaciones dentro del protocolo SNMP están estrechamente vinculadas a la representación de los datos de gestión en el formato MIB (siglas en inglés de Management Information Base, discutido en el punto d). Los datos de gestión en MIB se representan como un árbol jerárquico de información, donde cada punto tiene un identificador único.

d. Base de datos de información de gestión

Según lo expuesto por Espinel Villalobos et al. (2021), para administrar eficazmente los recursos de la red, se modela cada recurso como un objeto, lo que refleja un componente del agente bajo gestión. Estos objetos son organizados en una base de datos de información de administración (MIB, *Management Information Base*). La MIB funge como un conjunto de puntos de acceso que el administrador utiliza en la interacción con el agente, como se grafica en la Figura 6.

Figura 6

Comunicación simple entre un Gestor y un agente, y sus MIB



El gestor desempeña la función de supervisión al acceder a los valores de los objetos MIB. Además, tiene la capacidad de realizar acciones o cambiar la configuración de un agente al modificar los valores de los objetos. Cada objeto de gestión tiene un identificador de objeto el cuál es un nombre único estructurado jerárquicamente. El identificador de objeto; también llamado OID, en SNMP tiene una estructura de enteros separados por puntos (Por ejemplo, un identificador podría lucir como 1.3.6.1.2.1.6.5). De acuerdo a Aboubakar et al. (2022), la definición de cualquier objeto de gestión está hecha por medio de reglas llamada Estructura de Información de Gestión (SMI, *Structure of Management Information*), la cual provee la guía de uso de las cuáles nuevos MIB pueden ser definidos, como es mostrado en las Figura 7.

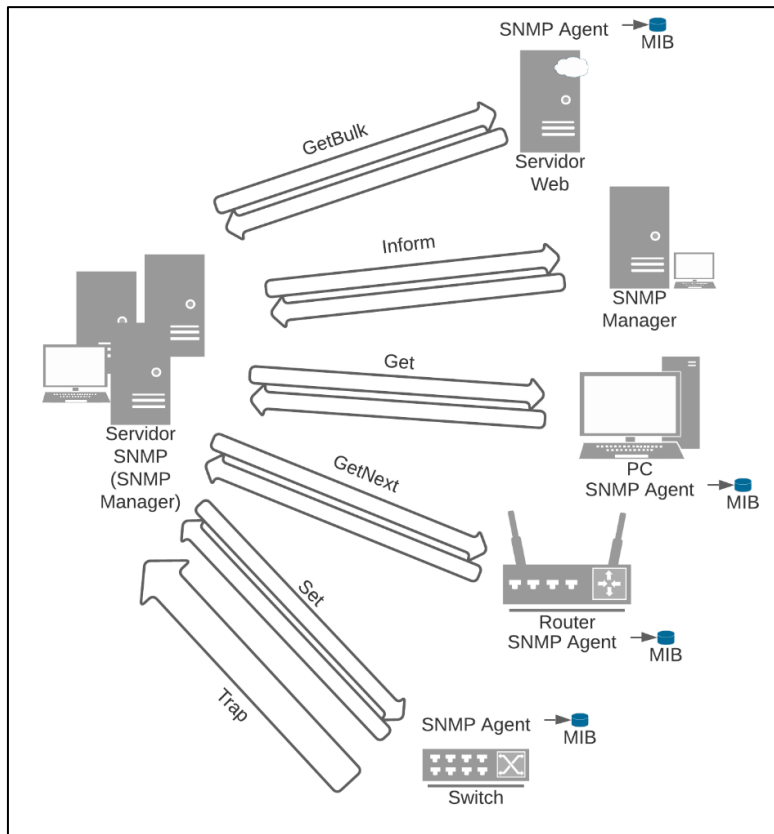
En la Figura 8 muestra la interacción de los dispositivos gestionados con el servidor gestor en el modelo SNMP de redes tradicionales.

Figura 7*Estructura de Información de Gestión OBJECT-TYPE*

```

nombreDelObjeto OBJECT-TYPE
    SYNTAX  sintáxis 
    ACCESS  acceso máximo 
    STATUS  estado 
    DESCRIPTION
         "Breve descripción" 
    ::= {  número del grupo  }

```

Figura 8*Modelo SNMP***CMIP**

CMIP (*Common Management Information Protocol*) es un protocolo de red encargado de la comunicación entre el gestor de red y los dispositivos gestionados. CMIP permite la gestión de fallos, la gestión de seguridad, la monitorización de

rendimiento, entre otras. CMIP fue diseñado para ser utilizado en el modelo de referencia OSI y amplía las capacidades de SNMP. Sin embargo, CMIP no ha sido ampliamente adoptado debido a la lentitud en el proceso de estandarización (Aboubakar et al., 2022).

Dicho anteriormente, CMIP es orientado a conexión, control de flujos, sincronización y recuperación. En contraste SNMP es no orientado a conexión por el uso del protocolo UDP. Como conclusión a los modelos de gestión de red revisados en la literatura podemos resumirlos en la Tabla 2:

Tabla 2

Tabla comparativa de Modelos de Redes Tradicionales y sus respectivos Protocolos de Gestión

Modelo/Arquitectura	OSI	TCP/IP
Protocolo de gestión de red	CMIP	SNMP
Nombre de base de datos	MIL	MIB
Funciones	Complejas	Simples para SNMPv1, se añaden más sofisticadas con la versión 2

Redes definidas por software

La irrupción de la computación en la nube y otras tecnologías novedosas ha presentado una serie de nuevos paradigmas en el ámbito de las redes, con el propósito de simplificar la administración de estas y ofrecer innovación mediante la programación de la infraestructura de red.

Las Redes Definidas por Software (SDN, por sus siglas en inglés) se refieren a un nuevo enfoque que busca la programabilidad de la red. Esto implica la facultad de inicializar, monitorear, controlar, modificar y administrar dinámicamente el comportamiento de la red a través de interfaces abiertas. SDN destaca el papel del software en el funcionamiento de las redes al introducir una abstracción para el plano

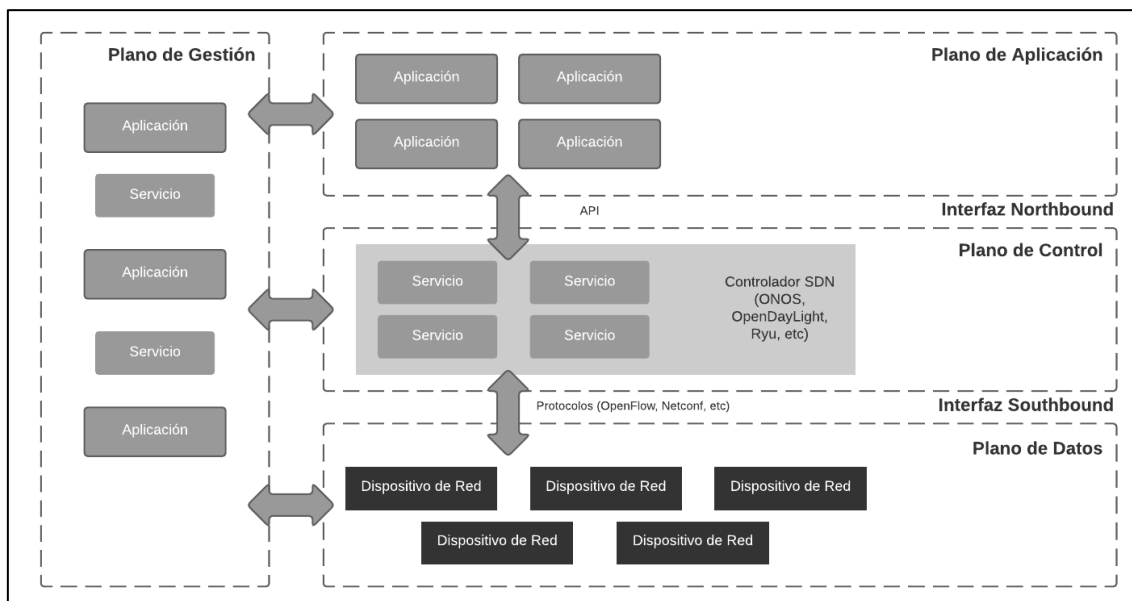
de datos, separándolo del plano de control. Esta separación permite acelerar los ciclos de innovación en ambos planos (Jain et al., 2019).

Capas y Arquitectura de SDN

La Figura 9 presenta la arquitectura de SDN en un esquema detallado y de alto nivel.

Figura 9

Arquitectura SDN



Nota. Adaptado de *Big Data and Software Defined Networks* (p. 15), por J. Taheri, 2018, Institution of Engineering and Technology.

SDN abarca múltiples planos (Khan et al., 2023):

Plano de Datos: El plano de datos es responsable de manejar los paquetes en función de las instrucciones recibidas del plano de control. Estas acciones pueden incluir reenviar, descartar o modificar paquetes, y generalmente representan el punto final de los servicios y aplicaciones. Además, se encarga de gestionar el estado operativo del dispositivo de red, como su disponibilidad, el número y estado de los puertos, entre otros aspectos. Esta información resulta útil para los servicios y aplicaciones del plano de gestión.

Plano de Control: El plano de control se encarga de tomar decisiones sobre cómo deben ser reenviados los paquetes a través de los dispositivos de red, y de transmitir esas decisiones a dichos dispositivos para su ejecución. La tarea principal del plano de control es ajustar las tablas de flujo, también conocidas como tablas de reenvío, que residen en el plano de datos. Estos ajustes se basan en la topología de la red.

Plano de Gestión o Aplicación: El plano de gestión se responsabiliza de supervisar, configurar y mantener los dispositivos de red, tomando decisiones sobre el estado de un dispositivo de red, así como configurando todas o parte de las reglas de reenvío. El plano de aplicación es el espacio en el que residen las aplicaciones que definen las políticas de la red. Es posible que las aplicaciones se implementen de manera modular y distribuida, lo que implica que pueden abarcar varios planos en la Figura 9. En la literatura reciente el plano de gestión y aplicación se utilizan indistintamente para referirse a las aplicaciones software que interactúan con los otros planos.

Sistemas operativos de red

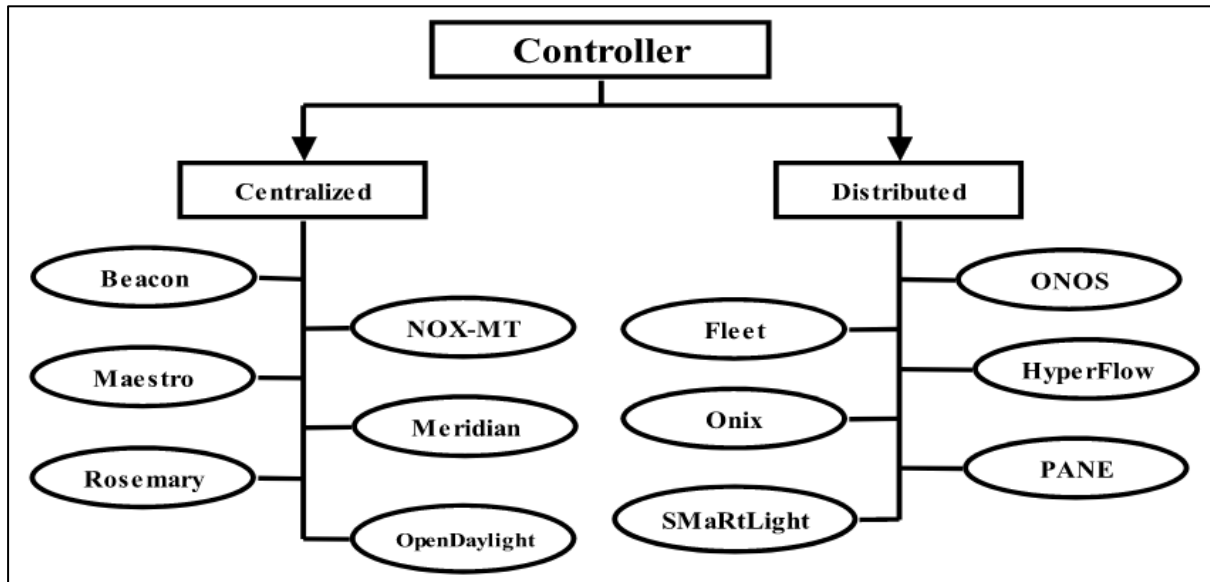
SDN ha transformado radicalmente las redes, separando el control de la red del tráfico de datos, lo que ha introducido flexibilidad y capacidad de programación en las redes. Dentro de esta nueva arquitectura, una entidad fundamental emerge como uno de los componentes más importantes: el Sistema Operativo de Red, denominado controlador SDN. Esta entidad juega un papel crucial al permitir el establecimiento de políticas y reglas en la red (Paliwal et al., 2018).

Los controladores SDN han sido desarrollados con diferentes arquitecturas y enfoques. Desde una perspectiva arquitectónica se podría clasificar en **centralizado** y **distribuido** como detallado en la Figura 10, dentro de este último se encuentran

diferentes enfoques que aprovechan la distribución física, el más común es el lógicamente centralizado.

Figura 10

Diagrama de perspectiva arquitectónica de los controladores



Nota. Recuperado de “Controllers in SDN: A Review Report” (p. 5), por Paliwal et al., 2018, *IEEE Access*.

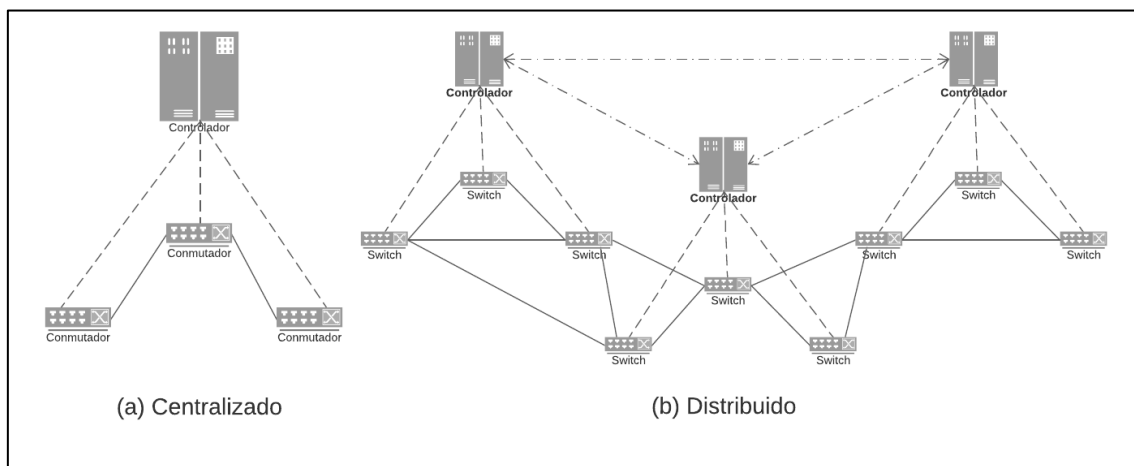
Centralizado: La gestión de arquitectura centralizada es hecha por un único controlador responsable de la distribución y carga de trabajo y rutas de reenvío. Esto hace más fácil su implementación, sin embargo, sufre de problemas de escalabilidad, tolerancia a fallas, interoperabilidad y confiabilidad en grandes redes. Uno de sus más conocidos exponentes es el controlador Ryu.

- **Ryu:** Es un componente que ofrece una interfaz de programación de aplicaciones (API) claramente definida, lo que facilita a los desarrolladores crear aplicaciones de control y gestión de redes. Ryu es compatible con varios protocolos de gestión de dispositivos, como OpenFlow, Netconf, OF-config, entre otros (*Ryu SDN Framework, s/f*).

Distribuido: A fin de satisfacer los requerimientos de escalabilidad y confiabilidad, y dejar atrás las limitaciones de una arquitectura centralizada, diferentes tipos de enfoques han sido propuestos dentro de una arquitectura distribuida, como detallado en Figura 11.

Figura 11

Enfoques arquitectónicos de los controladores



Dentro de esta arquitectura, podemos encontrar a los populares controladores ONOS y OpenDayLight, los cuales siguen un enfoque lógicamente centralizado y físicamente distribuido.

- **ONOS:** Open Network Operating System (ONOS) es un controlador extensible, modular y distribuido. ONOS provee gestión de componentes de red, con el fin de proveer servicios de comunicación a los host finales y redes vecinas (ONOS, 2020).
- **OpenDayLight:** Es un controlador modular, extensible, escalable, altamente disponible y multiprotocolo construido para desarrollo en SDN en redes heterogéneas y de varios fabricantes. Este controlador provee una plataforma de abstracción de servicios que permite a los desarrolladores escribir

aplicaciones que funcionan fácilmente entre una variedad de hardware y protocolos de interfaz southbound (OpenDayLight, 2021).

Ambos controladores están escritos en el lenguaje de programación JAVA y están cargados dentro de contenedores Karaf OSGi. OSGi es un componente de sistema para JAVA que permite a los módulos ser instalados y ejecutados dinámicamente en una JVM (Java Virtual Machine). Lo que hace que se pueda ejecutar en varios sistemas operativos.

Comparación entre sistemas operativos de red

La Tabla 3 detalla la comparación de estos sistemas operativos de red:

Tabla 3

Comparativa de controladores

Controlador	Arquitectura	Interfaz NorthBound	Lenguaje	Versiones de OpenFlow
Ryu	Centralizada multi-hilos	Ad-hoc API	Python	1.0-1.5
ONOS	Distribuida	Restful API	Java	1.0
OpenDayLight	Distribuida	REST, RESTCONF, Java API	Java	1.0-1.3

Nota. Adaptado de "Controllers in SDN: A Review Report" (p. 4) por Paliwal et al., 2018, *IEEE Access*.

Gestión de redes definidas por software

Los métodos de gestión de la red tradicionales no responden a los cambios vertiginosos y continuos eventos de las redes actuales. La configuración de las redes de datos se está volviendo cada vez más molesta, ya que los operadores de redes deben realizar tareas de gestión de redes cada vez más sofisticadas (Binsahaq et al., 2019). Las dos principales razones son:

- Continuos cambios en el estado de la red
- Configuración individual de los dispositivos de red a bajo nivel.

Cómo nuevo paradigma, SDN intenta resolver estos problemas y mejorar la gestión de la red. Algunos de sus beneficios son (Jain et al., 2019):

- Separación del plano de datos y el plano de control: En SDN, se implementa una separación entre ambos planos, lo cual implica que el plano de datos, encargado de reenviar los paquetes de la red, se separa del plano de control, donde se encuentra la lógica de control de los paquetes. Esta separación tiene como objetivo lograr una arquitectura más sencilla y eficiente.
- Centralización lógica del control: La centralización lógica del control implica que la inteligencia se traslada a un controlador SDN centralizado o a un Sistema Operativo de Red (NOS). Esto permite tener una visión global de la red, sobre la cual se pueden crear vistas abstractas de la red física mediante una capa de virtualización. Estas vistas abstractas son utilizadas por diferentes programas de control desarrollados por los desarrolladores de software.
- Control basado en flujos: El control basado en flujos se refiere a que las reglas de reenvío en una red se basan en las entradas de flujo almacenadas en tablas dentro del switch. Cada entrada en la tabla representa un flujo identificado por sus campos de coincidencia. Si una entrada coincide con las entradas de la tabla de flujos, se lleva a cabo una acción específica con el paquete y se incrementa un contador. En caso de que no haya coincidencia con las entradas de la tabla, el paquete se envía al controlador SDN para su procesamiento.
- Programabilidad: La programabilidad en SDN permite el desarrollo de APIs sobre el controlador SDN para crear aplicaciones y servicios de red, como firewall, equilibrador de carga e ingeniería de tráfico. SDN está transformando las redes en una disciplina de software, introduciendo un modelo de abstracción fundamental que antes no existía en el paradigma de las redes. Esto se alinea con la evolución de la informática, donde las abstracciones han

demostrado un crecimiento significativo en áreas como los sistemas operativos y las estructuras de datos.

La demarcación entre el plano de control y de gestión es un problema que surge en la implementación de SDN ya que algunas de las operaciones de gestión necesitan ser gestionadas en “tiempo real” o “casi tiempo real”, es por ello que una ambigua definición puede ocasionar un deficiente funcionamiento de la red.

Cloud computing

No hay duda de que las tecnologías **Cloud** están ayudando a cambiar cada negocio en cada industria. Los atributos claves de esta tecnología han permitido a los emprendedores competir con industrias bien establecidas. Así mismo, **Cloud** ha posibilitado que estos últimos añadan nuevas capacidades y transformar los procesos de negocio a la velocidad del cambio. Aunque el principal motivo de la adopción de **Cloud** es la reducción de costos y el cambio de gastos de tecnologías de un gasto de capital a un gasto operativo, las compañías están ahora apostando por los servicios **Cloud** en la transformación de sus respectivos negocios (Hurwitz y Kirsch, 2020).

Conceptos de Cloud

Cloud Computing: es un método para proveer recursos computacionales compartidos, incluyendo aplicaciones, almacenamiento, redes, desarrollo y plataformas de implementación, así como procesos negocios. **Cloud Computing** hace que los recursos computacionales más fáciles de usar mediante la estandarización y automatización (Hurwitz y Kirsch, 2020).

Estandarización: Es la implementación de servicios usando un enfoque consistente apoyado por una serie de interfaces consistentes. Igualmente, **Cloud** generalmente requiere que los procesos sean implementados a través del uso de la automatización (Hurwitz y Kirsch, 2020).

Automatización: Es un proceso que se activa en función de las reglas de negocio, disponibilidad de recursos y exigencias de seguridad. La automatización es necesaria para apoyar un modelo de aprovisionamiento de autoservicios, y promover la eficiencia garantizando que un servicio prestado ya no es necesario, se devuelva a la reserva de recursos (Hurwitz y Kirsch, 2020).

La Nube Pública

La **nube pública** es un conjunto de recursos como hardware, redes, almacenamiento, servicios, aplicaciones e interfaces que son propiedad y operados por un proveedor externo. Estos recursos están disponibles para su uso por parte de otras compañías o individuos. Las **nubes públicas** son viables porque ofrecen diferentes opciones de computación, almacenamiento, y muchos otros interesantes servicios. Con varios de estas opciones siempre disponibles, los clientes pueden rápidamente seleccionar, optimizar y usar los servicios que necesiten las aplicaciones que se ejecutarán en la **nube pública**. Todos estos servicios en la nube están disponibles bajo demanda (**on-demand**) (Hurwitz y Kirsch, 2020).

La nube Privada

La **nube privada** se refiere a un conjunto de recursos que incluye hardware, redes, almacenamiento, servicios, aplicaciones e interfaces, los cuales son propiedad y operados por una organización para su uso interno por parte de empleados, partes interesadas o clientes. A diferencia de la nube pública, una **nube privada** está diseñada para ser utilizada exclusivamente por una empresa específica y puede ser creada y administrada por la **propia organización** o por un **proveedor externo**. La **nube privada** se caracteriza por ser un entorno altamente controlado, no accesible al público en general. Está protegida por un firewall y se enfoca en la automatización de los servicios de TI para cumplir con las necesidades específicas de la empresa. Esto

permite establecer reglas y procesos de negocio dentro del software, lo que brinda mayor previsibilidad y capacidad de gestión en los entornos de la nube privada (Hurwitz y Kirsch, 2020).

Herramientas de simulación y emulación

Para la investigación es indispensable estudiar las herramientas que facilitan el análisis y evaluación de los sistemas de monitoreo de red a través de las características para la simulación y emulación. Para ello, los siguientes conceptos son definidos:

Simulación: Esta técnica de experimentación está basada en la creación de modelos lógicos de los sistemas de tal manera que ejecutar estos modelos en un entorno computacional simule el comportamiento del sistema. Usar el poder computacional para la ejecución de los modelos hace posible evaluar sistemas más complejos a más detalle, incluyendo dispositivos de red, aplicaciones y protocolos (Fitzek et al., 2020).

Emulación: Es una técnica híbrida entre la simulación de experimentos y las pruebas reales. La idea principal es reproducir en tiempo real y de manera controlada las funcionalidades esenciales de un sistema, así que pueda interactuar con otros sistemas reales que puedan, por ende, ser evaluados (Fitzek et al., 2020).

GNS3

Es un emulador gráfico multiplataforma disponible en Windows, OS X y Linux. La arquitectura GNS3 consiste de dos componentes de software (GNS3, s/f):

- El software GNS3-all-in-one (GUI)
- La máquina virtual de GNS3 (VM)

Opciones de Servidor:

Cuando se utiliza la GUI para crear topologías en GNS3, los elementos de la red generados necesitan ser ejecutados por un proceso de servidor. Para ello tenemos estas opciones:

- GNS3 servidor local
- GNS3 VM local
- GNS3 VM remoto

El servidor GNS3 local se ejecuta en la misma PC donde está instalada la GUI. También está la opción recomendada como GNS3 Virtual Machine, la cual se ejecuta localmente usando software de virtualización (Vmware Workstation, Virtualbox o Hyper-V). Sin embargo, también puedes ejecutar GNS3 Virtual Machine remotamente en un servidor usando la **nube**.

GNS3 soporta dispositivos emulados y simulados.

Emulación: GNS3 emula el hardware de un dispositivo y puede ejecutar imágenes actuales (por ejemplo, IOS de Cisco) en dispositivos virtuales.

Simulación: GNS3 imita las características y funcionalidades de un dispositivo como un switch (por ejemplo, Dentro de GNS3 se encuentra un built-in L2 switch que simula el dispositivo en lugar de ejecutar el sistema operativo IOS).

Ventajas:

- Software gratuito
- Software de código abierto.
- No hay limitaciones de número de dispositivos soportados.
- Soporta entornos de múltiples proveedores (Cisco, Huawei, Juniper, etc)

Mininet

Mininet es un sistema de emulación de redes que permite ejecutar una colección de hosts, switches, y enlaces en Linux. A través de una virtualización ligera,

Mininet crea la apariencia de una red en un único sistema. En Mininet, los host actúan como un dispositivo terminal accesible mediante SSH y que puede ejecutar programas como si estuviera en un dispositivo terminal físico. Los paquetes son procesados por componentes que simulan conmutadores Ethernet, enrutadores o cajas intermedias, y se pueden utilizar varias colas para gestionar su flujo (Mininet, s/f).

En resumen, Mininet permite la creación de hosts virtuales, switches, enlaces y controladores que, aunque se implementan utilizando software en lugar de hardware físico, se comportan de manera similar a los elementos de hardware reales. Esto permite emular y simular una red completa en un entorno virtual (Mininet, s/f). De acuerdo a (Mininet, s/f) las ventajas son las siguientes:

- Es rápido, empezar una simple red en minutos.
- Crea topologías personalizadas
- Es de código abierto.
- Puedes ejecutar programas reales a través de servidores web.

SCRUM

Según Layton (2022) Scrum se define como un marco de trabajo que permite a las personas abordar problemas complejos de adaptación mientras entregan productos de alto valor de manera productiva y creativa. Scrum no es una herramienta definitiva, sin embargo, es ampliamente utilizado por la industria y la academia debido a la posibilidad de integrar otras herramientas. De acuerdo a Ilyés (2019) Scrum puede adaptarse correctamente a las necesidades cambiantes de las investigaciones científicas. Scrum se aplica adecuadamente a organizaciones de investigación y desarrollo tecnológico. Los conceptos claves del marco de trabajo son los siguientes:

El equipo de scrum

El marco de trabajo Scrum se caracteriza por su enfoque en la entrega de productos de manera progresiva e incremental, lo que optimiza las oportunidades de recibir retroalimentación. A través de entregas incrementales, se garantiza la disponibilidad constante de una versión potencialmente útil del producto funcional. Un equipo Scrum está compuesto por tres roles fundamentales: el Propietario del Producto (Product Owner), el equipo de desarrollo (Development Team) y un Scrum Master. Estos roles trabajan en conjunto para impulsar la colaboración, la comunicación y la eficiencia en el proceso de desarrollo del producto (Layton, 2022).

Propietario del producto (Product Owner): Este rol asume la responsabilidad de administrar la cartera de productos y garantizar el éxito del producto. Puede llevar a cabo el trabajo directamente o delegarlo al equipo de desarrollo, pero siempre mantiene la responsabilidad última sobre el producto (Layton, 2022).

Equipo de desarrollo (Development Team): Conformado por especialistas que colaboran para producir un Incremento de producto que podría entregarse al final de cada Sprint. Son autoorganizados y multidisciplinarios, asumiendo la responsabilidad colectiva de completar las tareas necesarias para alcanzar los objetivos establecidos (Layton, 2022).

Scrum Master: Encargado de fomentar y respaldar la correcta aplicación de Scrum. Su papel central consiste en garantizar que el equipo internalice y aplique los fundamentos y enfoques de Scrum. El Scrum Master actúa como facilitador, eliminando obstáculos y fomentando un entorno de trabajo colaborativo y de mejora continua (Layton, 2022)

Eventos de scrum

Según Layton (2022), en Scrum se utilizan eventos para establecer una cadencia regular y minimizar la necesidad de reuniones adicionales fuera del marco

de trabajo. Todos los eventos en Scrum tienen una duración definida y acotada en el tiempo. Por ejemplo, el Sprint tiene una duración fija y no puede ser acortado ni extendido. Estas restricciones de tiempo proporcionan un marco claro para la planificación y ejecución de cada evento en Scrum.

Sprint: Dentro del marco de Scrum, un Sprint es un intervalo temporal, que usualmente no excede un mes, destinado a la creación de un Incremento funcional y posiblemente desplegable del producto. Los Sprints mantienen una duración uniforme a lo largo del proyecto, y un nuevo Sprint se inicia inmediatamente después de que concluye el anterior. También se incluye un Sprint 0 al inicio del proyecto, donde se definen los EPIC, las Features y las Historias de Usuario (Layton, 2022).

Planificación de Sprint (Sprint Planning): En este evento, que no excede las ocho horas en un Sprint de un mes, se establecen las Historias de Usuario y las actividades que se abordarán durante el periodo del Sprint (Layton, 2022).

Scrum diario (Daily Scrum): Es un evento diario de 15 minutos destinado al Equipo de Desarrollo. Durante el Scrum diario, se planifica el trabajo para las próximas 24 horas, se revisa el progreso y se identifican posibles obstáculos (Layton, 2022).

Revisión de Sprint (Sprint Review): Después de la conclusión del Sprint, se realiza una Evaluación de Sprint para examinar el Incremento de producto construido. Durante esta evaluación, tanto el Equipo Scrum como los stakeholders comparten detalles sobre los logros alcanzados durante el Sprint (Layton, 2022).

Retrospectiva de Sprint (Sprint Retrospective): Es una reunión que tiene lugar después de la Evaluación de Sprint y previo a la subsiguiente Planificación de Sprint. Durante esta Retrospectiva de Sprint, el Equipo Scrum lleva a cabo una

autoevaluación y elabora un plan para incorporar mejoras en el próximo Sprint. Para Sprints de un mes, esta reunión no excede las tres horas en duración (Layton, 2022).

Elementos de scrum

Los elementos de Scrum son importantes para mantener un consenso sobre los entregables que se entregan y como se comunica los requerimientos de los productos. Los elementos utilizados en el presente trabajo son los siguientes:

EPIC: Las épicas (EPIC) se describen en la etapa inicial del proyecto, son funcionalidades de alto nivel o descripciones de productos sin refinar (Layton, 2022).

Característica: La característica (Feature) es un elemento que componen a un EPIC, permite describir una funcionalidad de alto nivel jerárquicamente menor al EPIC. (KathrynEE, 2021).

Historia de usuario: La historia de usuario (HU) es una estructura predefinida y sencilla utilizada para documentar los requerimientos y funcionalidades deseadas por el usuario final. La HU responde a preguntas como quién la quiere, qué se quiere y para qué se quiere (Layton, 2022).

Tareas: Las tareas son actividades relacionadas con las historias de usuario y se enumeran como compromisos para ser realizados durante un evento Scrum. Representan el nivel mínimo de detalle que describe las acciones técnicas necesarias para cumplir con las historias de usuario (Layton, 2022).

Scrumboard: El Scrumboard es una herramienta para realizar el seguimiento de las tareas dentro de un Sprint. Se divide en tres secciones: Por hacer (To Do), En progreso (In Progress) y Hecho (Done). Estas secciones agrupan las tareas según su estado de avance (Layton, 2022).

Sprint Burndown Chart: El Sprint Burndown Chart representa de manera visual la cantidad de tareas restantes en el Sprint en curso. Su objetivo es visualizar el

progreso del trabajo realizado a lo largo del Sprint, mostrando una línea que indica la cantidad de trabajo ideal a medida que avanza el tiempo (Layton, 2022).

Sistema de monitoreo de red

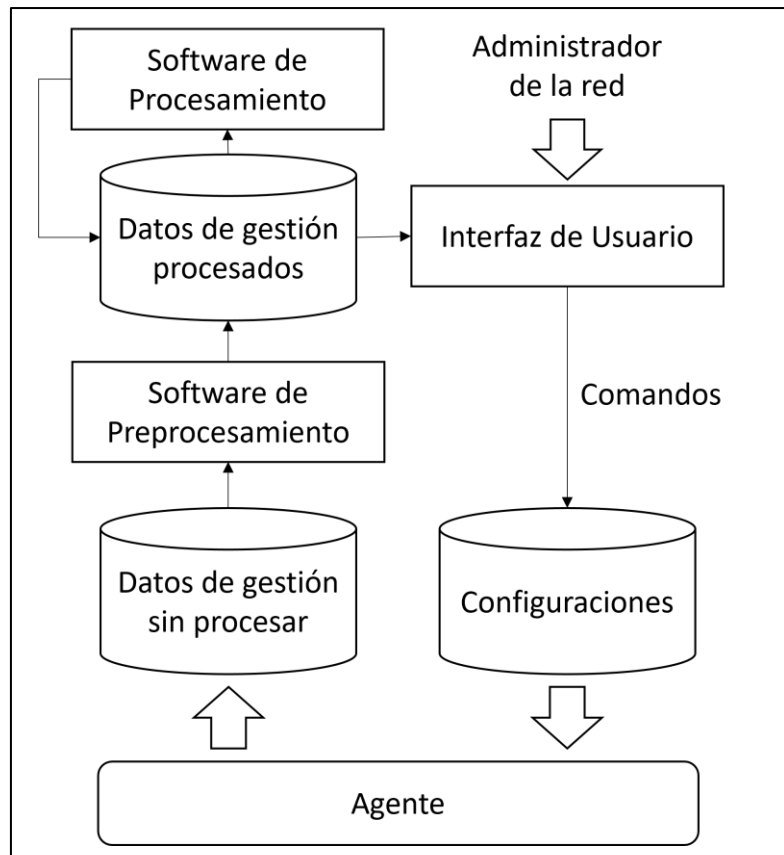
Sistemas de gestión de redes legadas

Las aplicaciones de gestión son software utilizados por los administradores de red para ayudar en sus tareas. Dentro de estas aplicaciones se encuentra el sistema de monitoreo de red, que representa una aplicación particular dentro del conjunto de sistemas de gestión de red. El sistema de gestión de red (Network Management System) es un conjunto de software que se ejecuta en los centros de operaciones de red (NOC, Network Operations Center) (Espinel Villalobos et al., 2021).

Un sistema de gestión de red común se muestra en Figura 12. Los datos recolectados de diferentes dispositivos agentes son guardados en una base de datos sin procesar. Entonces estos datos son procesados por un componente de software para guardar los datos en un base de datos esquematizada. Después del procesamiento, los datos son visualizados al administrador en los NOC. Los datos visualizados pueden proveer la información del estado de salud de los sistemas computacionales, identificar cualquier área de atención y señalar cualquier fallo en el sistema (Espinel Villalobos et al., 2021). Los administradores finalmente pueden elegir configurar o modificar el estado de los sistemas mediante la interfaz de usuario.

Figura 12

Estructura tradicional de un Sistema de Gestión de Red-NMS



Los sistemas de monitoreo son sistemas especializados en recolectar y monitorear eventos de la red. Estos sistemas procesan enormes volúmenes de eventos y alarmas de la red para finalmente mostrarlas al administrador de red u otras aplicaciones. También pueden ofrecer servicios adicionales para la síntesis de los eventos y alarmas en mensajes de alto nivel (Espinel Villalobos et al., 2021).

Monitoreo pasivo

En la supervisión pasiva, el sistema de gestión recoge la información que está disponible en el funcionamiento regular del sistema informático sin introducir ninguna carga de trabajo adicional que haga que el sistema informático realice un trabajo adicional (Tsai et al., 2018).

Monitoreo activo

A diferencia de la monitorización pasiva, la monitorización activa requiere realizar peticiones de medición explícitas para determinar la información disponible en la red. Se utilizan diferentes tipos de técnicas para supervisar varios tipos de información que está disponible en las diferentes fuentes. La monitorización activa impone una carga adicional a la red, pero puede proporcionar información dirigida que puede ser difícil de obtener sólo con la monitorización pasiva (Tsai et al., 2018).

La gestión de la red hacia el año 2030

Las redes hacia el año 2030 lo más probables es que continúe confiando en la “softwarización” de la red con tecnología como SDN y Virtualización de funciones de Red (NFV, *Network Function Virtualization*). El despliegue de aplicaciones que aprovechen estas tecnologías como las basadas en intenciones requerirá del mantenimiento de la red. Las redes y el rendimiento de los servicios deben ser constantemente monitoreados para asegurar que los requerimientos de servicios sean satisfechos. Esto implicaría que los sistemas de gestión dinámicamente se ajusten a la asignación de recursos y configuración de la red. Es así que adquiere gran relevancia el desarrollo de sistemas que monitoreen las redes con estas tecnologías (Clemm et al., 2020).

Sistema de monitoreo de red-NMS en redes definidas por software

Un sistema de monitoreo de red (NMS, *Network Monitoring System*) puede encontrar defectos en la red y ofrecer una rápida y precisa vista de la red para asegurar y optimizar las operaciones de red. La gestión de redes se espera también verse simplificada en la medida que los NMS's puedan interactuar con los controladores SDN en lugar de los múltiples elementos de red (Ndiaye et al., 2020).

El límite funcional del sistema de gestión de redes se basa en los componentes rudimentarios de la supervisión de fallos, la supervisión del rendimiento y el

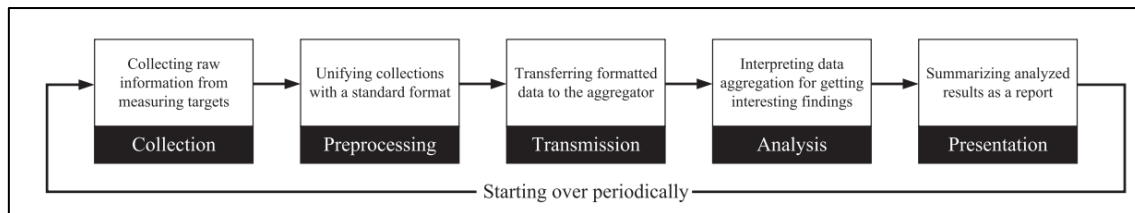
aprovisionamiento o la configuración de una red. Los sistemas de gestión de redes se guían por objetivos que deben cumplirse, como el mantenimiento del rendimiento de la red, el mantenimiento de la calidad del servicio, la fiabilidad, etc. Dentro del paradigma SDN, el plano de gestión es dónde el administrador de la red configura y monitorea las operaciones de la red mediante el uso de NMS's (Tran et al., 2019). Así las aplicaciones SDN se ubican; en el paradigma de SDN, encima del controlador SDN, la cual ofrece funciones de red de alto nivel.

En el presente informe de tesis, una aplicación SDN o aplicación de red es un NMS ubicado en el plano de gestión/aplicación, al cual se integra al plano de aplicación para ofrecer funciones extras para las aplicaciones de negocio.

Proceso de monitoreo de redes definidas por software

“El monitoreo es un concepto importante en la gestión de redes ya que ayuda a los operadores de red a determinar el comportamiento de la red y el estado de sus componentes” (Tsai et al., 2018, p. 1). El propósito fundamental del monitoreo de red radica en respaldar las operaciones de gestión. Este monitoreo presenta una representación visual del estado de la red, exponiendo su funcionamiento, un aspecto de suma utilidad para análisis posteriores como la Ingeniería de Tráfico (TE, Traffic Engineering), la Calidad de Servicio (QoS, Quality of Service) y la detección de anomalías (Anomaly Detection). En el entorno de las redes de computadoras, el modelo operativo de la red abarca diversas capas y múltiples dispositivos interconectados, que intercambian información entre sí. Para satisfacer los diferentes requerimientos de gestión es necesario para el administrador de red tener una vista global y estadísticas de uso.

Según Tsai et al. (2018) el proceso de monitoreo de red tiene 5 fases definidas en Figura 13.

Figura 13*Proceso de Monitoreo de Red*

Nota. “Network Monitoring in Software-Defined Networking: A Review” (p. 2), por P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, C.-S. Yang, 2018, *IEEE Systems Journal*, 12(4).

Recolección (*Collection*): Los dispositivos SDN pueden grabar información de tráfico usando funciones de recolección. Incluso con el uso de software personalizado se amplía la flexibilidad de medición de datos que los dispositivos legados basados en agentes de recolección. SDN decide cuáles de los dispositivos debería ser observador e intercambiar la frecuencia de “polling” de datos a través del controlador (Tsai et al., 2018).

Preprocesamiento (*Preprocessing*): En esta fase, los métodos de filtrado e indexación para SDN distinguen los datos válidos de los inválidos. Estos leen directamente los datos de los dispositivos a diferencia de las mediciones matemáticas y estadísticas de los dispositivos legados (Tsai et al., 2018).

Transmisión (*Transmission*): En las redes legadas, existe el protocolo estándar SNMP para el transporte de datos a través de la red. Sin embargo, en el paradigma SDN, la interfaz “southbound API” permite a los desarrolladores diseñar estructuras de datos personalizadas. También existen protocolos estándar como OpenFlow y su componente OFPMT METER (Tsai et al., 2018).

Análisis (*Analysis*): En esta fase se realizan diferentes funciones de interpretación de los datos recolectados. Estos datos en SDN pueden servir para

ajustar la red automáticamente. En cambio, en redes legadas esto solo se realiza manualmente por un operador (Tsai et al., 2018).

Presentación (*Presentation*): Finalmente, los datos analizados deben ser mostrados al administrador para operaciones de alto nivel. En SDN ofrece interfaces programables para el desarrollo de aplicaciones a través de la interfaz northbound esto facilita la visualización del comportamiento y anomalías en la red (Tsai et al., 2018).

SDN trae una arquitectura pensada en la automatización con software para aminorar los problemas existentes en las redes legadas. Sin embargo, Tsai et al. (2018) también muestra campos de investigación abiertos en monitoreo para mejorar las soluciones en SDN:

- En las tareas de **Análisis** se destaca la *detección de anomalías*; detectar tráficos de red anómalo mediante diferentes técnicas legadas hasta de aprendizaje automático. Así mismo, la *gestión de fallos*; dividiendo este campo en dos: la detección y recuperación de fallos.
- En las tareas de **Presentación** también se encuentran problemas y oportunidades de mejora en:
 - *Topology GUI*: El descubrimiento de la topología se hace mediante el protocolo LLDP, los administradores de la red necesitan visualizar la topología de la red.
 - *Presentación en tiempo real*: los datos monitoreados necesitan ser presentados a los administradores de red de forma inmediata.
 - *Interfaz de gestión*: permitiendo a los administradores modificar el comportamiento de la red.

Dimensiones

A. Eficiencia

Según Tsai et al. (2018) nos dice que: “hacer el monitoreo de red más eficiente es una tarea crucial, reduciendo la sobrecarga a la red midiendo las estadísticas y el estado de la red” (p. 8). La eficiencia del monitoreo de SDN puede verse afectada por diferentes parámetros de red. Para evaluar y medir dicha eficiencia, se utilizan métricas, como el uso de CPU y memoria, permitiendo analizar el impacto computacional generado por el procesamiento de información adicional. (Lange et al., 2018). Es por ello que los recursos computacionales que se consumen por los desarrollos tecnológicos también deben ser medidos y analizados en la investigación. De acuerdo a Hong y Varghese (2019) se consideran recursos computacionales a hardware, software y redes. Entonces el uso de recursos computacionales se refiere a la cantidad que utiliza con respecto a la disponibilidad de dicho recurso.

Cantidad de Uso de CPU: Determina el consumo de CPU durante la implementación y funcionamiento del NMS.

Cantidad de Uso de Memoria: Determina el consumo de Memoria durante la implementación y funcionamiento del NMS.

B. Presentación

La visualización o presentación de información es la representación visual e interactiva de datos que soportadas computacionalmente permiten la generación de información y conocimiento para apoyar las tareas de gestión. En la red, la visualización de la topología es un parte fundamental en el análisis de los administradores. Dicha visualización requiere de un tiempo de respuesta en los dispositivos para ser mostrados (Chen et al., 2021).

Tiempo de visualización de la topología: El tiempo de respuesta de la herramienta cuando se ejecuta la operación de visualización de topología (Montoya-Munoz et al., 2021; Villota et al., 2018).

De acuerdo a Tsai et al. (2018) la importancia de la visualización en el proceso de monitoreo de red recae en la detección de anomalías de red en tiempo real. Es por ello que el tiempo de presentación de los datos de red deben ser tomados en cuenta en el desarrollo de las investigaciones.

Gestión de fallos en SDN

Según Yu et al. (2019), dentro de una red, un fallo es la imposibilidad de que la red o uno de sus componentes ejerzan correctamente sus funciones; un error es una acción humana u otro factor que produzca un resultado erróneo; y un defecto, o más comúnmente conocido como un bug es la manifestación de un error en forma de una condición incorrecta o defecto que puede causar que la red se comporte de manera diferente a la prevista. Por ende, el resultado de un error es un defecto, y un defecto puede llevar a un fallo. La ocurrencia de fallos, errores, y defectos son las más comunes y directas formas por la cuáles la fiabilidad de la red es puesta en duda.

El manejo de fallos es el proceso de detectar, localizar, resolver y prevenir estos fallos en la red. Con lo expuesto, el diseño de soluciones de manejos de fallos es indispensable para satisfacer la necesidad de fiabilidad en las redes. Yu et al. (2019) nos define un framework de taxonomía de doble entrada sobre las soluciones de manejo de fallos SDN en la literatura académica, así mismo define cuatro tareas del proceso de manejo de fallos. Estos son: **Monitoreo de Sistema, Diagnóstico de Fallos, Reparación y recuperación del Fallo y Tolerancia a Fallos.**

Fiabilidad en la red

Fiabilidad se refiere a la probabilidad de operaciones libre de fallos durante un periodo de tiempo y bajo condiciones específicas. En la redes SDN es muy importante mantener esta fiabilidad por lo que en el manejo de fallos es importante lidiar con los defectos originados por los fallos (Yu et al., 2019).

Acceso remoto a laboratorios en tiempo de COVID-19

Según Mukhopadhyay et al. (2020) la pandemia global del COVID-19 ha rápidamente transformado las formas tradicionales de operación en la salud y la educación. Las medidas adoptadas por distintos gobiernos alrededor del mundo han llevado a que la educación sea de forma remota. Esto también ha supuesto un reto para el acceso a los laboratorios de telecomunicaciones que cuentan las distintas facultades de educación universitaria. Por lo que es necesario el empleo de diferentes tecnologías que permitan el acceso y control a estos laboratorios de manera remota, lo cual potenciará el aprendizaje en las especializadas asignadas.

En especial, los laboratorios que permiten a los estudiantes aprender diferentes tecnologías de última generación en telecomunicaciones en un mundo más conectado, donde cada día se hace más necesario implementar mecanismos para la seguridad de los datos hace esencial el acceso remoto a estos laboratorios y el estudio en ciberseguridad (Robles-Gómez et al., 2020). De acuerdo con Vollbrecht et al. (2020), recomiendan a las facultades considerar las necesidades de sus estudiantes y utilizar las mejores herramientas para facilitar el aprendizaje. También recomienda proveer mayores experiencias basadas en aplicaciones.

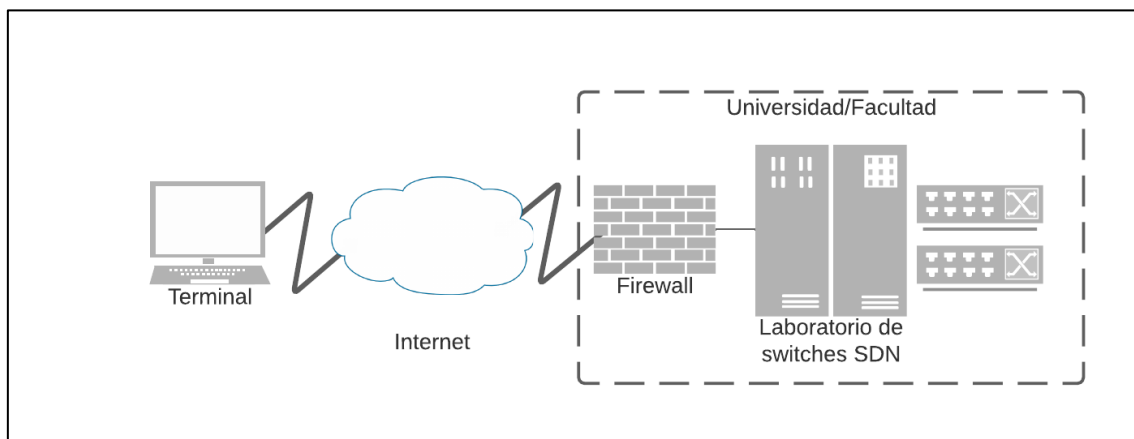
Es en base a esto que diferentes autores han provisto de soluciones tecnológicas para facilitar el monitoreo y control remoto a los laboratorios de telecomunicaciones. Los laboratorios son esenciales para el aprendizaje de los estudiantes, por lo que el acceso a ellos remotamente debido a las restricciones de

aprendizaje presencial en espacios cerrados se hace imprescindible (Mohammed et al., 2020).

Estas propuestas se fundamentan en la estructura de acceso delineada en la Figura 14. Esta arquitectura es el fundamento de la investigación y el punto de inicio del diseño y simulación del Sistema de Monitoreo de Red-NMS para la detección de fallos de enlace en una Red Definida por Software, ubicado en la Facultad de Ingeniería Eléctrica y Electrónica-FIEE de la Universidad Nacional de Ingeniería.

Figura 14

Arquitectura de referencia para el acceso remoto de laboratorios



El laboratorio al que se hace referencia se muestra en la Figura 15, este laboratorio es inaccesible presencialmente para los estudiantes de la FIEE por la pandemia del COVID-19.

Figura 15

Fotografía de referencia de laboratorio SDN de la FIEE-UNI



Nota. Fotografía tomada por el autor

Finalmente, se recalca la función del acceso remoto en la labor educativa y su importancia en el contexto de restricciones de acceso a los espacios de los laboratorios. Sin embargo, se deben hacer los esfuerzos para mejorarlo en beneficio de los estudiantes.

Definición conceptual de la terminología empleada

Redes Definidas por Software (SDN, *Software-Defined Networking*): Un enfoque de redes programables que soporta la separación de los planos de control y reenvío mediante interfaces estandarizadas (Jain et al., 2019).

Dispositivo de Red (*Network Device*): Un dispositivo o elemento que realiza una o más operaciones de red relacionadas a la manipulación o reenvío de paquetes (Jiménez et al., 2021).

Interfaz (*Interface*): Un punto de interacción entre dos entidades. La interfaz es usualmente implementada a través de un protocolo de red o Application Programming Interface (API) (Jain et al., 2019).

Aplicación (App, *Application*): Las aplicaciones en el contexto de SDN brindan una gestión centralizada de políticas y reglas de red. Además, también ofrecen una variedad de funciones que permiten a los administradores resolver de manera efectiva los problemas de red (Zhao et al., 2019).

Plano de Datos (DP, *Data Plane*): La colección que recursos que reúne a todos los dispositivos de red responsables de asegurar el tránsito apropiado de tráfico de reenvío y el manejo de todas las operaciones de los dispositivos de red individuales (Khan et al., 2023).

Plano de Control (CP, *Control Plane*): La colección de funciones responsables por el control de uno o más dispositivos de red. CP instruye a los dispositivos de red con respecto a cómo deben procesar y reenviar paquetes (Khan et al., 2023).

Plano de Gestión (MP, *Management Plane*) o **Aplicación** (AP, *Application Plane*): La colección de funciones responsables por el monitoreo, configuración, y mantenimiento de uno o más dispositivos de red o partes de este último (Khan et al., 2023).

Sistema de Monitoreo de Red (NMS, *Network Monitoring System*): Las aplicaciones de monitoreo de red son componentes de software que son utilizados por los administradores de red para asistirlos en sus funciones de gestión de la red. Por ende, los sistemas de monitoreo de red-NMS son necesarias para tener una vista global de la red (Ndiaye et al., 2020).

Proceso de Monitoreo de Redes Definidas por Software: El monitoreo emerge como un concepto crucial en la administración de redes, ya que proporciona a los operadores una visión del comportamiento y la condición de los elementos de la red. En las redes de computadoras dentro del paradigma de SDN, el modelo de operaciones de red incluye diferentes capas y numerosos dispositivos enlazados que

intercambian datos con el controlador SDN y entre sí. Para satisfacer los diferentes requerimientos de gestión es necesario para el administrador de red tener un vista global y estadísticas de uso (Tsai et al., 2018).

2.1. Tipo y diseño de la investigación

Tipo de investigación

Esta investigación es del tipo:

Aplicada: Porque se van a hacer uso de los conocimientos adquiridos por medio de la investigación y desarrollo con el objetivo de resolver un problema en el proceso de monitoreo de redes definidas por software (Hernández-Sampieri y Mendoza Torres, 2018).

Nivel de investigación

Explicativa: “Porque su interés se centra en explicar por qué ocurre un fenómeno y en qué condiciones se manifiesta, o por qué se relacionan dos o más variables” (Hernández-Sampieri y Mendoza Torres, 2018, p. 112).

Diseño de investigación

Esta investigación tiene un diseño:

Pre experimental: Esta investigación es de tipo preexperimental porque a un grupo se le aplica una prueba previa al estímulo o tratamiento experimental, después se le administra el tratamiento y finalmente se le aplica una prueba posterior al estímulo. Este diseño ofrece la ventaja de la existencia de un punto referencial inicial, para ver qué nivel tenía un grupo antes del estímulo (Hernández-Sampieri y Mendoza Torres, 2018, p. 163).

2.2. Población, muestra y muestreo

En este informe de Tesis, se ha optado por realizar las pruebas en ambientes emulados de procesos de monitoreo de redes definidas por software en entornos Cloud controlados. Por ello se ha determinado realizar 30 procesos de control; como población y muestra, a cada uno de los objetivos específicos junto a los indicadores a evaluar, lo cual se muestra en la Tabla 4.

Tabla 4*Cantidad de pruebas realizadas*

Objetivo Específico	Número de Pruebas
Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye el tiempo de visualización de la topología.	30
Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye la cantidad de uso de memoria.	30
Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye la cantidad de uso de CPU.	30

Así mismo, cada uno de los procesos de control tiene una distribución diferente de los números de los switches y enlaces, ordenados en la Tabla 5.

Tabla 5*Pruebas y características de cada proceso de control*

Proceso de Control	Switches	Enlaces	Hosts
1	1	1	1
2	2	3	2
3	4	7	4
4	6	11	6
5	8	15	8
6	10	19	10
7	12	23	12
8	14	27	14
9	16	31	16
10	18	35	18
11	20	39	20
12	22	43	22
13	24	47	24
14	26	51	26
15	28	55	28
16	30	59	30
17	32	63	32
18	34	67	34
19	36	71	36
20	38	75	38
21	40	79	40
22	42	83	42

23	44	87	44
24	46	91	46
25	48	95	48
26	50	99	50
27	52	103	52
28	54	107	54
29	56	111	56
30	58	115	58

2.3. Hipótesis

Hipótesis general

Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de redes definidas por software.

Hipótesis específicas

- Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología.
- Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.
- Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de CPU.

2.4. Variables y Operacionalización

Variables

Variable independiente (X)

- Sistema de Monitoreo de Red-NMS de detección de fallos de enlace

Variable dependiente (Y)

- Proceso de monitoreo de redes definidas por software
 - Indicadores
 - Tiempo de visualización de la topología.
 - Cantidad de uso de memoria.

- Cantidad de uso de CPU.

2.5. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Método de investigación

El informe de la tesis utiliza la ruta, método o enfoque cuantitativo porque se llevó a cabo un análisis estadístico de los datos recolectados para evaluar las dimensiones de la variable de estudio y extraer conclusiones respecto de la hipótesis (Hernández-Sampieri y Mendoza Torres, 2018).

Técnica de investigación

La investigación empleó las técnicas e instrumentos descritos en la Tabla 6, los cuales permitieron la recolección de datos.

Tabla 6

Técnicas, instrumentos y herramientas de la investigación

Técnicas	Instrumentos	Herramientas
Observación	Ficha de Observación	Navegador Chrome
		Script escrito en Python

Dentro del informe de Tesis se aplicó la técnica de observación y un instrumento: Ficha de Observación, este instrumento tiene dos herramientas: cronómetro y contador que permiten recolectar los datos necesarios en tiempo y cantidad respectivamente para evaluar las dimensiones definidas anteriormente de la variable de estudio.

Análisis de fiabilidad de las variables

De acuerdo Hernández-Sampieri y Mendoza Torres (2018) indica que en la investigación: “existen múltiples instrumentos para medir toda clase de variables y en algunos casos puedes combinar varias técnicas de recolección de los datos” (p. 250). Para este estudio, se empleó el siguiente instrumento con el propósito de recolección de datos.

Instrumento:**Ficha de Observación**

Conforme lo señalado por Hernández-Sampieri y Mendoza Torres (2018), la observación representa un método de recolección de datos que conlleva el registro sistemático, válido y confiable de comportamientos y situaciones observables, utilizando una serie de categorías y subcategorías.

Confiabilidad:

Según Hernández-Sampieri y Mendoza Torres (2018) indica que “la confiabilidad o fiabilidad de un instrumento de medición se refiere al grado en que su aplicación repetida al mismo individuo, caso o muestra produce resultados iguales” (p. 228).

Cálculo de la confiabilidad o fiabilidad:

El cálculo de la confiabilidad de un instrumento puede ser abordada mediante diversas metodologías y utilizando variados procesos. De acuerdo con Hernández-Sampieri y Mendoza Torres (2018), todos estos enfoques involucran procedimientos y fórmulas que generan coeficientes que varían en un rango de cero a uno. Un coeficiente de cero denota una confiabilidad nula, mientras que uno representa una confiabilidad máxima. A medida que este coeficiente se aproxima a cero, aumenta la imprecisión en la medición con el instrumento.

Según Hernández-Sampieri y Mendoza Torres (2018) deja entrever que:

La validez, la confiabilidad y la objetividad no deben tratarse de forma separada, sino de manera interdependiente. Sin alguna de las tres, el instrumento no es útil para llevar a cabo un estudio (p. 238).

Para ello se realizó la correlación de Pearson, dónde los valores se interpretan en la Tabla 7.

Tabla 7*Coeficiente de Relación de Pearson*

R	Correlación
-0.90	Correlación negativa muy fuerte.
-0.75	Correlación negativa considerable.
-0.50	Correlación negativa media.
-0.25	Correlación negativa débil.
-0.10	Correlación negativa muy débil.
0.00	No existe correlación alguna entre las variables.
0.10	Correlación positiva muy débil.
0.25	Correlación positiva débil.
0.50	Correlación positiva media.
0.75	Correlación positiva considerable.
0.90	Correlación positiva muy fuerte.
1.00	Correlación positiva perfecta.

Nota. Recuperado de *METODOLOGÍA DE LA INVESTIGACIÓN: LAS RUTAS CUANTITATIVA, CUALITATIVA Y MIXTA* (p. 346), por Hernández-Sampieri y Mendoza Torres, 2018, McGraw-Hill Interamericana de España S.L.

De acuerdo con Hernández-Sampieri y Mendoza Torres (2018) indica que: “la confiabilidad varía de acuerdo con el número de indicadores específicos o ítems¹⁰ que incluya el instrumento de medición. Cuantos más ítems haya, mayor tenderá a ser la confiabilidad, lo cual resulta lógico” (p. 240).

También, Hernández-Sampieri y Mendoza Torres (2018) refiere que: “ Para estimar la confiabilidad de su instrumento debe aplicarse a su muestra y sobre la base de los resultados calcular tal coeficiente” (p. 240).

Confiabilidad

Se llevó a cabo un análisis de Test-Retest al propósito de determinar la confiabilidad del instrumento, utilizando una evaluación que abarcó 10 procesos de control. Para abordar esta situación, se aplicó la prueba de Normalidad de Shapiro-Wilk debido a la naturaleza de la evaluación con los 10 procesos de control. Se

formularon las hipótesis estadísticas para las pruebas de Normalidad y se estableció la regla de decisión.

Hipótesis estadísticas:

H₀: La muestra cuenta con una distribución Normal

H₁: La muestra cuenta con una distribución No Normal

Regla de decisión:

Nivel de confianza: 95%

Si $p < 0.05$, la hipótesis nula es rechazada, indicando que la muestra no sigue una distribución normal.

Si $p \geq 0.05$, se acepta la hipótesis nula, lo que sugiere que la muestra presenta una distribución normal.

Los indicadores que comprueben ser de distribución **Normal** se aplican Correlación de Pearson para medir la confiabilidad del instrumento.

Indicador 1: Tiempo de visualización de la topología

Tabla 8

Prueba de normalidad del indicador: tiempo de visualización de la topología

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Tiempo de Visualización de Topología Test	.121	10	.200*	.961	10	.797
Tiempo de Visualización de Topología Retest	.121	10	.200*	.956	10	.743

Nota. Elaborado con el Software SPSS Versión 25

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

De acuerdo con los resultados en la Tabla 8, el valor del sig. de Shapiro-Wilk para el test es 0.797, mientras que para el retest es 0.743. Los dos valores superan

el límite establecido en 0.05, lo que nos conduce a la aceptación de la hipótesis nula. Como resultado, se infiere que los datos siguen una distribución **normal**, lo que motiva la elección de la prueba de correlación de Pearson.

Tabla 9

Prueba de confiabilidad del indicador: tiempo de visualización de la topología

		Tiempo de Visualización de Topología Test	Tiempo de Visualización de Topología Retest
Tiempo de Visualización de Topología Test	Correlación de Pearson	1	.991**
	Sig. (bilateral)		.000
	N	10	10
Tiempo de Visualización de Topología Retest	Correlación de Pearson	.991**	1
	Sig. (bilateral)	.000	
	N	10	10

Nota. Elaborado con el Software SPSS Versión 25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Basándonos en los datos presentados en la Tabla 9, el coeficiente de correlación de Pearson es de 0.991. Según los criterios de Pearson, esta correlación señala una correlación **positiva muy fuerte**. Por consiguiente, podemos concluir que el instrumento es **confiable**.

Indicador 2: Cantidad de uso de memoria

Tabla 10

Prueba de normalidad del indicador: cantidad de uso de memoria

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Cantidad de Uso de Memoria Test	.159	10	.200*	.914	10	.311
Cantidad de Uso de Memoria Retest	.197	10	.200*	.942	10	.574

Nota. Elaborado con el Software SPSS Versión 25

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

De acuerdo con los resultados en la Tabla 10, el valor del sig. de Shapiro-Wilk para el test es 0.311, mientras que para el retest es 0.574. Los dos valores superan el límite establecido en 0.05, lo que nos conduce a la aceptación de la hipótesis nula. Como resultado, se infiere que los datos siguen una distribución **normal**, lo que motiva la elección de la prueba de correlación de Pearson.

Tabla 11

Prueba de confiabilidad del indicador: cantidad de uso de memoria

		Tiempo de Visualización de Topología Test	Tiempo de Visualización de Topología Retest
Cantidad de Uso de Memoria Test	Correlación de Pearson	1	.830**
	Sig. (bilateral)		.003
	N	10	10
Cantidad de Uso de Memoria Retest	Correlación de Pearson	.830**	1
	Sig. (bilateral)	.003	
	N	10	10

Nota. Elaborado con el Software SPSS Versión 25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Según los datos presentados en la Tabla 11, el coeficiente de correlación de Pearson es de 0.830. Según los criterios de Pearson, esta correlación es **positiva considerable**. Por consiguiente, se concluye que el instrumento es **confiable**.

Indicador 3: Cantidad de uso de CPU

Tabla 12

Prueba de normalidad del indicador: cantidad de uso de CPU

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Cantidad de Uso de CPU Test	.241	10	.105	.886	10	.154
Cantidad de Uso de CPU Retest	.187	10	.200*	.904	10	.241

Nota. Elaborado con el Software SPSS Versión 25

*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

De acuerdo con los resultados en la Tabla 12, el valor del sig. de Shapiro-Wilk para el test es 0.154, mientras que para el retest es 0.241. Los dos valores superan el límite establecido en 0.05, lo que nos conduce a la aceptación de la hipótesis nula. Como resultado, se infiere que los datos siguen una distribución **normal**, lo que motiva la elección de la prueba de correlación de Pearson.

Tabla 13

Prueba de confiabilidad del indicador: cantidad de uso de CPU

		Cantidad de Uso de CPU Test	Cantidad de Uso de CPU Retest
Cantidad de Uso de CPU Test	Correlación de Pearson	1	.864**
	Sig. (bilateral)		.001
	N	10	10
Cantidad de Uso de CPU Retest	Correlación de Pearson	.864**	1
	Sig. (bilateral)	.001	
	N	10	10

Nota. Elaborado con el Software SPSS Versión 25

** . La correlación es significativa en el nivel 0,01 (bilateral).

Según los datos presentados en la Tabla 13, el coeficiente de correlación de Pearson es de 0.864. Según los criterios de Pearson, esta correlación es **positiva considerable**. Por consiguiente, se concluye que el instrumento es **confiable**.

Validez

Según Hernández-Sampieri y Mendoza Torres (2018): "la validez se refiere al grado en que un instrumento mide realmente la variable que pretende medir" (p. 302). Con el propósito de evaluar la validez del instrumento utilizado para recopilar datos cuantitativos (Ficha de Observación), se recurrió al método del "juicio de experto". En este proceso, se contó con la colaboración de profesores de la escuela de Ingeniería de Sistemas, la Tabla 14 lista a los profesores consultados.

Tabla 14

Lista de especialistas que avalaron la validez del instrumento

DNI	Grado Académico	Nombres y Apellidos	Institución	Calificación
07139361	Mg.	Daniel Díaz Ataucuri	Universidad Nacional de Ingeniería / Universidad Nacional Mayor de San Marcos / Universidad Autónoma del Perú	Aplicable
17906323	Dr.	Javier Gamboa Cruzado	Universidad Nacional Mayor de San Marcos / Universidad Autónoma del Perú	Aplicable
41647498	Mg.	Celis Henry Ochoa Jayo	Universidad Autónoma del Perú	Aplicable

2.6. Procedimientos

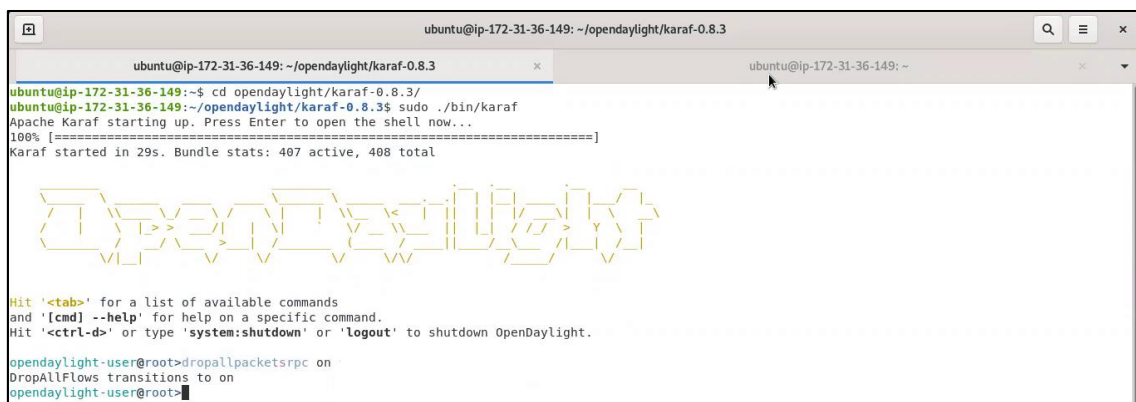
Esta subsección describe la recolección de los datos para analizar las variables de estudio. La recolección de datos se realizó siguiendo el procedimiento descrito a continuación para las prepruebas y postpruebas con diferentes configuraciones de switches. Específicamente, esta subsección muestra el procedimiento para el proceso de control 5 que representa 8 switches en topología lineal, como se indica en la Tabla 5.

Para la recolección de datos en ambas pruebas: preprueba (sin el NMS) y postprueba (con el NMS), se ejecuta el controlador SDN OpenDayLight y el comando “dropallpacketsrpc on” para evitar conflictos con el Cbench, como se ilustra en la

Figura 16.

Figura 16

Comandos para controlador SDN OpenDayLight



```
ubuntu@ip-172-31-36-149: ~/opendaylight/karaf-0.8.3
ubuntu@ip-172-31-36-149:~$ cd opendaylight/karaf-0.8.3/
ubuntu@ip-172-31-36-149:~/opendaylight/karaf-0.8.3$ sudo ./bin/karaf
Apache Karaf starting up. Press Enter to open the shell now...
100% [=====]
Karaf started in 29s. Bundle stats: 407 active, 408 total

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>dropallpacketsrpc on
DropAllFlows transitions to on
opendaylight-user@root>
```

Nota. Pantalla capturada por el autor

Esta configuración permitió realizar la obtención de datos de las prepruebas (sin el NMS). Para realizar la postprueba (con el NMS), se añade el NMS al entorno de pruebas. Para ello se ejecuta las aplicaciones que conforman el NMS como se puede ver en la Figura 17.

Figura 17

Ejecución del sistema de monitoreo de red (NMS)



The image shows two terminal windows. The top window is titled 'ubuntu@ip-172-31-36-149: ~/back-end/dev-back' and shows the execution of a Django application using 'python3 manage.py runserver 0.0.0.0:8000'. The output indicates that the server is running successfully on http://0.0.0.0:8000/. The bottom window is titled 'ubuntu@ip-172-31-36-149: ~/front-end/nuxt-front' and shows the execution of a Next.js application using 'npx run start'. The output displays the Next.js version (v2.15.6), environment (production), rendering (client-side), target (server), and memory usage (28.8 MB RSS, 92.2 MB). The application is listening on http://172.31.36.149:3000/.

```

ubuntu@ip-172-31-36-149:~/back-end/dev-back$ python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 03, 2021 - 20:37:32
Django version 3.0.5, using settings 'auth.settings'
Starting ASGI/Channels version 3.0.2 development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.

ubuntu@ip-172-31-36-149:~/front-end/nuxt-front$ npx run start
> nuxt-app-nms@1.0.0 start /home/ubuntu/front-end/nuxt-front
> nuxt start

WARN: You are using an unsupported version of Node.js (v10.19.0). It is recommended to use the latest LTS version (https://nodejs.org/en/about/releases)

Next @ v2.15.6
- Environment: production
- Rendering: client-side
- Target: server

Memory usage: 28.8 MB (RSS: 92.2 MB)
Listening: http://172.31.36.149:3000/
  
```

Nota. Pantalla capturada por el autor

La evaluación del rendimiento en el canal de control entre los switches y el controlador OpenDayLight hace uso de Cbench, una herramienta de medición de red para controladores OpenFlow (Mininet, 2018). La evaluación se hizo en solo un modo: “Latency Mode”. El modo latencia envía mensajes OpenFlow al controlador y espera por la respuesta del controlador antes de enviar un nuevo mensaje OpenFlow.

Para cada prueba con distintas configuraciones de switches, se ejecutó en modo latencia del Cbench y con las configuraciones de la herramienta Cbench por defecto. La Figura 18 muestra el comando usado para ejecutar Cbench en modo latencia y con 8 switches.

Figura 18

Ejecución de herramienta de red: Cbench

```

ubuntu@ip-172-31-36-149:~$ cbench -s 8
cbench: controller benchmarking tool
running in mode 'latency'
connecting to controller at localhost:6633
faking 8 switches offset 1 :: 16 tests each; 1000 ms per test
with 100000 unique source MACs per switch
learning destination mac addresses before the test
starting test with 0 ms delay after features reply
ignoring first 1 "warmup" and last 0 "cooldown" loops
connection delay of 0ms per 1 switch(es)
debugging info is off
19:27:59.570 8 switches: flows/sec: 111 106 93 90 98 2 98 113 total = 0.710999 per ms
19:28:00.670 8 switches: flows/sec: 1141 1140 1288 1268 1288 1291 1420 1305 total = 10.140402 per ms
19:28:01.771 8 switches: flows/sec: 2972 2688 2780 2939 2946 2783 2925 2796 total = 22.828932 per ms
19:28:02.871 8 switches: flows/sec: 3774 3757 3699 3711 3738 3625 3510 3561 total = 29.374941 per ms
19:28:03.971 8 switches: flows/sec: 3695 3748 3694 3786 3761 3744 3721 3771 total = 29.919731 per ms
19:28:05.071 8 switches: flows/sec: 4518 4405 4370 4448 4382 4382 4387 4363 total = 35.254788 per ms
19:28:06.172 8 switches: flows/sec: 1880 1877 1929 1834 1873 1868 1820 1777 total = 14.857955 per ms
19:28:07.272 8 switches: flows/sec: 4403 4333 4341 4524 4469 4291 4415 4413 total = 35.188930 per ms
19:28:08.372 8 switches: flows/sec: 4362 4448 4386 4265 4447 4388 4322 4205 total = 34.902756 per ms
19:28:09.472 8 switches: flows/sec: 4498 4569 4448 4484 4474 4507 4380 4411 total = 35.770356 per ms
19:28:10.573 8 switches: flows/sec: 4379 4269 4254 4347 4371 4174 4352 4302 total = 34.447966 per ms
19:28:11.673 8 switches: flows/sec: 3478 3483 3459 3346 3469 3452 3467 3289 total = 27.442726 per ms
19:28:12.773 8 switches: flows/sec: 3650 3531 3578 3605 3608 3609 3628 3620 total = 28.828971 per ms
19:28:13.873 8 switches: flows/sec: 4414 4322 4133 4361 4444 4489 4368 4238 total = 34.768896 per ms
19:28:14.973 8 switches: flows/sec: 3933 4236 4198 4178 4129 3996 4183 4189 total = 33.041901 per ms
19:28:16.074 8 switches: flows/sec: 4522 4502 4212 4630 4331 4620 4316 4509 total = 35.641180 per ms
RESULT: 8 switches 15 tests min/max/avg/stddev = 10140.40/35770.36/29494.03/7637.85 responses/s
ubuntu@ip-172-31-36-149:~$

```

Nota. Pantalla capturada por el autor

Cada resultado de la prueba fue registrado en una tabla que tiene de entradas la configuración de la prueba, la prueba sin NMS (preprueba) y la prueba con el NMS (postprueba).

Recolección de datos del indicador de cantidad de uso de memoria y CPU

El indicador cantidad de uso de memoria se refiere a la evaluación de rendimiento de la aplicación en términos de cantidad de uso de memoria RAM que utiliza el sistema para realizar el monitoreo. El indicador uso de CPU se refiere a la evaluación de rendimiento de la aplicación en términos de la cantidad de uso de CPU que utiliza el sistema para realizar el monitoreo. Para la evaluación se hace uso de la herramienta "test_cpu_memory.py", mostrada en la Figura 19, elaborada por el autor en python3.8 haciendo uso de la librería psutil. Esta herramienta toma un registro de la memoria y CPU cada segundo durante un tiempo determinado. Cada medición es registrada en un archivo que contiene las mediciones por segundo, y el promedio de todas las mediciones en este intervalo.

Figura 19

Ejecución de herramienta para memoria y CPU

```

ubuntu@ip-172-31-36-149: ~/metrics
├── Desktop
├── Downloads
├── Pictures
├── Templates
├── back-end
├── metrics
├── mininet
├── oftest
├── openflow
├── prueba1.txt
├── pruebas3.txt
├── Documents
├── Music
├── Public
├── Videos
├── front-end
├── metrics.zip
├── oflops
├── opendaylight
├── pox
├── pruebas2.txt
├── thinclient_drives
└──

ubuntu@ip-172-31-36-149:~/metrics$ cd metrics/
ubuntu@ip-172-31-36-149:~/metrics$ ls
confiabilidad_final  t_p_c_15.txt  t_p_c_23.txt  t_p_c_5.txt  t_post_p_c_13.txt  t_post_p_c_21.txt  t_post_p_c_3.txt  test_cpu_memory.py
primeras-pruebas    t_p_c_16.txt  t_p_c_24.txt  t_p_c_6.txt  t_post_p_c_14.txt  t_post_p_c_22.txt  t_post_p_c_30.txt  test_cpu_memory.py.save
t_p_c_4.txt         t_p_c_17.txt  t_p_c_25.txt  t_p_c_7.txt  t_post_p_c_15.txt  t_post_p_c_23.txt  t_post_p_c_4.txt
t_p_c_1.txt         t_p_c_18.txt  t_p_c_26.txt  t_p_c_8.txt  t_post_p_c_16.txt  t_post_p_c_24.txt  t_post_p_c_5.txt
t_p_c_10.txt        t_p_c_19.txt  t_p_c_27.txt  t_p_c_9.txt  t_post_p_c_17.txt  t_post_p_c_25.txt  t_post_p_c_6.txt
t_p_c_11.txt        t_p_c_2.txt  t_p_c_28.txt  t_post_p_c_1.txt  t_post_p_c_18.txt  t_post_p_c_26.txt  t_post_p_c_7.txt
t_p_c_12.txt        t_p_c_20.txt  t_p_c_29.txt  t_post_p_c_10.txt  t_post_p_c_19.txt  t_post_p_c_27.txt  t_post_p_c_8.txt
t_p_c_13.txt        t_p_c_21.txt  t_p_c_3.txt  t_post_p_c_11.txt  t_post_p_c_2.txt  t_post_p_c_28.txt  t_post_p_c_9.txt
t_p_c_14.txt        t_p_c_22.txt  t_p_c_30.txt  t_post_p_c_12.txt  t_post_p_c_20.txt  t_post_p_c_29.txt  test

ubuntu@ip-172-31-36-149:~/metrics$ python3 test_cpu_memory.py
Nombre del proceso de control:
proceso_de_control_5_switches_8_pre
1
2
3
4
5
6
7

```

Nota. Pantalla capturada por el autor

Para cada prueba con distintas configuraciones de switches, se ejecutó la herramienta que tomaba las mediciones cada segundo en 180 segundos. Y se toma en cuenta como resultado el promedio de las mediciones durante los 180 segundos. El resultado para cada configuración es el promedio de las mediciones en los 180 segundos. Cada resultado de la prueba fue registrado en una tabla que tiene de entradas la configuración de la prueba, la prueba sin NMS y la prueba con el NMS.

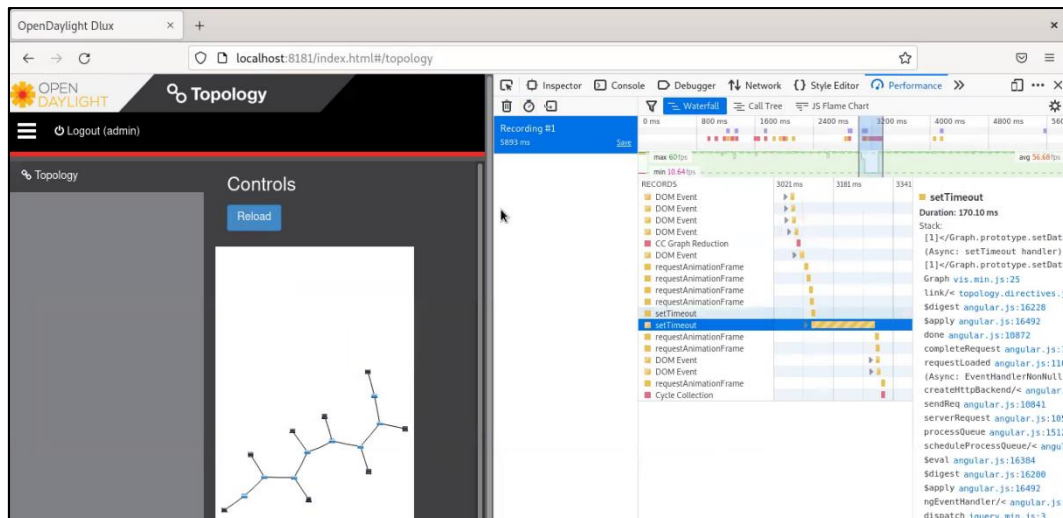
Recolección de datos de indicador tiempo de visualización de topología

El indicador tiempo de visualización de topología se refiere a la evaluación de la visualización de la aplicación en términos de tiempo que utiliza el sistema para presentar la topología monitoreada. En este indicador, la medición sin el NMS (preprueba) toma en cuenta los valores obtenidos del componente web que muestra la topología en OpenDayLight. Para la evaluación se hace uso de la herramienta de desarrollo web del navegador Chrome, DevTools. Esta herramienta registra mediante una función denominada “Performace”, los tiempos de respuesta de los componentes de visuales de una aplicación web, es útil en la medida que registra los tiempos de

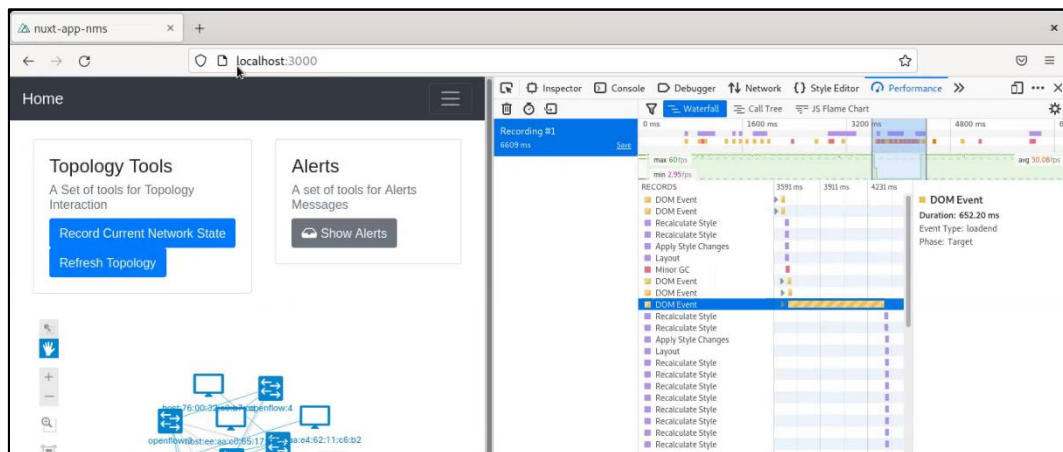
los componentes específicos para mostrar la topología. La función de registro de tiempos se activa cuándo en la sección "Performance" se comienza presionando el botón "start". Para cada prueba con distintas configuraciones de switches, se ejecutó la herramienta que registra los tiempos de respuesta de los componentes web, en particular, se toma en cuenta solo el tiempo del componente web que muestra la topología como se puede ver en la Figura 20 (a) para la preprueba (sin el NMS) y Figura 20 (b) para la postprueba (con el NMS).

Figura 20

Pantallas de tiempo de visualización de componentes topología



(a) preprueba (sin el NMS)



(b) postprueba (con el NMS).

Nota. Pantalla capturada por el autor

Cada resultado de la prueba fue registrado en una tabla que tiene de entradas la configuración de la prueba, la prueba sin NMS (preprueba) y la prueba con el NMS (postprueba).

2.7. Análisis de datos

Los datos obtenidos mediante los instrumentos de investigación se sometieron a procesamiento utilizando el programa IBM SPSS Statistics 25 en el entorno del sistema operativo Windows 10. En este entorno, se llevaron a cabo análisis e

interpretaciones de los mismos a través de métodos estadísticos descriptivos e inferenciales.

2.8. Aspectos éticos

Durante la investigación, se enfatizó la procedencia exclusiva de los datos de mediciones de recursos computacionales, como el tiempo de visualización, el uso de memoria y la capacidad de procesamiento (CPU). Como resultado, no se emplearon datos relacionados con participantes humanos, eliminando así inquietudes éticas en términos de privacidad, consentimiento informado o confidencialidad de datos personales. En todo momento, se mantuvo un compromiso firme con los principios fundamentales de integridad y ética académica, lo que aseguró la precisión, objetividad y la correcta citación de fuentes durante todas las etapas de la investigación. La honestidad y el rigor científico se mantuvieron como pilares esenciales en el desarrollo del estudio, a pesar de la naturaleza técnica y computacional de los datos analizados. Además, se siguieron las directrices de integridad académica y honestidad al presentar resultados y conclusiones, destacando la transparencia y el rigor ético que guiaron todo el proceso de investigación.

Estudio de factibilidad

Factibilidad técnica

El informe de tesis es técnicamente factible ya que los recursos en cuanto a Software y Hardware son accesibles tanto físicamente como virtualmente. Los datos presentados se midieron en los servicios cloud de Amazon Web Services (AWS), específicamente la herramienta EC2 y la configuración t2.xlarge. Los recursos informáticos necesarios son descritos en la Tabla 15.

Tabla 15

Recursos informáticos

	Tipo		Descripción
Hardware	Laptop		Windows© 10 Intel-i5© 8 th RAM 4Gb disco duro 1TB
	AWS Service	Web	Conjunto de servicios de infraestructura virtualizada y tecnologías en la nube pública. Herramienta EC2 y configuración t2.xlarge (Ubuntu server Intel© Xeon Scalable 3,0 GHz 8 núcleos virtuales 32 GB RAM disco duro 1TB SSD)
Software	Lenguaje de programación		Python 3.8/Javascript
	Mininet		Software de emulación de redes SDN
	MongoDB		Base de datos no relacionales
	Editor de código		Visual Studio Code
	Framework		Cisco NeXt-Ui Bootstrap 4 NuxtJS Django
	Microsoft Office 2019	Office	Suite de aplicaciones de Microsoft tales como Word, Excel, Power Point.

Factibilidad operativa

El informe de tesis es factible operativamente ya que se dispone de información necesaria para el desarrollo e implementación del NMS. Esto enmarcado en un ambiente de investigación científica tiene el valor de autoaprendizaje, la investigación, y apoyo de nuevas tecnologías como SDN. Además se encontraron, luego de una revisión de la literatura, investigaciones relacionadas a las herramientas de monitoreo de SDN.

Factibilidad económica

El informe de tesis es factible económicamente porque los recursos financieros necesarios existen y están a disposición directa o indirectamente para su uso, como se puede ver en la Tabla 16. La investigación, por ende, es sustentable económicamente.

Tabla 16

Recursos de la investigación

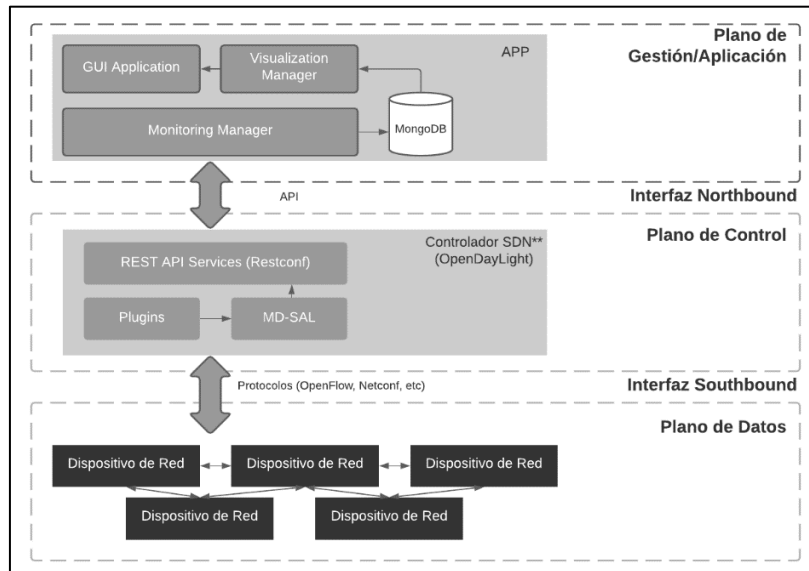
Recursos	Cantidad	Unidad	Costo unitario (S/)	Total (S/)
1. Recurso humano				
Investigador	12	Meses	S/1,200.00	S/14,400.00
Supervisor Inictel-Uni	10	Meses	S/6,000.00	S/60,000.00
Supervisor Inictel-Uni	10	Meses	S/6,000.00	S/60,000.00
2. Recurso técnicos				
2.1 Hardware				
Laptop	1	Unidad	S/2,000.00	S/2,000.00
2.2 Software				
Amazon Web Services	10	Meses	S/1,600.00	S/16,000.00
Microsoft office 2019	1	Año	S/220.00	S/220.00
Microsoft OneDrive	1	Año	S/500.00	S/500.00
Windows	1	Año	S/1,000.00	S/1,000.00
2.3 Otros				
Internet	12	Meses	S/200.00	S/2,400.00
				S/154,120.00

Modelamiento

La Figura 21 se muestra la aplicación en el paradigma de SDN, esta aplicación se encuentra en el Plano de Gestión/Aplicación, este plano se encarga de analizar los datos generados por el controlador y generar las órdenes correspondientes a la gestión de la red.

Figura 21

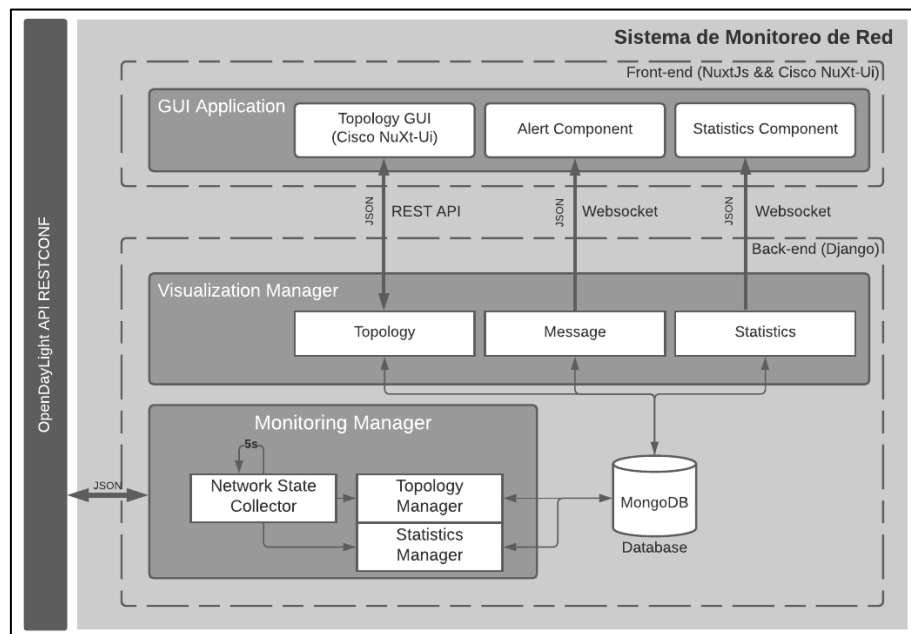
Aplicación SDN en el paradigma de Redes Definidas por Software (SDN)



En la Figura 22 se muestra la Arquitectura de Software de la Aplicación SDN de monitoreo, las tecnologías involucradas y los mecanismos internos y su interacción con el controlador SDN.

Figura 22

Arquitectura de Software del Sistema de Monitoreo de Red



Monitoring Manager – Se encarga de recuperar información actualizada sobre la red, analizar los estados de los dispositivos y enlaces, y almacenar la información en una base de datos local. El componente Network State Collector recoge información como estadísticas de tráfico, topología y datos de los dispositivos de la red, accediendo por medio de la interfaz northbound a uno o varios controladores situados en el plano de control, el tiempo configurado para esta recolección es de 5 segundos. En este caso, se utiliza la API RESTful proporcionada por el controlador OpenDayLight para acceder a la información sobre la topología física, incluidos los enlaces, los switches y los hosts en formato Yang. Además, el controlador también permite recoger los contadores de tráfico de datos de cada regla instalada en cada switch. El componente de Topology Manager es el encargado de realizar las operaciones de la topología de la red como comparar las topologías de red con switches y enlaces, así también, crear las alertas de detección de enlaces caídos. El componente Statistics Manager permite definir las funciones para guardar las

estadísticas de la red de los switches. Ambos componentes guardan los datos procesados en un base de datos no relacional.

Visualization Manager – Comprende los componentes denominados Topology, Statistics, Message. Con los datos almacenados en la Base de Datos no relacional, los tres componentes son los encargados de revisar los cambios en la base de datos para inmediatamente mostrarlos en la interfaz gráfica de usuario (GUI). El componente Topology es el encargado de gestionar las peticiones HTTP y preparar el JSON que es visto en la aplicación GUI u otras aplicaciones. Así también, el componente Message es el encargado de preparar en formato JSON los nuevos mensajes de alerta generados por el Monitoring Manager a los clientes suscritos por medio de WEBSOCKET. También usando WEBSOCKET, el componente Statistics es el encargado de actualizar las estadísticas de la red, esto se hace mediante la utilización de respuestas JSON. Así, a medida que la red es supervisada periódicamente por el componente Monitoring Manager, las visualizaciones se actualizan al mismo ritmo.

GUI Application – Es una aplicación cliente que se ejecuta en un navegador web. Esta aplicación comprende tres componentes principales: Topology GUI, Alert Component, Statistics Component. Topology GUI es un componente NuxtJS que tiene incrustado una instancia del framework Cisco NeXt-UI, este componente se encarga de mostrar la topología de forma visualmente interactiva y dinámica, de esta manera también nos permite mostrar los enlaces caídos en color gris en el momento que estos enlaces ya no se detectan. Alert Component es un componente de NuxtJS que crea un enlace WEBSOCKET con el componente Message para recibir y presentar los mensajes de alerta. Este componente también indica al componente Topology GUI que refresque la vista para mostrar la nueva topología. Finalmente, Statistics

Component es un componente NuxtJS que presenta las estadísticas de los switches de manera general, es una plataforma que solo muestra los datos directos de los switches.

La interacción entre el Monitoring Manager y Visualization Manager a través de una base de datos no relacional común que permite a GUI Application mostrar visualizaciones de la red, tales como: la vista de la topología, la detección de enlaces caídos (**Anexo 6**), los mensajes de alerta, datos de los switches y la visualización de estadísticas de la red por switches.

El NMS fue diseñado siguiendo una arquitectura cliente-servidor y se eligió Python 3.8 con el framework Django 3.0.5 para la aplicación Back-end, y se eligió NuxtJS 2.15.3 y Cisco NeXt-Ui 0.9 para la aplicación Front-end. La base de datos utilizada para almacenar la información de los dispositivos de red, así como las estadísticas de tráfico es MongoDB y ha sido implementada como Django Models a través de la librería Django. La funcionalidad de monitorización fue desarrollada como Clases y Jobs, visualización como Vistas y la GUI como una aplicación interactiva en Javascript.

Metodología aplicada

El desarrollo de la tesis se llevó a cabo usando el marco de trabajo SCRUM el cuál fue seleccionado porque ofrece herramientas ágiles para la adopción al cambio de requerimientos en una investigación y desarrollo (Ilyés, 2019). Los elementos de SCRUM elaborados para el presente trabajo fueron elaborados con Azure DevOps, herramienta disponible para la gestión de proyectos de Microsoft©. Los roles dentro del equipo de investigación y desarrollo fueron distribuidos de la siguiente forma:

- Development Team: Andres Junior Aparcana Tasayco, Victor Salazar
- Product Owner: Daniel Díaz Ataucuri

- Scrum Master: Fredy Mendoza Cárdenas

Sprint 0

En el sprint 0 se definen los requerimientos iniciales para la formulación de la investigación, acotando los alcances de la investigación y desarrollo para dar viabilidad a la tesis. Es por ello que se define los EPIC los suficientemente grandes para abarcar la totalidad de la propuesta de la tesis – Sistema de Monitoreo de Red, los EPIC-1 y EPIC-2 son descritos en las Figura 23 y Figura 24, respectivamente.

Figura 23

EPIC-1 Detección de topología y estado de enlace, alcance de la tesis

EPIC 2

2 EPIC-1 Detecting links states and topology

ANDRES JUNIOR APARCANA TASAYCO 0 comments OpenDayLight X Topology X +

State **Done** Area sdn-monitoring-system
Reason Work finished Iteration sdn-monitoring-system

Description

As a laboratory manager,
I want to visualize the link states and the topology
so that monitoring the FIEE SDN laboratory

Acceptance Criteria

Discussion

AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Status

Start Date
06/04/2021 0:00
Target Date
17/07/2021 0:00

Details

Priority
2
Effort
20
Business Value
40
Time Criticality
Value area
Business

Nota. Pantalla capturada por el autor

Figura 24

EPIC-2 Monitoreo de estadísticas de la red, alcance de la tesis

EPIC 42

42 EPIC-2 Statistics Monitoring

ANDRES JUNIOR APARCANA TASAYCO 0 comments Monitoring X OpenDayLight X SDN X Statistics X +

State **Done** Area sdn-monitoring-system
Reason Work finished Iteration sdn-monitoring-system\Sprint 5

Description

As a Network Operator
I want to view statistics from the switches on the screen
so that I can monitoring the packets generated from the network

Acceptance Criteria

Discussion

AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Status

Start Date
26/07/2021 0:00
Target Date
06/08/2021 0:00

Details

Priority
1
Effort
40
Business Value
40
Time Criticality
20
Value area
Business

Nota. Pantalla capturada por el autor

Los elementos EPIC-1 denominado “Detección de topología y estado de enlace” en la Figura 23 y EPIC-2 denominado “Monitoreo de estadísticas” en la Figura 24 son descompuestos en las características principales que tiene el sistema. Esta descomposición resulta en las características descritas en la las figuras Figura 25 y Figura 26 relacionada al EPIC-1, y la Figura 27 relacionada al EPIC-2.

Figura 25

Elaboración de la Característica del Sistema 1

The screenshot shows a Jira issue page for 'FE-1 Topology Visualization'. The issue is assigned to 'ANDRES JUNIOR APARCANA TASAYCO' and has 0 comments. The status is 'Done' and the area is 'sdn-monitoring-system'. The reason for completion is 'Work finished' and the iteration is 'sdn-monitoring-system'. The description is: 'As a laboratory manager, I want to visualize the SDN network topology so that getting a visualization of the FIEE SDN laboratory'. The acceptance criteria section is empty with a link to add criteria. The discussion section is empty with a placeholder for a comment. The details section shows: Priority 2, Effort 20, Business Value 20, Time Criticality, and Value area Business.

FEATURE 3	
3 FE-1 Topology Visualization	
ANDRES JUNIOR APARCANA TASAYCO 0 comments Add tag	
State	Done
Area	sdn-monitoring-system
Reason	Work finished
Iteration	sdn-monitoring-system
Description	Status
As a laboratory manager, I want to visualize the SDN network topology so that getting a visualization of the FIEE SDN laboratory	Start Date
	Target Date
Acceptance Criteria	Details
Click to add Acceptance Criteria	Priority
	2
	Effort
	20
	Business Value
	20
	Time Criticality
	Value area
	Business
Discussion	
Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.	

Nota. Pantalla capturada por el autor

Figura 26

Elaboración de la Característica del Sistema 2

FEATURE 4

4 FE-2 Detecting SDN network changes

ANDRES JUNIOR APARCANA TASAYCO 0 comments OpenDayLight X SDN X Topology X +

State **Done** Area sdn-monitoring-system
Reason **Work finished** Iteration sdn-monitoring-system

Description

As the head of the laboratory,
I want to visualize the changes in the SDN network topology
so that to monitor the status of the FIEE's SDN laboratory.

Acceptance Criteria

[Click to add Acceptance Criteria](#)

Discussion

AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Status

Start Date
Target Date

Details

Priority
2
Effort
20
Business Value
20
Time Criticality
Value area
Business

Nota. Pantalla capturada por el autor

Figura 27

Elaboración de la Característica del Sistema 3

FEATURE 43

43 FE-3 Restconf Statistics Monitoring

ANDRES JUNIOR APARCANA TASAYCO 0 comments Monitoring X OpenDayLight X Restconf X SDN X Statistics X +

State **Done** Area sdn-monitoring-system
Reason **Work finished** Iteration sdn-monitoring-system\Sprint 5

Description

As a Network Operator
I want to view statistics from the OpenDayLight SDN controller
so that I can monitoring the packets generated from the network

Acceptance Criteria

[Click to add Acceptance Criteria](#)

Discussion

AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Status

Start Date
26/07/2021 0:00
Target Date
06/08/2021 0:00

Details

Priority
1
Effort
40
Business Value
40
Time Criticality
10
Value area
Business

Nota. Pantalla capturada por el autor

Las características denominados FE-1, FE-2, FE-3 se deben descomponer en elementos (Historias de Usuario) manejables dentro del sprint que ayudan a estimar mejor las tareas que se deben realizar. Con ello, se elaboran las Historias de Usuario que debemos cumplir en la investigación y desarrollo del Sistema de Monitoreo de Red. Las figuras Figura 28, Figura 29, Figura 30, Figura 31, Figura 32, Figura 33, Figura 34, Figura 35, Figura 36, Figura 37 y Figura 38 muestran las Historias de Usuario utilizadas.

Figura 28

Historia de Usuario 1

The screenshot shows a Jira Product Backlog item with the following details:

- Item ID:** 5 HU-1 Laboratory SDN Topology Set Up
- Assignee:** ANDRES JUNIOR APARCANA TASAYCO
- Comments:** 0 comments
- Tags:** SDN, Set Up, Topology, VNRT
- State:** Done (indicated by a green dot)
- Area:** sdn-monitoring-system
- Reason:** Work finished
- Iteration:** sdn-monitoring-system\Sprint 2
- Description:** As a researcher, I want to configure the simulated environment of the FIEE SDN laboratory
- Acceptance Criteria:** Click to add Acceptance Criteria
- Discussion:** (Section header)
- Details:**
 - Priority: 1
 - Effort: 4
 - Business Value: 5
 - Value area: Business

Nota. Pantalla capturada por el autor

Figura 29

Historia de Usuario 2

PRODUCT BACKLOG ITEM 6

6 HU-2 Getting information about Ryu Controller

ANDRES JUNIOR APARCANA TASAYCO 0 comments Information X Ryu X SDN X +

State ● Done Area sdn-monitoring-system
Reason 🔒 Work finished Iteration sdn-monitoring-system\Sprint 1

Description	Details
Me, As a researcher I want to get information about Ryu Controller so that I can understand how it works	Priority 1
Acceptance Criteria	Effort 🔒 2
Given	Business Value 🔒 6
When	Value area
Then	Business

Nota. Pantalla capturada por el autor

Figura 30

Historia de Usuario 3

PRODUCT BACKLOG ITEM 27

27 HU-3 Topology Display SDN

ANDRES JUNIOR APARCANA TASAYCO 0 comments Add tag

State ● Commit... Area sdn-monitoring-system
Reason 🔒 Commitment ... Iteration sdn-monitoring-system\Sprint 3

Description	Details
As head of laboratory I want to visualize SDN topology on screen so that I can see nodes and links between them	Priority 2
Acceptance Criteria	Effort 20
<i>Click to add Acceptance Criteria</i>	Business Value 14
	Value area
	Business

Nota. Pantalla capturada por el autor

Figura 31

Historia de Usuario 4

PRODUCT BACKLOG ITEM 8

8 HU-4 Detecting SDN Network Changes

ANDRES JUNIOR APARCANA TASAYCO 0 comments Add tag

State Commitment... Area sdn-monitoring-system
Reason Commitment ... Iteration sdn-monitoring-system\Sprint 3

Description	Details
<p>As head of laboratory I want viewing network state changes like nodes up/down, links up/down, stats from each port of the nodes so that I can monitoring the network state changes</p>	<p>Priority 2</p> <p>Effort 15</p> <p>Business Value 10</p> <p>Value area Business</p>
<p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p>	

Nota. Pantalla capturada por el autor

Figura 32

Historia de Usuario 5

PRODUCT BACKLOG ITEM 7

7 HU-5 Viewing SDN topology API

ANDRES JUNIOR APARCANA TASAYCO 0 comments Add tag

State Commitment... Area sdn-monitoring-system
Reason Additional wo... Iteration sdn-monitoring-system\Sprint 2

Description	Details
<p>As head of laboratory I want viewing SDN topology from a API so that I can see key-value format of the nodes</p>	<p>Priority 2</p> <p>Effort 10</p> <p>Business Value 16</p> <p>Value area Business</p>
<p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p>	

Nota. Pantalla capturada por el autor

Figura 33

Historia de Usuario 6

[PRODUCT BACKLOG ITEM 18](#)

18 HU-6 Understanding LLDP protocol

AT ANDRES JUNIOR APARCANA TASAYCO 0 comments [Add tag](#)

State	● Done	Area	sdn-monitoring-system
Reason	🔒 Work finished	Iteration	sdn-monitoring-system\Sprint 1

Description

As a researcher,
I want to know about LLDP functionality on legacy and SDN implementation
so that I can understand LLDP works on both.

Acceptance Criteria

Click to add Acceptance Criteria

Details

Priority
2

Effort
🔒

Business Value
🔒

Value area
Business

Nota. Pantalla capturada por el autor

Figura 34

Historia de Usuario 7

[PRODUCT BACKLOG ITEM 34](#)

34 HU-7 Real-Time SDN topology

AT ANDRES JUNIOR APARCANA TASAYCO 0 comments [Add tag](#)

State	● Done	Area	sdn-monitoring-system
Reason	🔒 Work finished	Iteration	sdn-monitoring-system\Sprint 4

Description

As a Laboratory Chief
I want to get SDN topology change each 5 seconds
so that SDN topology can show a graph of changes without recharge of the page

Acceptance Criteria

Click to add Acceptance Criteria

Discussion

AT *Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.*

Details

Priority
2

Effort
🔒 20

Business Value
🔒 12

Value area
Business

Nota. Pantalla capturada por el autor

Figura 35

Historia de Usuario 8

PRODUCT BACKLOG ITEM 38

38 HU-8 Alert Message

ANDRES JUNIOR APARCANA TASAYCO 0 comments [Add tag](#)

State ● Done Area sdn-monitoring-system
Reason Work finished Iteration sdn-monitoring-system\Sprint 4

Description	Details
<p>As a Laboratory Chief I want to get Alert Message of nodes and links changes so that App can show node and links changes</p> <p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p> <p>Discussion</p> <p><i>Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.</i></p>	<p>Priority 2</p> <p>Effort 10</p> <p>Business Value 15</p> <p>Value area Business</p>

Nota. Pantalla capturada por el autor

Figura 36

Historia de Usuario 9

PRODUCT BACKLOG ITEM 45

45 HU-9 Statistics Visualization

ANDRES JUNIOR APARCANA TASAYCO 0 comments [NeXt-UI](#) [SDN](#) [Statistics](#) [Visualization](#)

State ● Done Area sdn-monitoring-system
Reason Work finished Iteration sdn-monitoring-system\Sprint 5

Description	Details
<p>As a Network Operator I want to view OpenFlow Statistics on the screen</p> <p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p> <p>Discussion</p> <p><i>Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.</i></p>	<p>Priority 2</p> <p>Effort 10</p> <p>Business Value 10</p> <p>Value area Business</p>

Nota. Pantalla capturada por el autor

Figura 37

Historia de Usuario 10

PRODUCT BACKLOG ITEM 46

46 HU-10 Features Node Visualization

ANDRES JUNIOR APARCANA TASAYCO 0 comments Features X NeXt-UI X Nodes X SDN X Visual

Status: Done Area: sdn-monitoring-system
Reason: Work finished Iteration: sdn-monitoring-system\Sprint 5

Description	Details
<p>As a Network Operator I want to view OpenFlow features from node on the screen</p>	<p>Priority 2</p> <p>Effort 10</p> <p>Business Value 10</p> <p>Value area Business</p>
<p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p>	
<p>Discussion</p> <p>AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.</p>	

Nota. Pantalla capturada por el autor

Figura 38

Historia de Usuario 11

PRODUCT BACKLOG ITEM 44

44 HU-11 Restconf Statistics Collector

ANDRES JUNIOR APARCANA TASAYCO 0 comments Collect X OpenDayLight X Restconf X SDN X

Status: Done Area: sdn-monitoring-system
Reason: Work finished Iteration: sdn-monitoring-system\Sprint 5

Description	Details
<p>As a Network Operator I want to collect statistics from OpenDayLight SDN controller so that I can save and query them</p>	<p>Priority 1</p> <p>Effort 20</p> <p>Business Value 20</p> <p>Value area Business</p>
<p>Acceptance Criteria</p> <p><i>Click to add Acceptance Criteria</i></p>	
<p>Discussion</p> <p>AT Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.</p>	

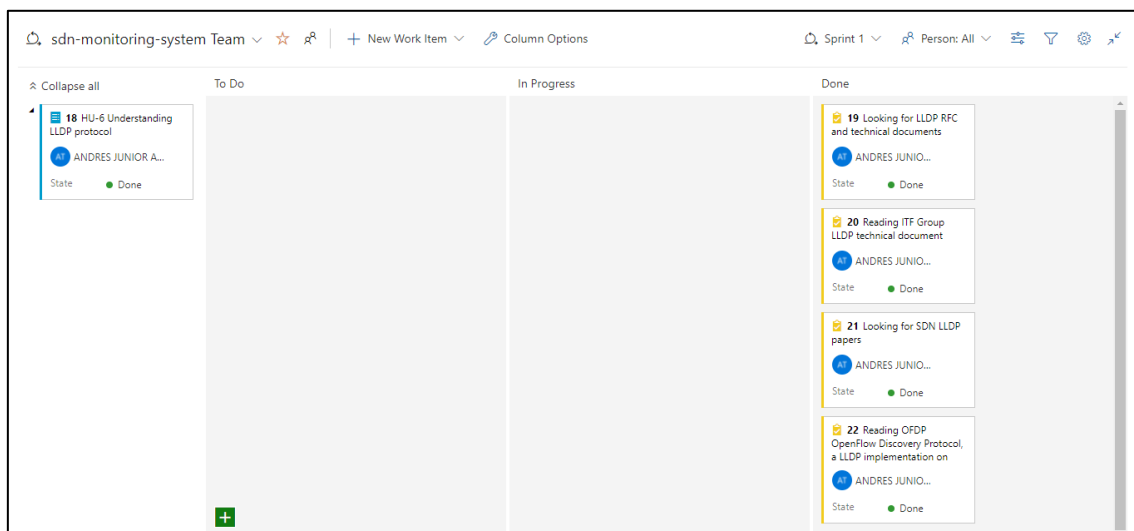
Nota. Pantalla capturada por el autor

Sprint 1

Aquí se presentan los resultados del primer sprint. Este tiene el objetivo principal de conocer y entender lo que se va a investigar. Por ende, se hace una revisión de literatura sobre los componentes que se gestionan en el desarrollo. El Scrumboard elaborado para dar seguimiento a las tareas relacionadas a las HU-6 y HU-2 en la Figura 39 y Figura 40.

Figura 39

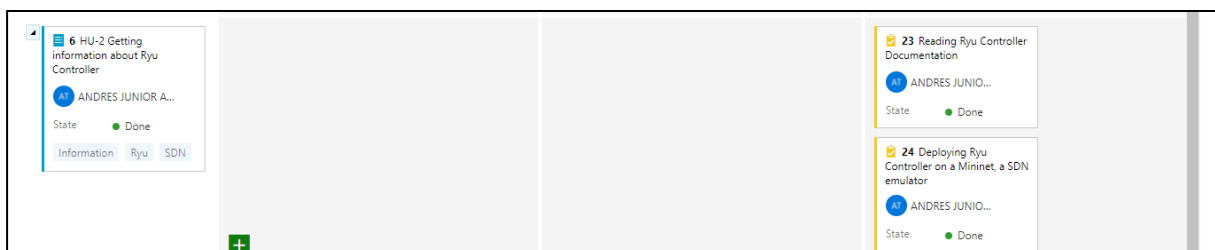
Scrumboard Sprint 1-1



Nota. Pantalla capturada por el autor

Figura 40

Scrumboard Sprint 1-2



Nota. Pantalla capturada por el autor

Como resultado de este sprint 1 se mostró diagramas de explicación de la teoría sobre LLDP y el monitoreo de Redes Definidas por Software. La

Figura 41 y Figura 42 presentan capturas de estos resultados.

Figura 41

Diagramas de explicación de protocolo LLDP y monitoreo en Redes Definidas por Software

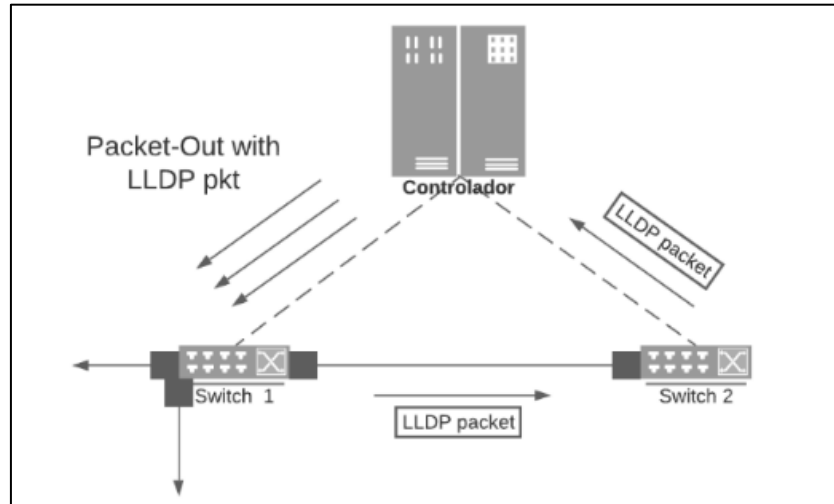
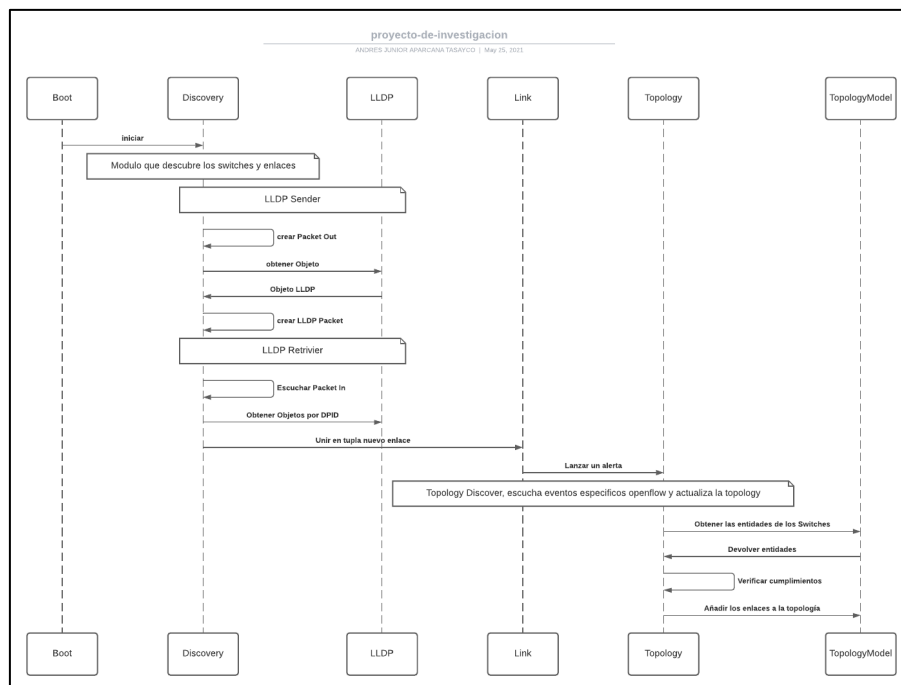


Figura 42

Diagrama de actividades del monitoreo de red en controlador POX

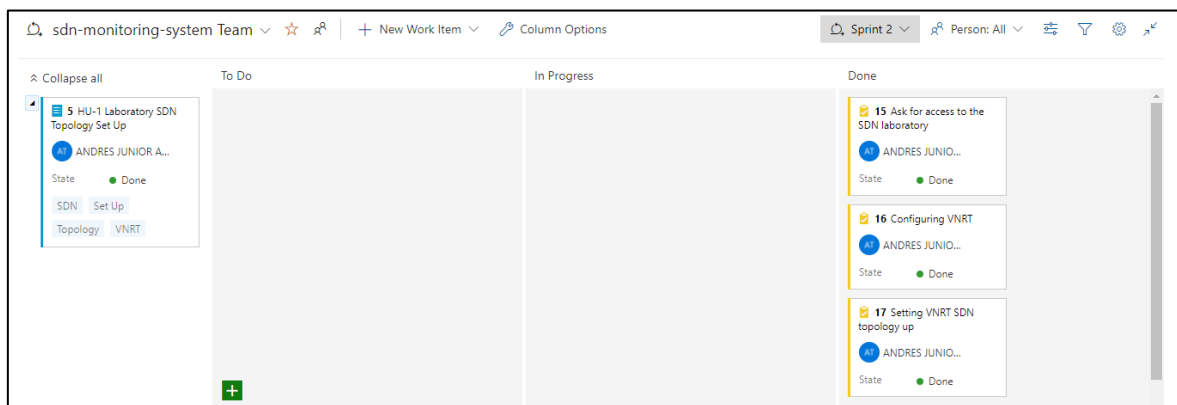


Sprint 2

El objetivo de este sprint es principalmente comenzar el desarrollo de los componentes técnicos en back-end. La base del proyecto se encuentra en los más altos niveles de valor. El Scrumboard elaborado para dar seguimiento a las tareas relacionadas a las HU-1 y HU-5 en la Figura 43 y Figura 44.

Figura 43

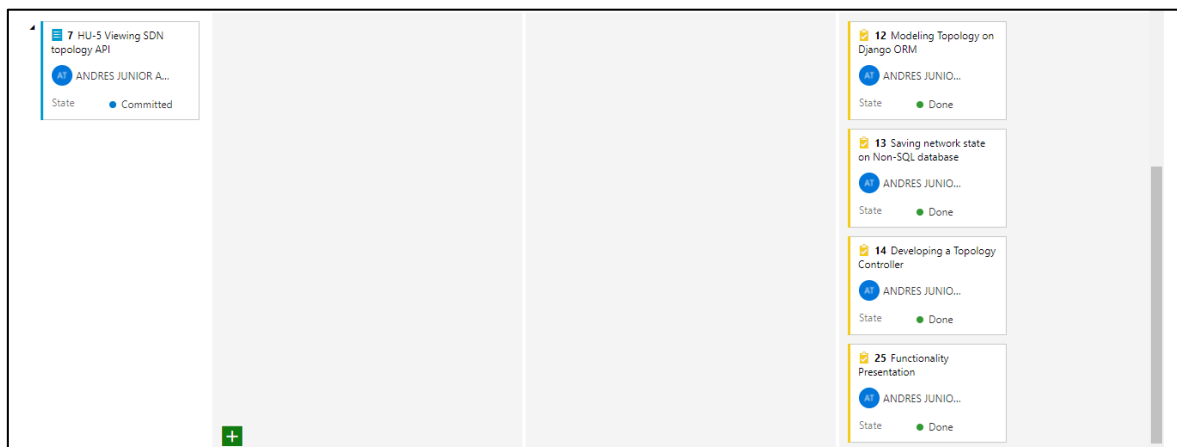
Scrumboard Sprint 2-1



Nota. Pantalla capturada por el autor

Figura 44

Scrumboard Sprint 2-2

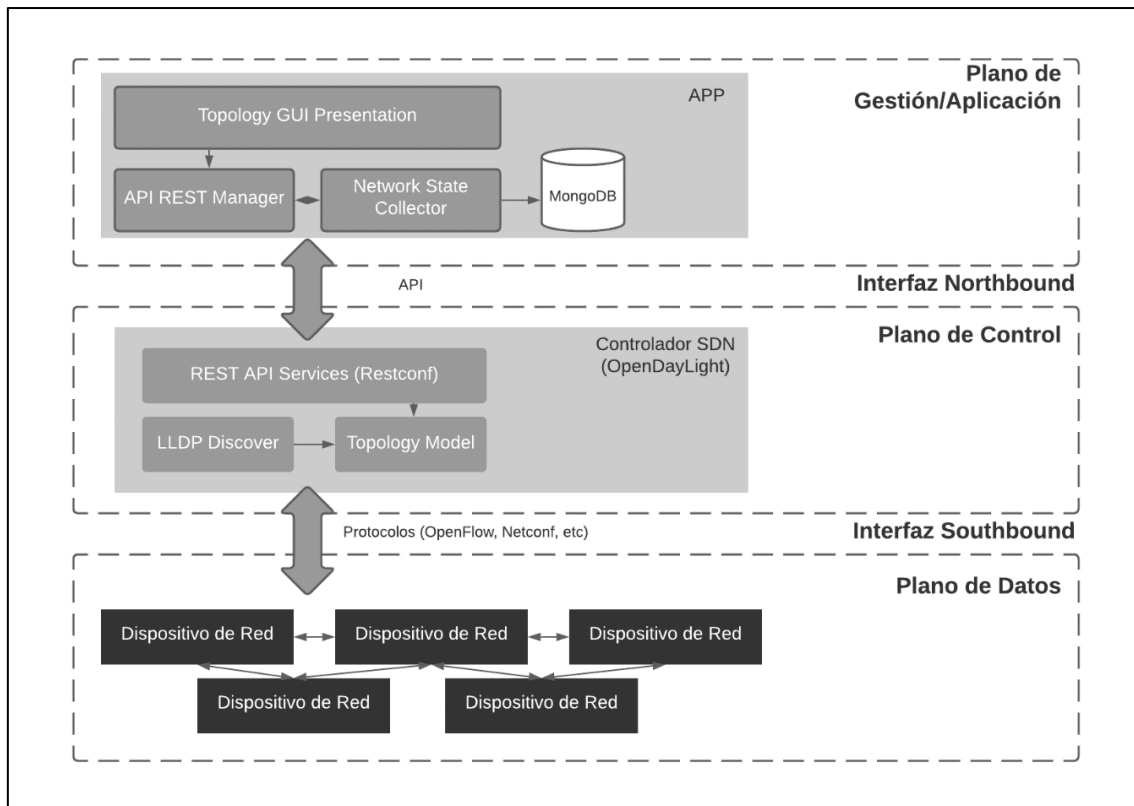


Nota. Pantalla capturada por el autor

Cómo productos del Sprint 2 se obtuvo la base del proyecto (back-end) funcionando en el framework Django, escrito en Python. Con ello mostramos la arquitectura del sistema simplificada en el paradigma de Redes Definidas por Software en la Figura 45.

Figura 45

Arquitectura del Sistema simplificada bajo el paradigma SDN



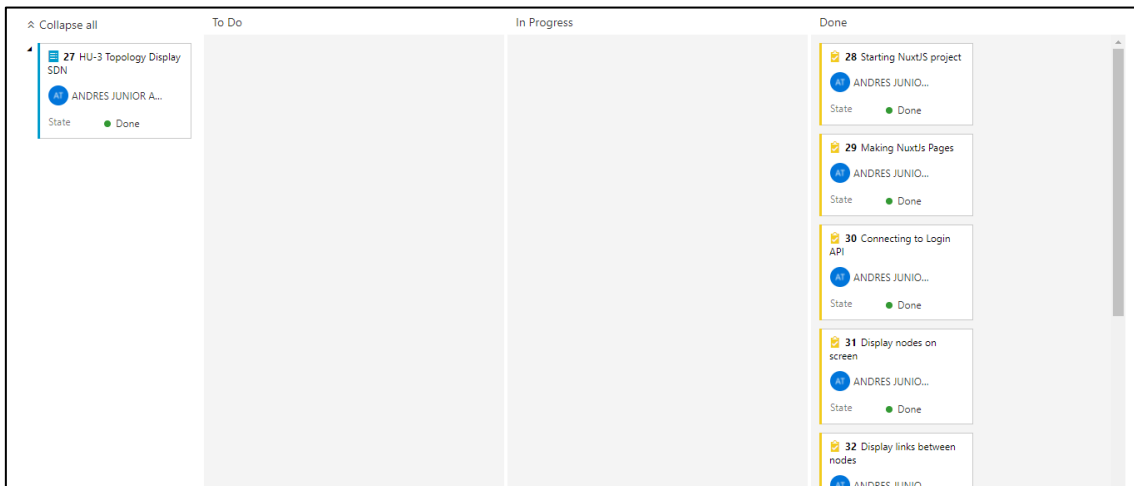
Este se encarga de procesar las peticiones HTTP, y reaccionar entregando la topología actual de la red.

Sprint 3

El objetivo de este sprint es principalmente comenzar el desarrollo de los componentes técnicos en front-end de parte del cliente. La Aplicación cliente es la que hace las peticiones HTTP hacia el servidor. A continuación, se presentan las historias y tareas relacionadas que se enmarcan en el sprint. El Scrumboard elaborado para dar seguimiento a las tareas relacionadas a las HU-3, HU-4 y HU-5 en la Figura 46 y Figura 47.

Figura 46

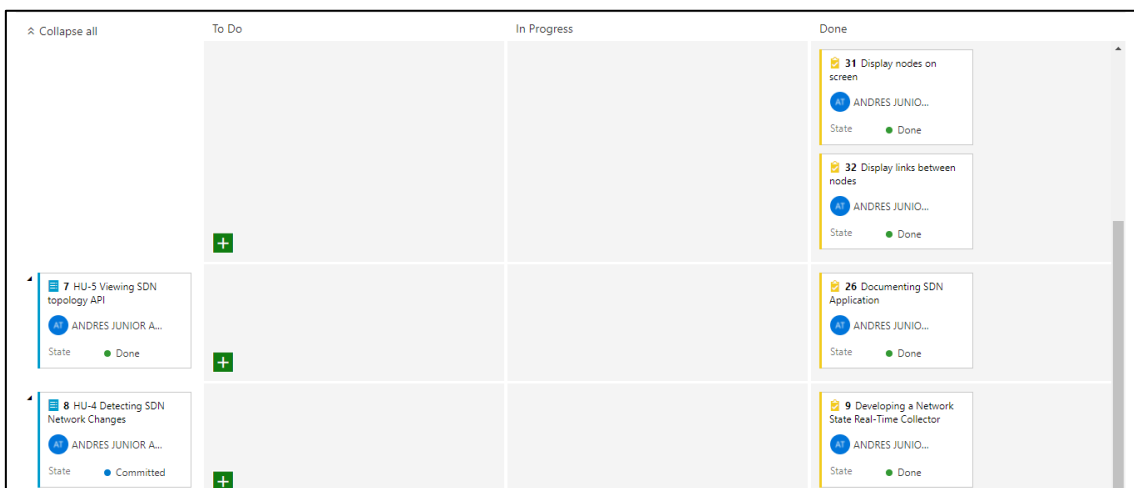
Scrumboard Sprint 3-1



Nota. Pantalla capturada por el autor

Figura 47

Scrumboard Sprint 3-2

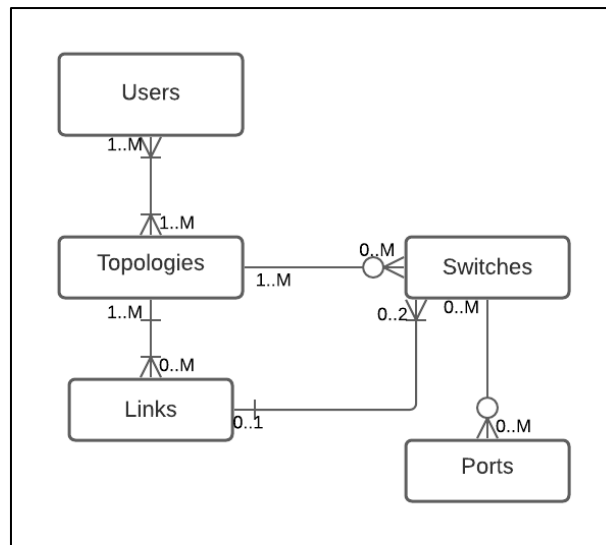


Nota. Pantalla capturada por el autor

También se presenta como producto de este sprint el modelo de base de datos que se utiliza para monitorear la red. Este modelo está basado en un diagrama de clases debido a que la base de datos utilizada es no relacional. Sin embargo, cumple ciertas reglas lógicas, las cuáles son descritas en la Figura 48.

Figura 48

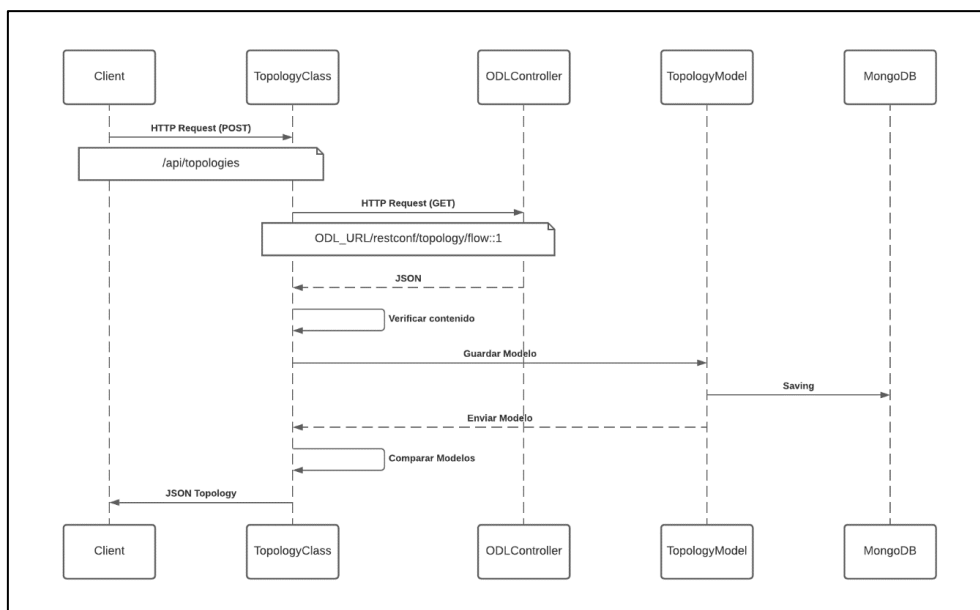
Modelo de base de datos no relacional



En la Figura 49 se describen las interacciones entre la Aplicación de Monitoreo de Red y el Controlador SDN. El intercambio de datos es importante para asegurar un correcto funcionamiento del sistema propuesto.

Figura 49

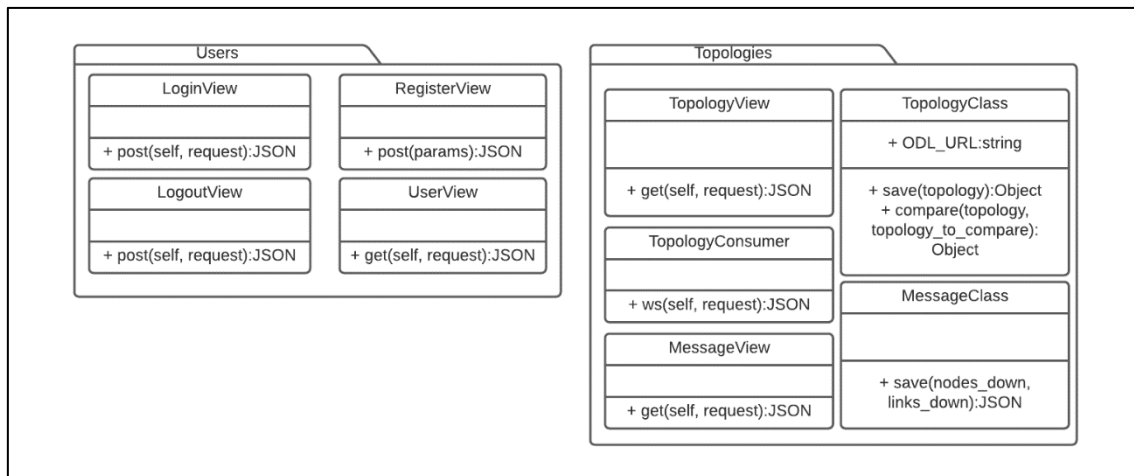
Diagrama de Actividad entre aplicación Servidor y Controlador SDN



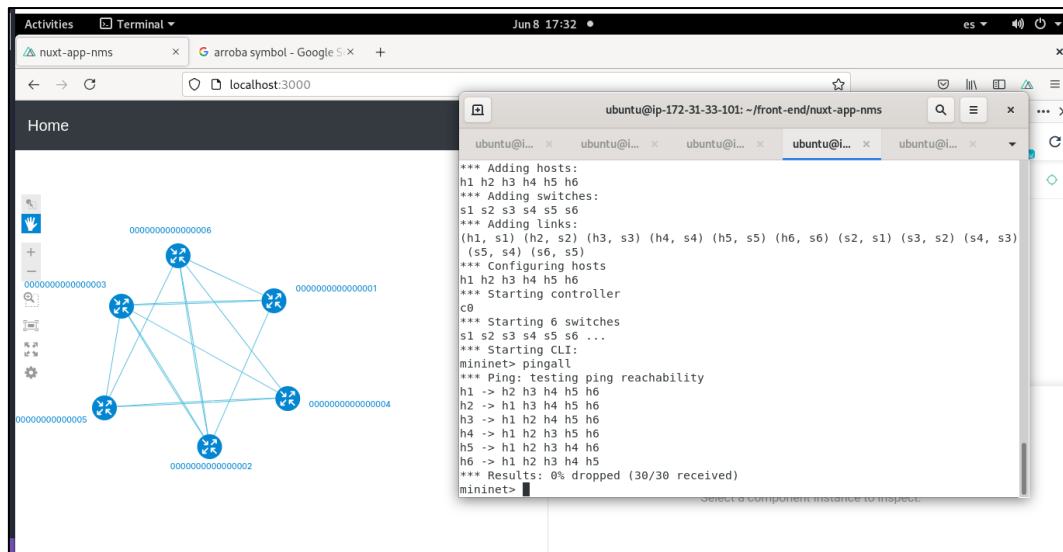
En la Figura 50 se describe los paquetes que forman parte de la Aplicación de Monitoreo de Red del lado del Servidor. Estos paquetes y clases interactúan para lograr un funcionamiento de consistencia de la Red.

Figura 50

Diagrama de Paquetes de la Aplicación Back-end

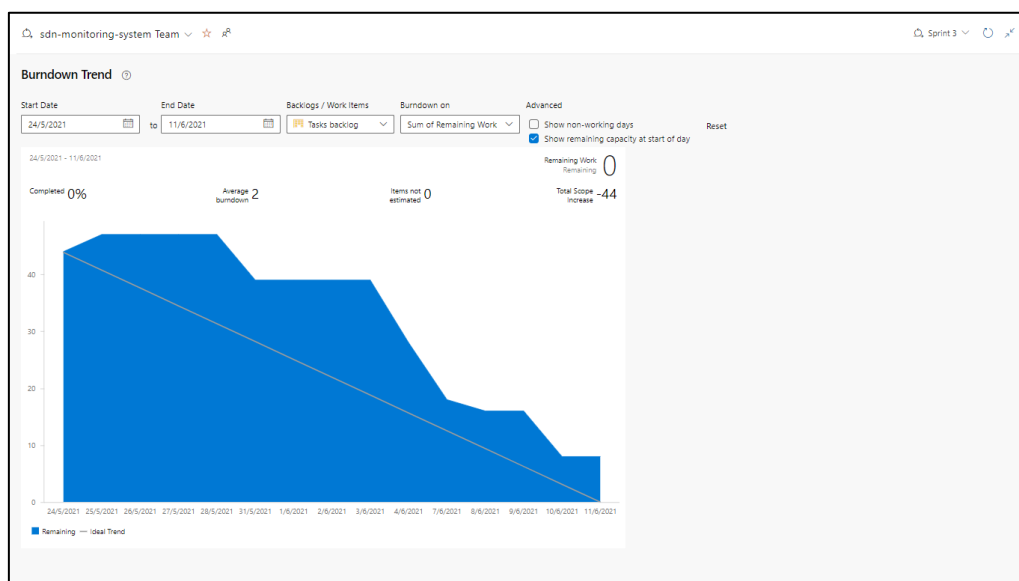


Finalmente, la Figura 51 muestra la primera interfaz de monitoreo. Esta aplicación grafica las topologías existentes en la red. Las primeras pruebas del segundo prototipo fueron realizadas en un emulador de Red SDN que crea los switches y un controlador real SDN que los gestiona para posteriormente colocar a disposición de la Aplicación de Monitoreo de Red el estado actual de la red.

Figura 51**Segundo Prototipo Funcional del Sistema de Monitoreo de Red SDN**

Nota. Pantalla capturada por el autor

Durante las fechas 24 de mayo hasta 11 de junio del 2021 se desarrolló el Sprint 3. Las tareas en este Sprint fueron culminadas como las más importantes pues constituyen el núcleo del proyecto. El Burdown Chart para este Sprint es mostrado en la Figura 52.

Figura 52**Burndown Chart del Sprint 3**

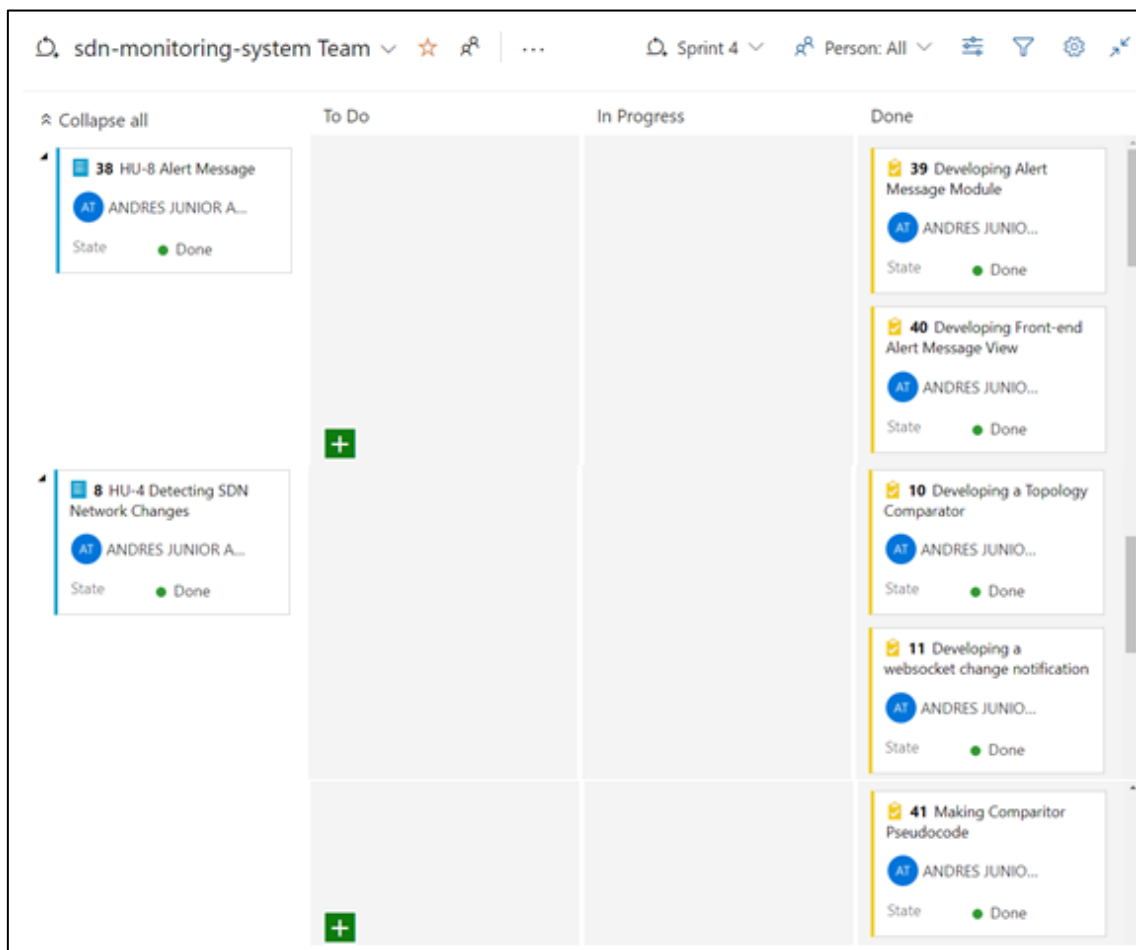
Nota. Pantalla capturada por el autor

Sprint 4

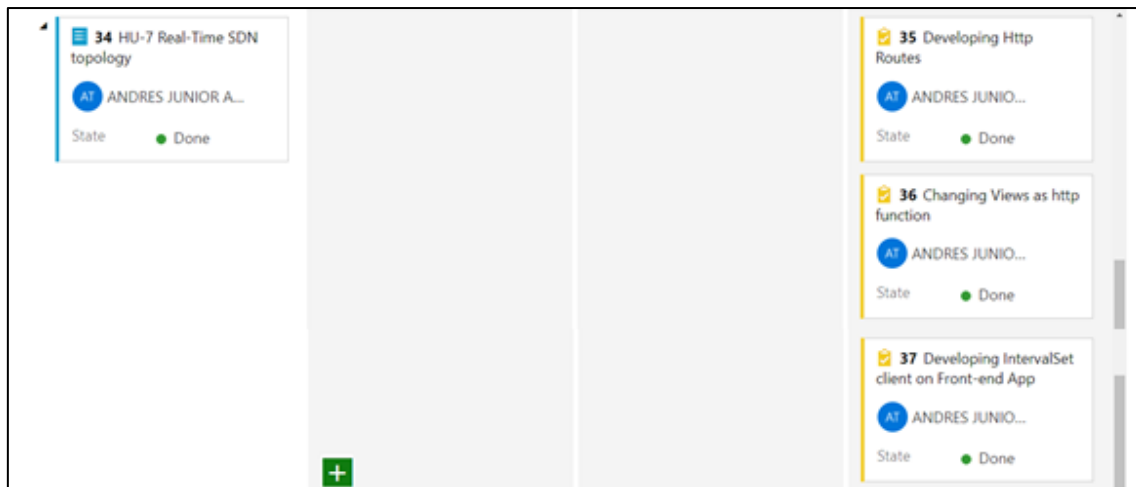
El sprint 4 es creado como predecesor y solucionador de fallos, así también integra funcionalidades visuales al Sistema de Monitoreo de Red. Estas nuevas funciones se describen en Figura 53 y Figura 54.

Figura 53

Scrumboard Sprint 4-1

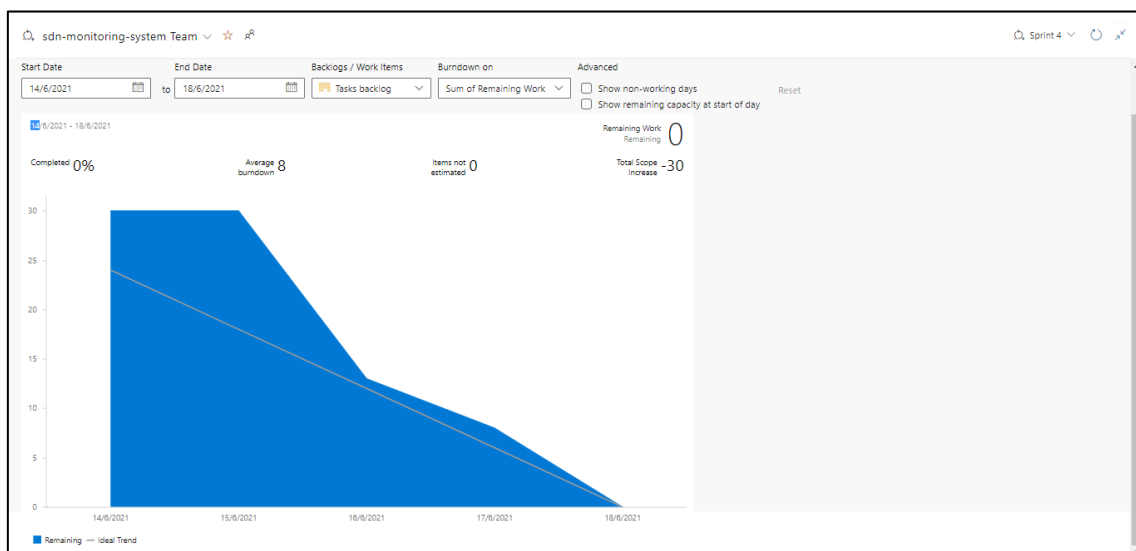


Nota. Pantalla capturada por el autor

Figura 54*Scrumboard Sprint 4-2*

Nota. Pantalla capturada por el autor

En la Figura 55 se muestra el Burndown Chart, este gráfico representa el trabajo hecho en el Sprint 4 durante el 14 hasta el 18 de junio del 2021.

Figura 55*Burndown Chart del Sprint 4*

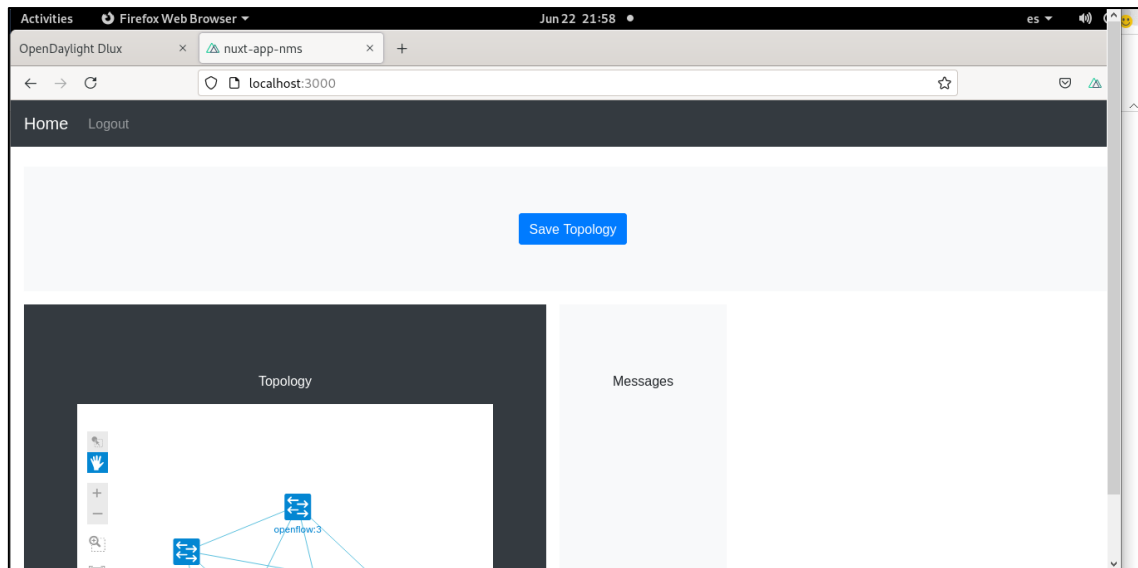
Nota. Pantalla capturada por el autor

El producto de este sprint es finalmente el Sistema de Monitoreo de Red desarrollado para el EPIC-1 como se muestra en la

Figura 56.

Figura 56

Pantalla del Sistema de Monitoreo de Red SDN para el Sprint 4

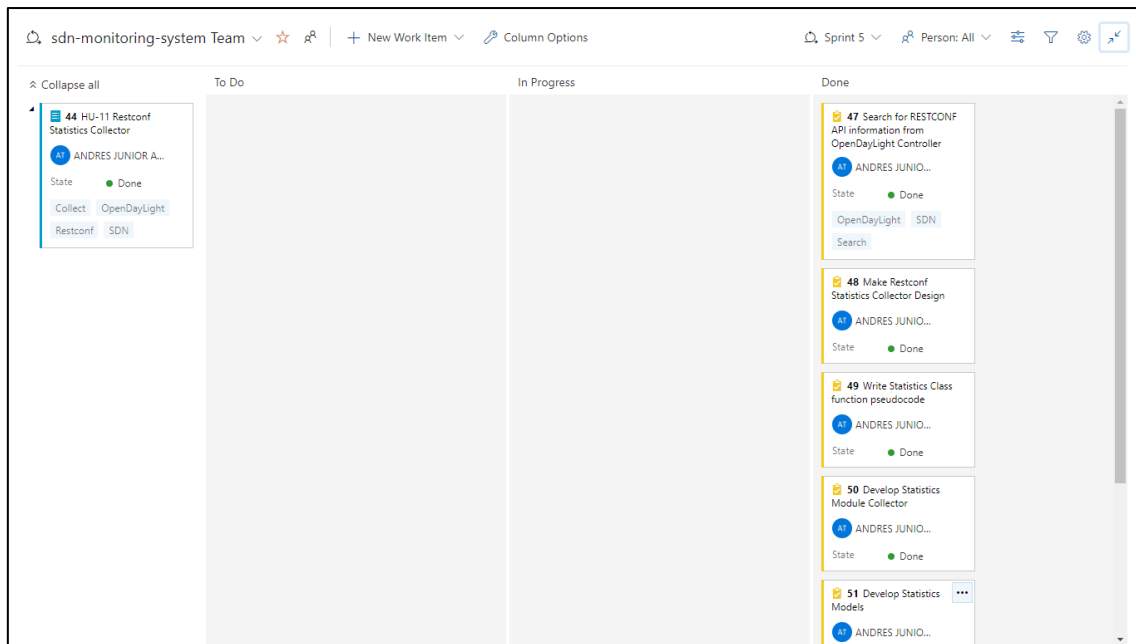


Nota. Pantalla capturada por el autor

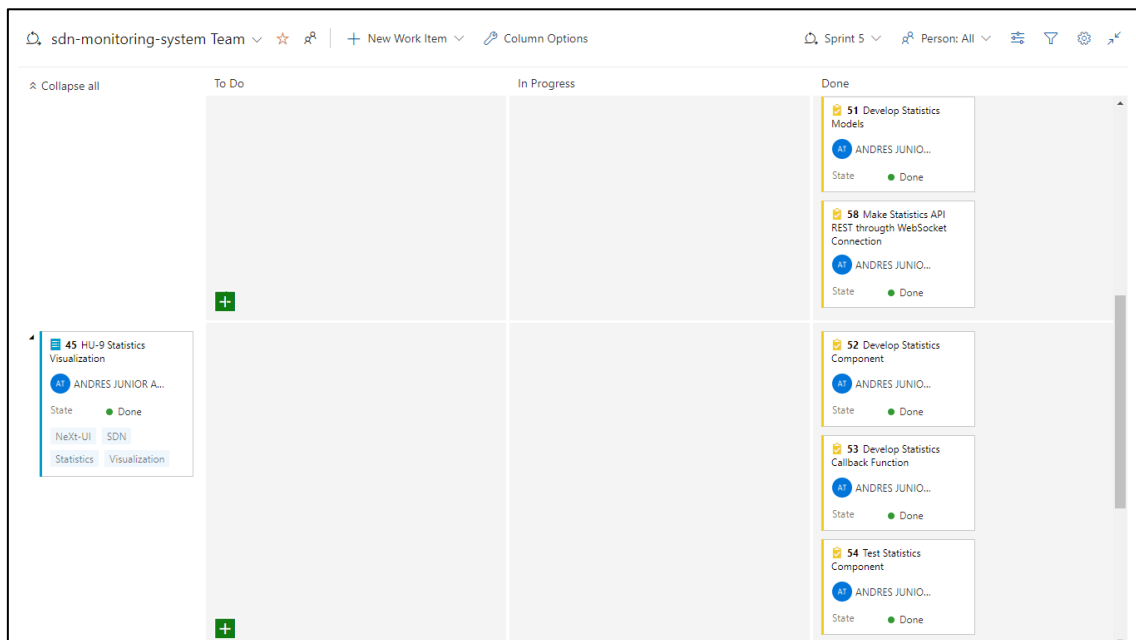
Sprint 5

El sprint 5 es tiene como objetivo el rediseño de la pantalla principal del sistema de monitoreo y el desarrollo de un módulo de estadísticas de monitoreo de la red. Las figuras Figura 57 y Figura 58 y

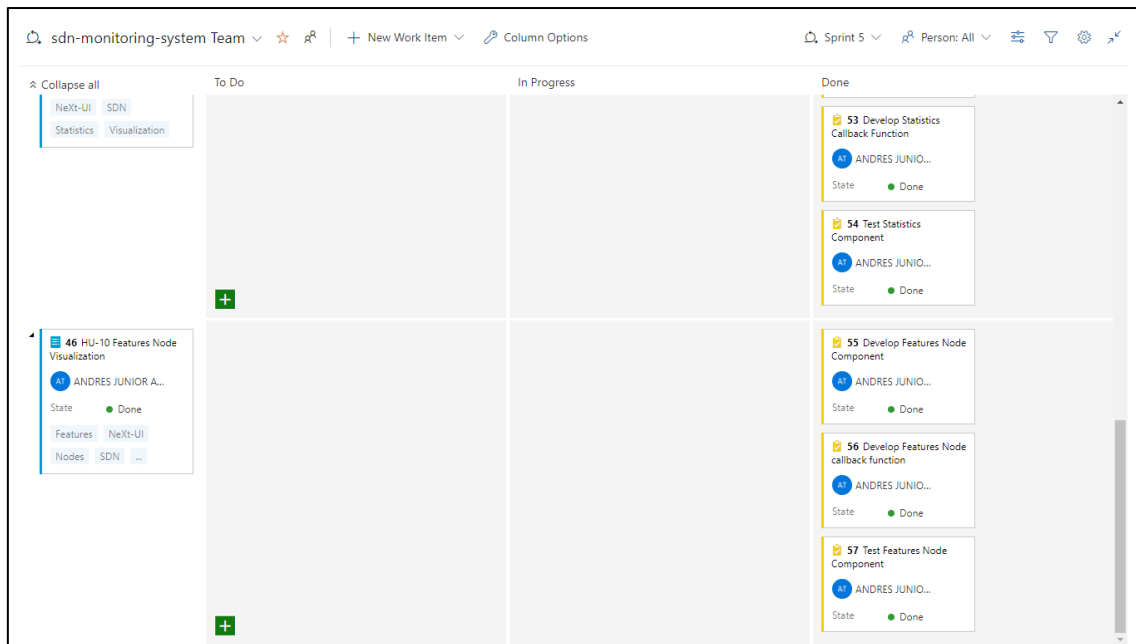
Figura 59 muestran el Scrumboard del Sprint 5 asociando las Historias de Usuario 9, 10, 11 con las tareas que se desarrollaron para culminar el Sprint.

Figura 57**Scrumboard Sprint 5-1**

Nota. Pantalla capturada por el autor

Figura 58**Scrumboard Sprint 5-2**

Nota. Pantalla capturada por el autor

Figura 59**Scrumboard Sprint 5-3**

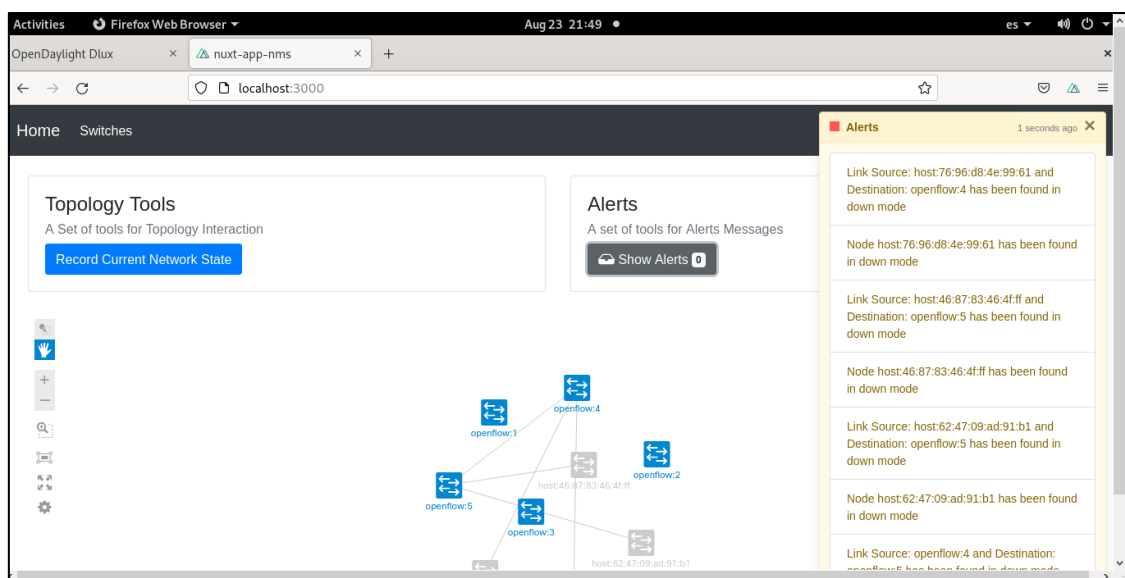
Nota. Pantalla capturada por el autor

Los resultados de del Sprint 5 se pueden visualizar en las siguientes figuras.

La Figura 60 muestra la pantalla principal y la detección de enlaces caídos.

Figura 60

Pantalla final del sistema de monitoreo de red-NMS. Vista principal y detección de enlaces caídos

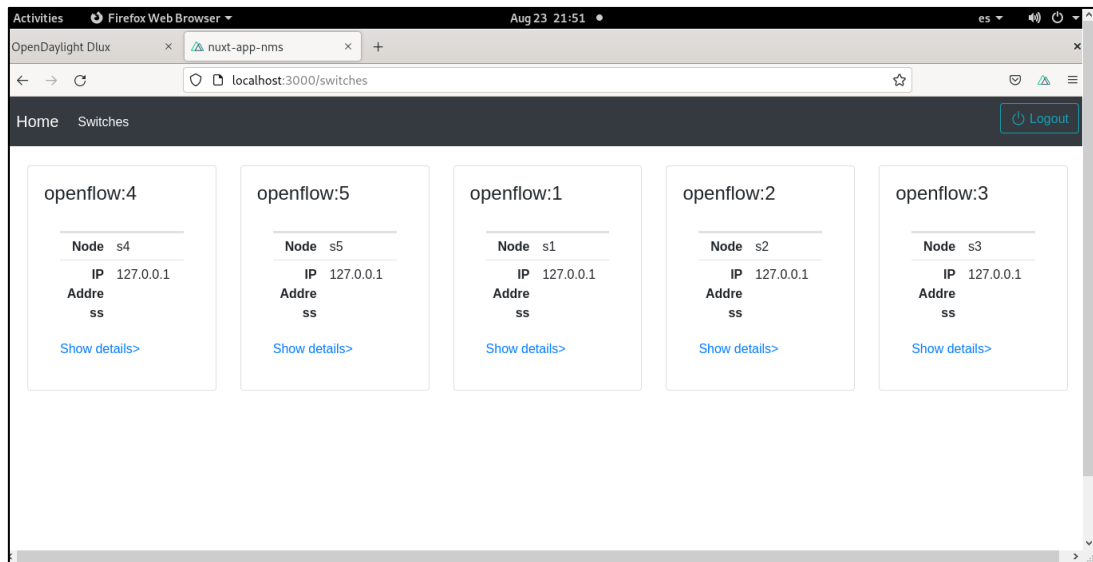


Nota. Pantalla capturada por el autor

La Figura 61 muestra la pantalla para el listado de los switches de la red. La opción “show details” le permite al administrador de la red ingresar al monitoreo de estadísticas de cada switch.

Figura 61

Pantalla final del sistema de monitoreo de red-NMS. Vista de switches



Nota. Pantalla capturada por el autor

Las figuras Figura 62, Figura 63 y Figura 64 muestran las opciones de detalles dentro de los switches y las estadísticas correspondientes a los puertos del switch y de sus tablas de flujo.

Figura 62

Pantalla final del sistema de monitoreo de red-NMS. Vista del Switch 5

The screenshot shows the OpenDaylight Dlux web interface. The browser address bar indicates the URL is localhost:3000/switches/openflow:5. The page title is 'Home Switches' and there is a 'Logout' button in the top right corner. The main content area is divided into two sections. On the left, there is a 'Node Information' table for the switch 'openflow:5' (IP: 127.0.0.1). On the right, there is a vertical list of six port buttons labeled 'Port openflow:5:5' through 'Port openflow:5:4'. At the bottom of the page, a table header 'Table 0' is visible.

Node Information	
ID	openflow:5
Description	s5
Serial Number	None
Hardware	Open vSwitch
Software	2.13.1
Manufacturer	Nicira, Inc.
IP Address	127.0.0.1
Snapshot starts	8/23/2021, 21:51:42
Snapshot ends	8/23/2021, 21:51:42 <input checked="" type="checkbox"/>

Nota. Pantalla capturada por el autor

Figura 63

Pantalla final del sistema de monitoreo de red-NMS. Vista del puerto Openflow:5 del Switch 5

The screenshot shows the OpenDaylight Dlux web interface, specifically the configuration page for port 'openflow:5:5'. The browser address bar indicates the URL is localhost:3000/switches/openflow:5. The page title is 'Home Switches' and there is a 'Logout' button in the top right corner. The main content area is divided into three sections. On the left, there is a 'Node Information' table for the switch 'openflow:5' (IP: 127.0.0.1). In the center, there is a 'Port Information' table for the port 'openflow:5:5'. On the right, there is a 'Statistics Port Information' table showing receive and transmit statistics.

Node Information	
ID	openflow:5
Description	s5
Serial Number	None
Hardware	Open vSwitch
Software	2.13.1
Manufacturer	Nicira, Inc.
IP Address	127.0.0.1
Snapshot starts	8/23/2021, 21:51:36
Snapshot ends	8/23/2021, 21:51:36 <input checked="" type="checkbox"/>

Port Information	
ID	openflow:5:5
Port Number	5
Hardware Address	aa:0d:9c:ca:67:d0
Name	s5-eth5
State Blocked	false <input type="checkbox"/>
State Link-down	true <input checked="" type="checkbox"/>
State Live	false <input type="checkbox"/>

Statistics Port Information	
Receive frame error	0
Packets Received	186257
Packets Transmitted	187392
Bytes Received	15833740
Bytes Transmitted	15918851
Duration (ns)	93000000
Duration (s)	3093008

Nota. Pantalla capturada por el autor

Figura 64

Pantalla final del sistema de monitoreo de red-NMS. Vista de las tablas de flujos del Switch

5

The screenshot shows a web browser window with the URL `localhost:3000/switches/openflow:5`. The page title is 'Table 0'. Below the title, there is a section for 'Table Information' and 'Flow table statistics'. A table shows the following data:

ID	Active Flows	Packets looked-up	Packets matched
0	7	772694	772691

Below this table is a 'See Flows' button. Further down, there is a 'Flow Statistics Information' section with a table for 'Flow Information' and 'Flow Statistics'.

Flow Information		Flow Statistics									
ID	Priority	Packet count	Byte count	Duration (ns)	Duration (s)	Table ID	Cookie mask	Hard timeout	Match in port	Cookie	Idle Flags timeout
L2switch-2519	2	880	75932	211000000	336366	0	0	0	5	3098476543630904000	0

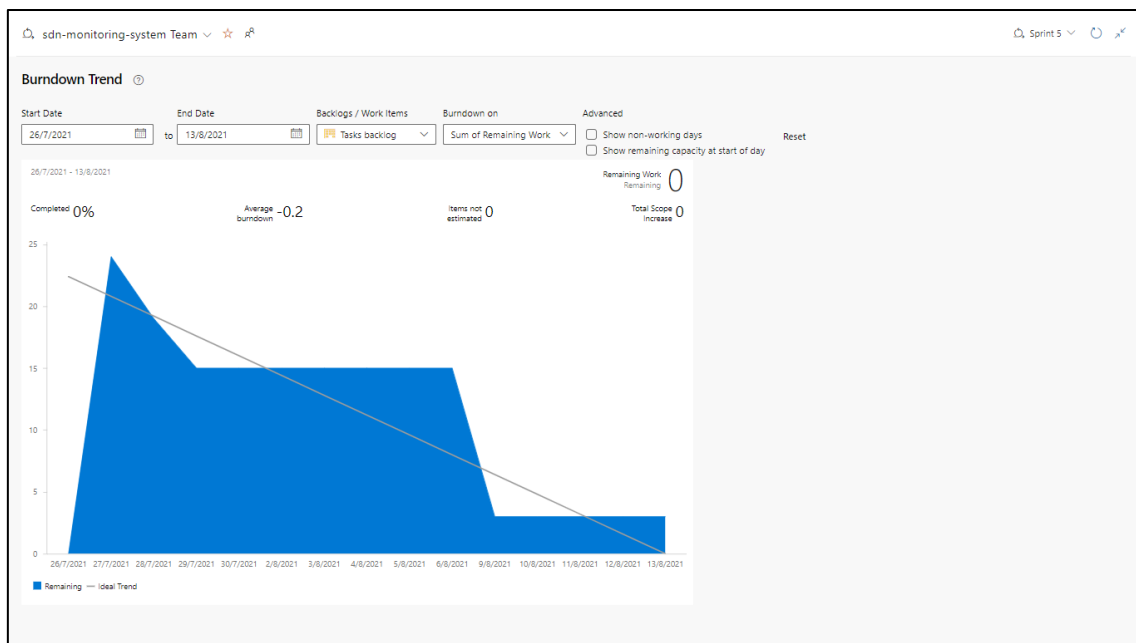
There is also a 'See Instructions' button at the bottom left of the flow statistics table.

Nota. Pantalla capturada por el autor

El Burndown Chart del Sprint 5 se muestra en el Figura 65 el trabajo realizado durante las fechas 26 de julio a 13 de agosto del 2021.

Figura 65

Burndown Chart del Sprint 5



Nota. Pantalla capturada por el autor

CAPÍTULO III
RESULTADOS

Resultados descriptivos de las dimensiones con la variable

Análisis descriptivo del indicador de tiempo de visualización de topología

Tabla 17

Frecuencia del indicador de tiempo de visualización de topología

		Tiempo de Visualización de Topología Preprueba	Tiempo de Visualización de Topología Postprueba
N	Válido	30	30
	Perdidos	0	0
Media		1021.8993	1827.9293
Error estándar de la media		133.99343	202.10222
Mediana		953.8250	1863.3500
Moda		.05 ^a	54.34 ^a
Desv. Desviación		733.91226	1106.95945
Varianza		538627.210	1225359.225
Asimetría		.201	.135
Error estándar de asimetría		.427	.427
Curtosis		-1.252	-1.005
Error estándar de curtosis		.833	.833
Rango		2288.59	3902.59
Mínimo		.05	54.34
Máximo		2288.64	3956.93
Suma		30656.98	54837.88
Percentiles	25	336.5300	876.7475
	50	953.8250	1863.3500
	75	1674.1925	2794.9600

Nota. Elaborado con el Software SPSS Versión 25

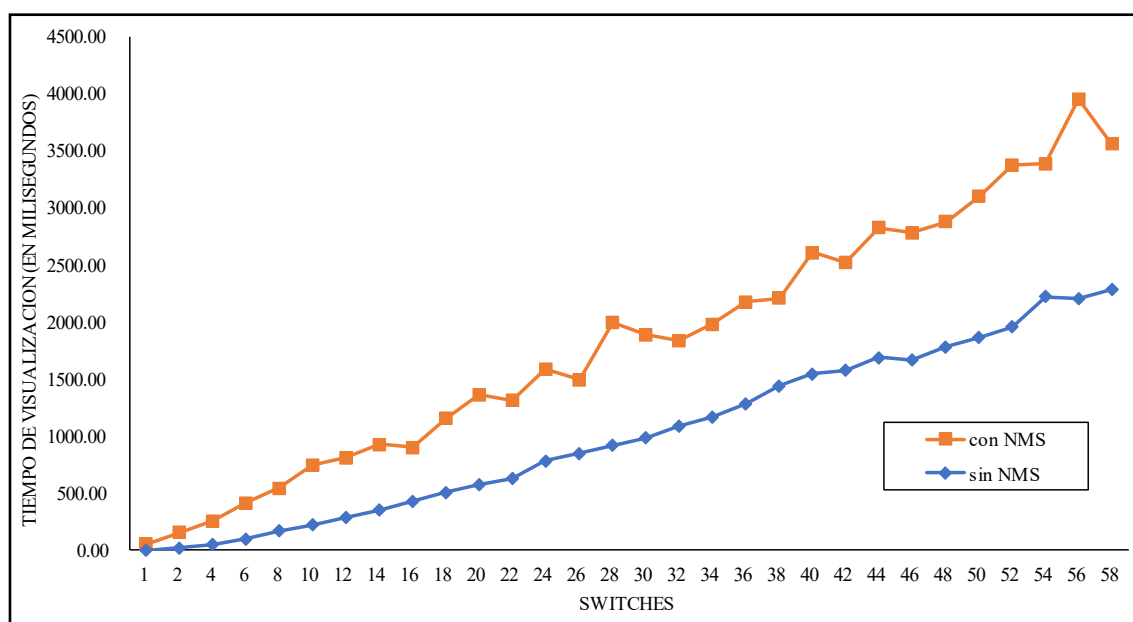
a. Existen múltiples modos. Se muestra el valor más pequeño.

Según los datos presentados en la Tabla 17, el tiempo de visualización de la topología en la preprueba tuvo un promedio de 1021,8993 ms, mientras que en la postprueba fue de 1827,9293 ms. Estos resultados indican un aumento de alrededor del 79%, lo que sugiere que el tiempo de visualización de la topología ha

incrementado con el uso del NMS propuesto. En la Figura 66 muestra los datos tabulados del tiempo de respuesta de visualización de la topología y las diferentes configuraciones de switches. Se puede visualizar que el tiempo de respuesta sin el NMS, muestra un aumento con respecto al número de switches, en promedio, el porcentaje de 1000ms. En mismo sentido, el tiempo de respuesta con el NMS aumenta con respecto al número de switches y en todas las pruebas es mayor que la evaluación sin el NMS. El tiempo de respuesta del componente web para la visualización de la topología, en promedio, es 1500ms, donde la configuración inicial arroja 50ms y la configuración final 3500ms.

Figura 66

Tiempo de respuesta del componente topología con respecto al número de switches



En la figura anterior se puede visualizar cambios significativos en el tiempo de visualización de la topología con el NMS en comparación sin el NMS, por lo que se puede afirmar que el NMS afecta significativamente la visualización de la topología, aumentando el tiempo de respuesta.

Análisis descriptivo del indicador de cantidad de uso de memoria

Tabla 18

Frecuencia del indicador de cantidad de uso de memoria

		Cantidad de Uso de Memoria Preprueba	Cantidad de Uso de Memoria Postprueba
N	Válido	30	30
	Perdidos	0	0
Media		2542892.3350	4316567.0180
Error estándar de la media		66740.31321	212033.02314
Mediana		2567381.8900	4572714.9950
Moda		1908315.43 ^a	2710866.90 ^a
Desv. Desviación		365551.75040	1161352.69712
Varianza		133628082220.285	1348740087111.714
Asimetría		-.441	-.231
Error estándar de asimetría		.427	.427
Curtosis		-1.140	-1.708
Error estándar de curtosis		.833	.833
Rango		1090034.26	3093508.96
Mínimo		1908315.43	2710866.90
Máximo		2998349.69	5804375.86
Suma		76286770.05	129497010.54
Percentiles	25	2228576.6000	3018106.7400
	50	2567381.8900	4572714.9950
	75	2877565.1475	5433425.2525

Nota. Elaborado con el Software SPSS Versión 25

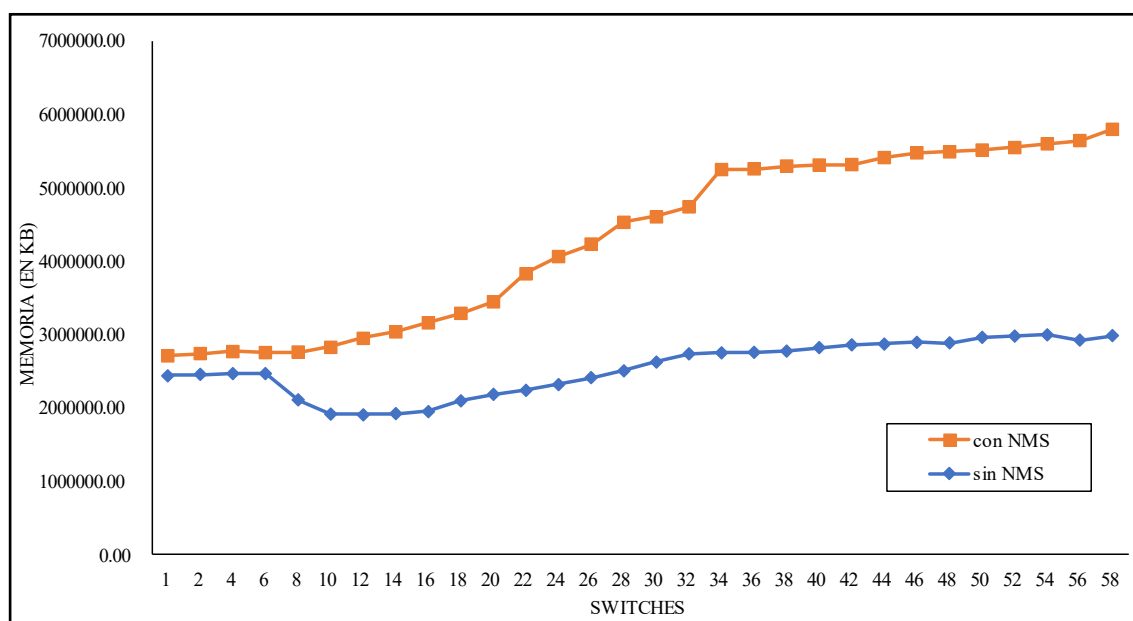
a. Existen múltiples modos. Se muestra el valor más pequeño.

Según los datos presentados en la Tabla 18, la cantidad de uso de memoria en la preprueba tuvo un promedio de 2542892 KB (equivalente a 2,5 GB), mientras que en la postprueba fue de 4316567 KB (equivalente a 4,3 GB). Estos resultados indican un aumento de alrededor del 70%, lo que sugiere que la cantidad de uso de memoria ha aumentado con el uso del NMS propuesto.

La Figura 67 presenta los datos tabulados de la cantidad de uso de memoria RAM y las diferentes configuraciones de switches. Se puede visualizar que la cantidad de memoria utilizada sin el NMS los datos no muestran variación con respecto al número de switches, en promedio, el uso de memoria empleado es de 2,4 Gigabytes. En cambio, la cantidad de uso de memoria con el NMS aumenta con respecto al número de switches y en todos los casos es mayor que la evaluación sin el NMS. Esta nueva cantidad de uso de memoria promedia en 3,5 Gigabytes desde 2,8 Gb con la configuración inicial hasta 5,2 con la configuración final.

Figura 67

Cantidad de uso de memoria RAM con diferentes configuraciones de switches



En la figura anterior se puede visualizar cambios significativos en el consumo generado con el NMS en comparación sin el NMS, por lo que se puede afirmar que el NMS afecta significativamente el rendimiento en cuanto a memoria RAM, aumentando la cantidad de uso de memoria.

Análisis descriptivo del indicador de cantidad de uso de CPU

Tabla 19

Frecuencia del indicador de cantidad de uso de CPU

		Cantidad de Uso de CPU Preprueba	Cantidad de Uso de CPU Postprueba
N	Válido	30	30
	Perdidos	0	0
Media		3.5920	6.3347
Error estándar de la media		.44570	.61351
Mediana		3.0050	5.7850
Moda		.67 ^a	1.65 ^a
Desv. Desviación		2.44119	3.36032
Varianza		5.959	11.292
Asimetría		.824	.466
Error estándar de asimetría		.427	.427
Curtosis		.107	-.690
Error estándar de curtosis		.833	.833
Rango		9.37	12.50
Mínimo		.67	1.65
Máximo		10.04	14.15
Suma		107.76	190.04
Percentiles	25	1.5950	3.4225
	50	3.0050	5.7850
	75	5.5800	9.0650

Nota. Elaborado con el Software SPSS Versión 25

a. Existen múltiples modos. Se muestra el valor más pequeño.

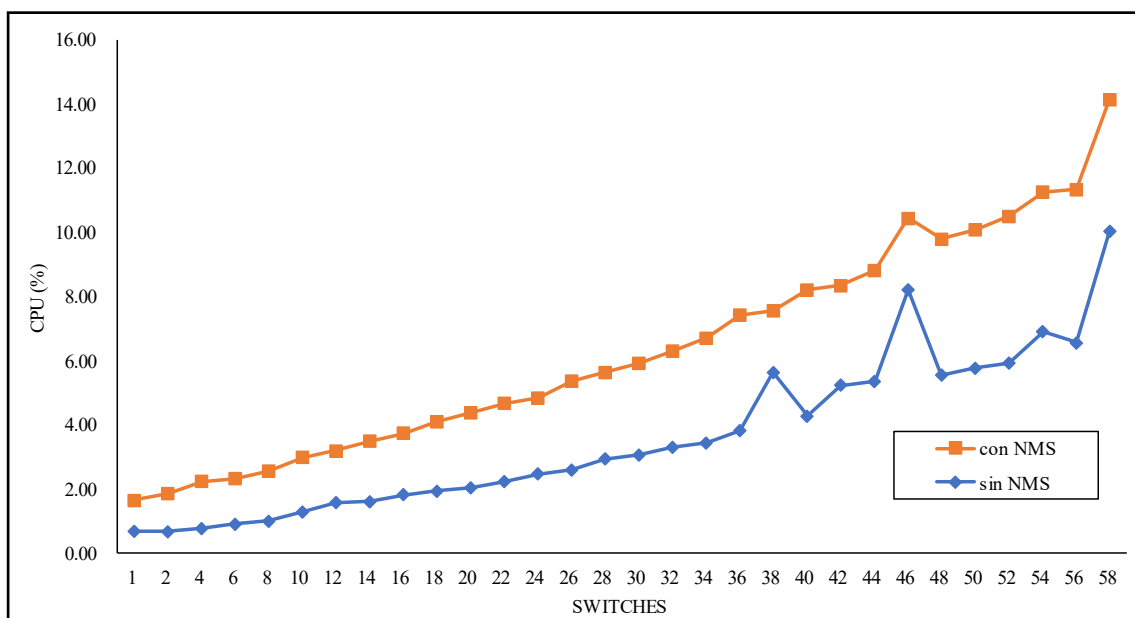
Los datos presentados en la Tabla 19 muestran que el indicador de cantidad de uso de CPU en la preprueba tuvo un promedio de 3,592000 %, mientras que en la postprueba fue de 6,334667 %. Estos resultados indican un aumento de aproximadamente 76,35%, lo que sugiere que la cantidad de uso de CPU ha aumentado con el uso del NMS propuesto.

En la Figura 68 muestra los datos tabulados del porcentaje de uso de CPU y las diferentes configuraciones de switches. Se puede visualizar que el porcentaje de

CPU utilizado sin el NMS, los datos muestran un aumento con respecto al número de switches, en promedio, el porcentaje de CPU es 4%. En la misma línea, el porcentaje con el NMS aumenta con respecto al número de switches y en todas las pruebas es mayor que la evaluación sin el NMS. El porcentaje de uso de CPU con el NMS muestra en promedio 6% desde 1,8% de la configuración inicial hasta 12% con la configuración final.

Figura 68

Porcentaje de uso de CPU con diferentes configuraciones de switches



En la Figura 68, se visualizan cambios significativos en el porcentaje de uso de CPU con el NMS en comparación sin el NMS, por lo que se puede afirmar que el NMS afecta significativamente el rendimiento en cuánto a uso de CPU, aumentando el porcentaje de uso de CPU.

Contrastación de hipótesis

Prueba de normalidad

Se realizó la prueba de normalidad usando método Shapiro-Wilk en todos los indicadores debido al tamaño reducido de las muestras (menor a 50). Se establecieron las hipótesis de normalidad y se aplicó una regla de decisión para determinar distribución normal o no normal de los datos.

Hipótesis estadísticas:

H_0 : La muestra cuenta con una distribución Normal

H_1 : La muestra cuenta con una distribución No Normal

Regla de decisión:

Nivel de confianza 95%

$p < 0.05$; si el valor de p es menor que 0.05, se rechaza la hipótesis nula, lo que revela que los datos no siguen una distribución normal.

$P \geq 0.05$; si el valor de p es mayor o igual a 0.05, la hipótesis nula es aceptada, entonces se revela que los datos siguen una distribución normal.

Indicador 1: Tiempo de Visualización de Topología

Para elegir la prueba de hipótesis apropiada, se verificó la distribución de los datos, en particular, si los datos de tiempo de visualización de la topología seguían una distribución normal.

Tabla 20

Prueba de normalidad del indicador de tiempo de visualización de topología

		Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
Tiempo	de	.103	30	.200*	.937	30	.077
Visualización	de						
Topología	Preprueba						

Tiempo	de	.093	30	.200*	.968	30	.474
Visualización	de						
Topología Postprueba							

Nota. Elaborado con el Software SPSS Versión 25

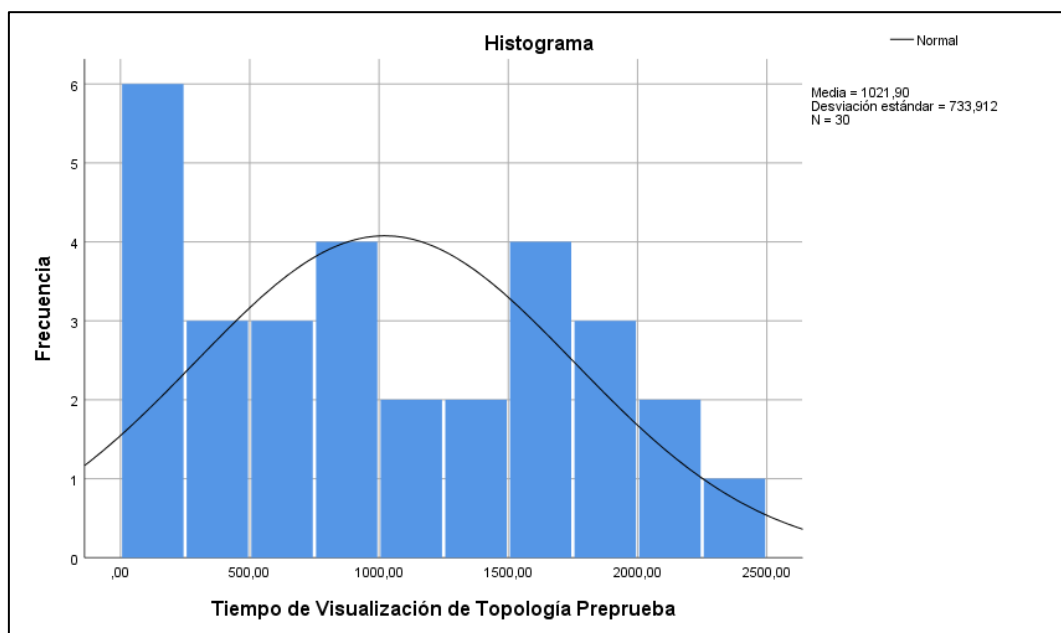
*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

Los resultados presentados en la Tabla 20 indican que el valor del sig. Para la preprueba es 0.077 y para la postprueba es 0.474. Ambos valores son mayores a 0.05, lo que nos permite concluir que los datos siguen una distribución **normal**. Esta observación se respalda visualmente en la Figura 69 y Figura 70.

Figura 69

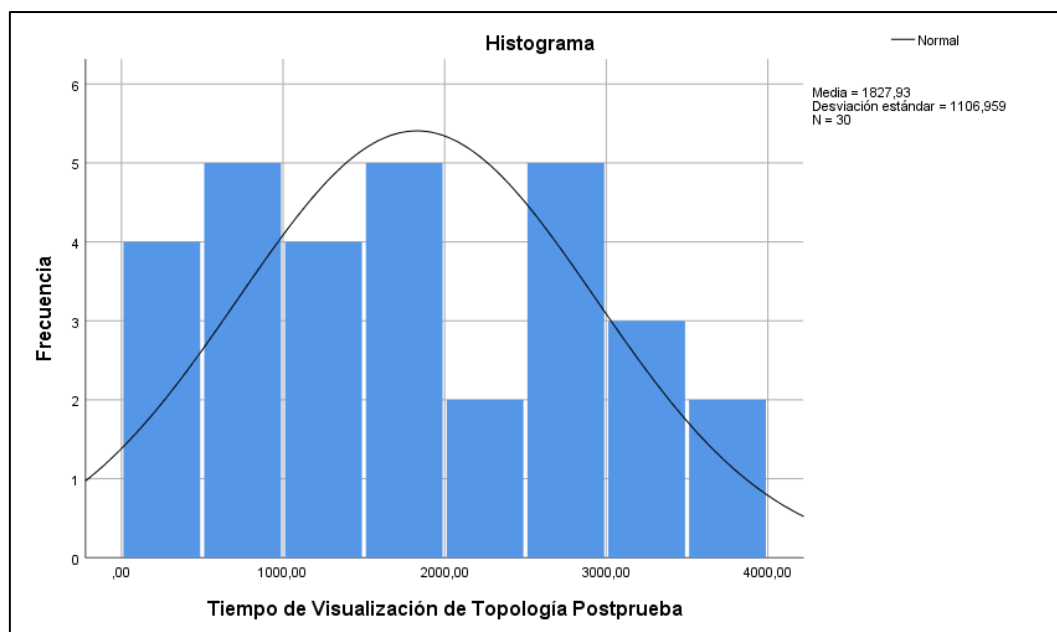
Prueba de normalidad del tiempo de visualización de topología en la preprueba



Nota. Elaborado con el Software SPSS Versión 25

Figura 70

Prueba de normalidad del tiempo de visualización de topología en la postprueba



Nota. Elaborado con el Software SPSS Versión 25

Indicador 2: Cantidad de uso de memoria

Se examinó si los datos de cantidad de uso de memoria presentaban una distribución normal para tomar una decisión sobre la prueba de hipótesis estadística apropiada a utilizar.

Tabla 21

Prueba de normalidad del indicador de cantidad de uso de memoria

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Cantidad de Uso de Memoria Preprueba	.171	30	.026	.906	30	.012
Cantidad de Uso de Memoria Postprueba	.223	30	.001	.848	30	.001

Nota. Elaborado con el Software SPSS Versión 25

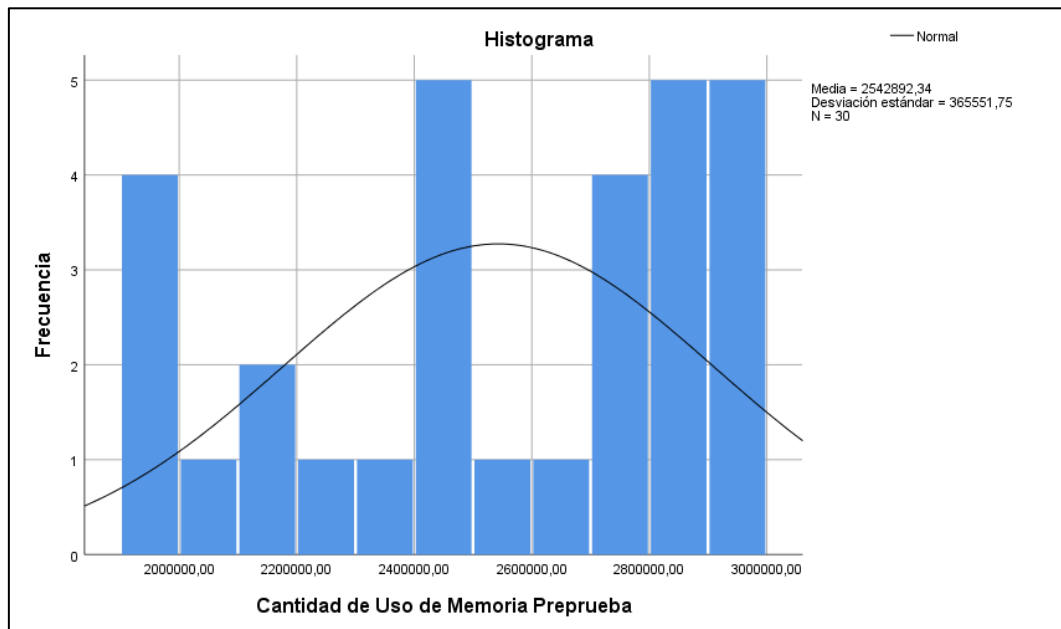
a. Corrección de significación de Lilliefors

En la Tabla 21, el valor p de la preprueba es 0.012 y el valor p de la postprueba es 0.001. Estos dos valores son inferiores a 0.05, lo que indica que los datos **no**

tienen una distribución normal. Esto también se puede apreciar en la Figura 71 y Figura 72.

Figura 71

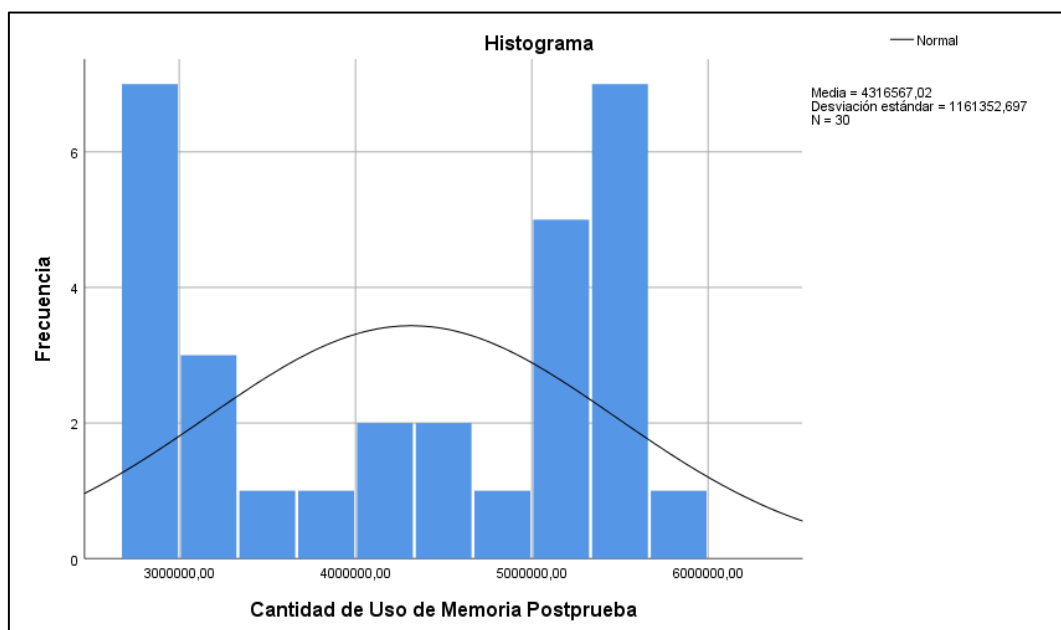
Prueba de normalidad de la cantidad de uso de memoria en la preprueba



Nota. Elaborado con el Software SPSS Versión 25

Figura 72

Prueba de normalidad de la cantidad de uso de memoria en la postprueba



Nota. Elaborado con el Software SPSS Versión 25

Indicador 3: Cantidad de uso de CPU

Para seleccionar la prueba de hipótesis, se realizó un análisis para determinar la distribución de los datos de la cantidad de uso de CPU.

Tabla 22

Prueba de normalidad del indicador de cantidad de uso de CPU

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Cantidad de Uso de CPU Preprueba	.126	30	.200*	.921	30	.029
Cantidad de Uso de CPU Postprueba	.105	30	.200*	.950	30	.173

Nota. Elaborado con el Software SPSS Versión 25

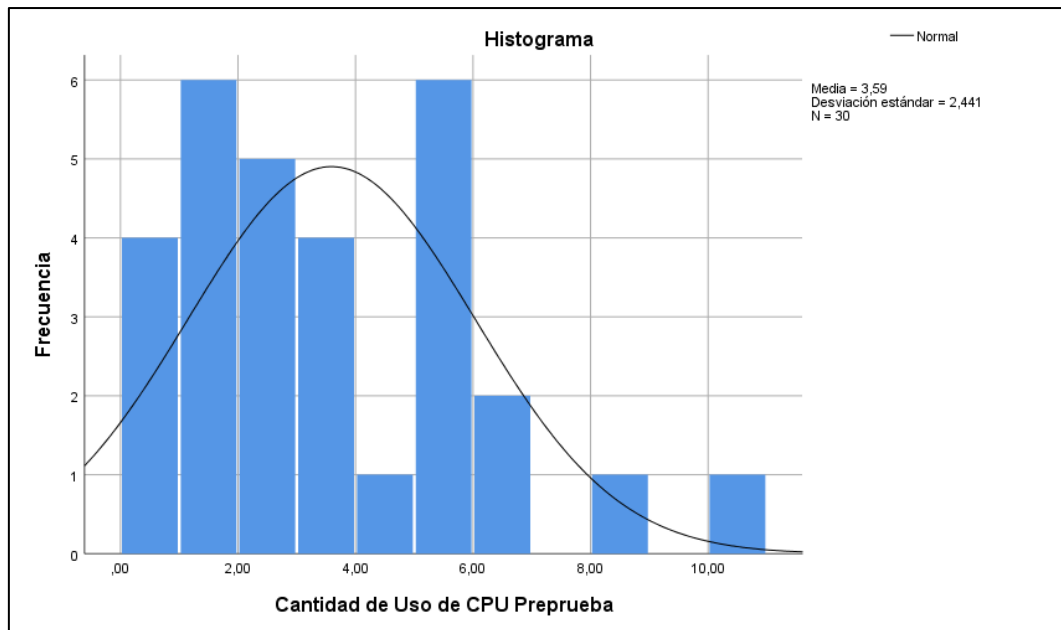
*. Esto es un límite inferior de la significación verdadera.

a. Corrección de significación de Lilliefors

La Tabla 22 contiene los resultados del análisis, mostrando que el valor de significancia (sig) para la preprueba es 0.029 y para la postprueba es 0.173. Dado que al menos uno de los valores es menor a 0.05, se concluye que los datos **no siguen una distribución normal**. Esta conclusión también es respaldada por las Figuras 73 y 74 que muestran dicha distribución.

Figura 73

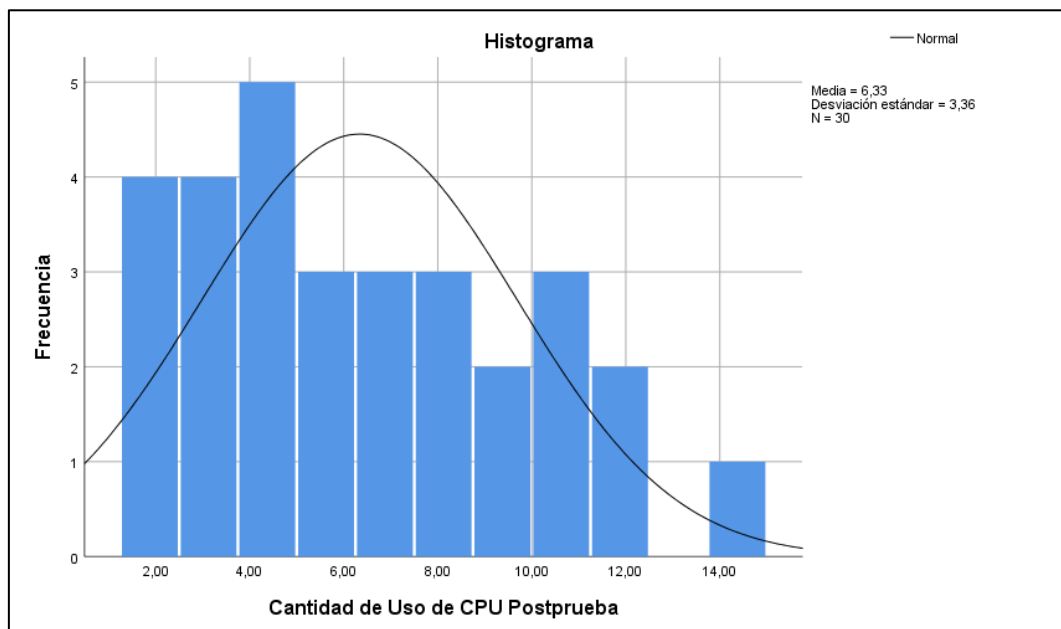
Prueba de normalidad de cantidad de uso de CPU en la preprueba



Nota. Elaborado con el Software SPSS Versión 25

Figura 74

Prueba de normalidad de cantidad de uso CPU en la postprueba



Nota. Elaborado con el Software SPSS Versión 25

Prueba de hipótesis

Formulación de hipótesis específica 1

H_0 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace no influye en el tiempo de visualización de la topología.

H_1 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología.

Tabla 23

Resumen del modelo del indicador 1: Tiempo de Visualización de Topología

		Tiempo de Visualización de Topología Preprueba – Tiempo de Visualización de Topología Postprueba
Diferencias emparejadas	Media	-806.03
	Desv. Desviación	392.645
	Desv. Error promedio	71.6868
	95% de intervalo de confianza de la diferencia	Inferior -952.65
		Superior -659.41
t		-11.244
gl		29
Sig. (bilateral)		0

Nota. Elaborado con el Software SPSS versión 25

La Tabla 23 detalla los resultados de la prueba paramétrica T-Student para muestras relacionadas. El valor de significancia (sig.) obtenido para el indicador de tiempo de visualización de la topología es 0.00, siendo menor que 0.05, utilizado como límite para aceptar la hipótesis de investigación.

Así, al tener un p-valor inferior a 0.05, se puede inferir que se respalda la hipótesis alternativa (H_1) y se descarta la hipótesis nula (H_0). Esto indica que un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología.

Formulación de hipótesis específica 2

H_0 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace no influye en la cantidad de uso de memoria.

H_1 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.

Tabla 24

Resumen del modelo del indicador 2: Cantidad de Uso de Memoria

Estadísticos de prueba ^a	
Cantidad de Uso de Memoria Postprueba – Cantidad de Uso de Memoria Preprueba	
Z	-4.782 ^b
Sig. Asintótica(bilateral)	.000

Nota. Elaborado con el Software SPSS versión 25

a. Prueba de rangos con signo de Wilcoxon

b. Se basa en rangos negativos.

La Tabla 24 muestra los resultados de la prueba no paramétrica de Wilcoxon. El valor de significancia (sig.) obtenido es 0.00, siendo menor que 0.05, utilizado como límite para aceptar la hipótesis de investigación.

Así, al tener un p-valor inferior a 0.05, se puede inferir que se respalda la hipótesis alternativa (H_1) y se descarta la hipótesis nula (H_0). Esto indica que un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.

Formulación de hipótesis específica 3

H_0 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace no influye en la cantidad de uso de CPU

H_1 : Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de CPU.

Tabla 25

Resumen del modelo del indicador 1: Cantidad de Uso de CPU

Estadísticos de prueba ^a	
Cantidad de Uso de CPU Postprueba – Cantidad de Uso de CPU Preprueba	
Z	-4.782 ^b
Sig. Asintótica(bilateral)	.000

Nota. Elaborado con el Software SPSS versión 25

a. Prueba de rangos con signo de Wilcoxon

b. Se basa en rangos negativos.

La Tabla 25 muestra los resultados de la prueba no paramétrica de Wilcoxon. El valor de significancia (sig.) obtenido es 0.00, siendo menor que 0.05, utilizado como límite para aceptar la hipótesis de investigación.

Así, al tener un p-valor inferior a 0.05, se puede inferir que se respalda la hipótesis alternativa (H_1) y se descarta la hipótesis nula (H_0). Esto indica que un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de CPU.

CAPÍTULO IV

DISCUSIÓN

Basándonos en los resultados derivados de este estudio, se pueden apreciar en la Tabla 17 y Figura 66 los valores correspondientes al tiempo de visualización de topología. En la preprueba, el tiempo medio fue de 1021,8993 ms, mientras que en la postprueba fue de 1827,9293 ms. Estos resultados indican un aumento aproximado del 79%. Además, los resultados de la prueba de hipótesis mostrados en la Tabla 23, mediante la prueba paramétrica T-Student para muestras relacionadas, revelan un valor de significancia (sig.) igual a 0,000, que es menor que el nivel de significancia de 0,05. Esto demuestra que el Sistema de Monitoreo de Red-NMS de detección de fallos de enlace tiene una influencia significativa en el tiempo de visualización de la topología. Estos hallazgos coinciden con estudios previos realizados por Usman et al. (2019) y Montoya-Munoz et al. (2021), que también sugieren tiempos no mayores a 10000 ms. Los tiempos obtenidos por el NMS propuesto son inferiores a los 10000 ms para un número de switches menor a 58. Por lo tanto, los tiempos logrados por el NMS se consideran aceptables, lo que permite la interacción visual y el desarrollo tecnológico del NMS para investigaciones futuras.

Al mismo tiempo, se observa en la Tabla 18 y Figura 67 que el indicador de cantidad de uso de memoria tuvo un valor promedio de 2542892 KB (2,5 GB) en la preprueba y de 4316567 KB (4,3 GB) en la postprueba. Estos resultados indican un aumento de aproximadamente 70%, lo que significa que el uso de memoria ha aumentado con la implementación del NMS propuesto. Además, en la Tabla 24, mediante la prueba no paramétrica de Wilcoxon, se obtiene un valor de significancia (sig.) igual a 0,00 para el indicador de cantidad de uso de memoria, que es menor que el nivel de significancia de 0,05. Esto demuestra que el Sistema de Monitoreo de Red-NMS de detección de fallos de enlace tiene una influencia significativa en la cantidad de uso de memoria. Estos hallazgos son consistentes con el estudio realizado por

Lange et al. (2018), que también señaló un incremento en el uso de memoria. Sin embargo, este estudio solo se aplica para configuraciones de switches menores a 8. En el caso de configuraciones mayores, solo se cumplen las pruebas sin NMS, mientras que con el NMS propuesto se observa un notorio incremento. Este aumento se justifica debido a que la incorporación de un NMS carga los recursos del sistema computacional. No obstante, el NMS propuesto ofrece características de monitoreo valiosas, como la detección de fallos de enlace.

Como observado en la Tabla 19 y Figura 68 que el indicador de cantidad de uso de CPU tuvo un valor promedio de 3,592000% en la preprueba y de 6,334667% en la postprueba. Estos resultados indican un aumento de aproximadamente 76,35% en el uso de CPU. Asimismo, al analizar los datos presentados en la Tabla 25 mediante la prueba no paramétrica de Wilcoxon, se obtiene un valor de significancia (sig.) igual a 0,00, el cual es menor que el nivel de sig. de 0,05. Esto sugiere que el Sistema de Monitoreo de Red-NMS de detección de fallos de enlace tiene una influencia significativa en la cantidad de uso de CPU. Estos resultados son consistentes con el estudio realizado por Lange et al. (2018), que también señaló un aumento en el uso de CPU. Este incremento se debe a que la incorporación de un NMS recarga los recursos del sistema computacional. Sin embargo, es importante destacar que el NMS propuesto también ofrece características valiosas de monitoreo, como la detección de fallos de enlace.

CAPÍTULO V
CONCLUSIONES

En la conclusión del estudio, se determinó que el Sistema de Monitoreo de Red (NMS) para la detección de fallos de enlace influye significativamente en el proceso de monitoreo de Redes Definidas por Software (SDN). El NMS se presenta como una plataforma abierta y visualmente interactiva que fomenta la innovación en las redes en laboratorios SDN para la formación de futuros especialistas en telecomunicaciones. Los resultados obtenidos en este trabajo de investigación llevan a las siguientes conclusiones:

Para cumplir con el objetivo 1 se presentan que el tiempo de visualización de la topología con el NMS es significativamente mayor al monitoreo sin NMS, solo con el OpenDayLight GUI.

Cumpliendo con el objetivo 2 se muestra que el NMS influye significativamente en el aumento de uso memoria RAM con respecto a sin el NMS.

Para completar el objetivo 3, se presenta que el NMS influye significativamente en el aumento del porcentaje de uso CPU con respecto a un entorno sin el NMS.

CAPÍTULO VI
RECOMENDACIONES

La principal limitación fue la disponibilidad limitada de recursos humanos especializados en redes definidas por software para mejorar el software y reducir los consumos. Se recomienda contar con personal experto en esta área para optimizar el desarrollo del NMS y mejorar su eficiencia en futuras implementaciones y evaluaciones en entornos reales de laboratorio SDN.

Se recomienda a los próximos investigadores utilizar herramientas estandarizadas y controladores actuales como OpenDayLight y ONOS. Estas herramientas permitirán a futuros investigadores implementar las últimas tecnologías y avanzar en soluciones reales en las Redes Definidas por Software. Al aprovechar estas plataformas, se facilitará el desarrollo y evaluación del NMS proceso de monitoreo, mejorando así la eficiencia y eficacia en entornos reales de laboratorio SDN.

Se recomienda que el NMS propuesto también se utilice para posteriores investigaciones en ciberseguridad y control de tráfico en la red debido a su modularidad y fácil adopción en entornos de desarrollo.

REFERENCIAS

- Aboubakar, M., Kellil, M., & Roux, P. (2022). A review of IoT network management: Current status and perspectives. *Journal of King Saud University - Computer and Information Sciences*, 34(7), 4163–4176. <https://doi.org/10.1016/j.jksuci.2021.03.006>
- AlZoman, R., & Alenazi, M. J. F. (2020). Exploiting SDN to Improve QoS of Smart City Networks Against Link Failures. *2020 Seventh International Conference on Software Defined Systems (SDS)*, 100–106. <https://doi.org/10.1109/SDS49854.2020.9143878>
- Bafoutsou, G., Dekker, M., & European Union Agency for Cybersecurity. (2020). *Telecom security during a pandemic: Telecom security good practices and lessons learned from the COVID-19 outbreak*. https://op.europa.eu/publication/manifestation_identifier/PUB_TP0220882EN
N
- Binsahaq, A., Sheltami, T. R., & Salah, K. (2019). A Survey on Autonomic Provisioning and Management of QoS in SDN Networks. *IEEE Access*, 7, 73384–73435. <https://doi.org/10.1109/ACCESS.2019.2919957>
- Bonfim, M. S., Dias, K. L., & Fernandes, S. F. L. (2019). Integrated NFV/SDN Architectures: A Systematic Literature Review. *ACM Computing Surveys*, 51(6), 1–39. <https://doi.org/10.1145/3172866>
- Brooks, C. J., Grow, C., Craig, P., & Short, D. (2018). *Cybersecurity essentials*. John Wiley & Sons Inc.
- Chen, W., Haque, A., & Sedig, K. (2021). Design of Interactive Visualizations for Next-Generation Ultra-Large Communication Networks. *IEEE Access*, 9, 26968–26982. <https://doi.org/10.1109/ACCESS.2021.3057803>

- Cisco. (2020). *2020 Global Networking Trends Report*.
https://www.cisco.com/c/dam/m/en_us/solutions/enterprise-networks/networking-report/files/GLBL-ENG_NB-06_0_NA_RPT_PDF_MOFU-no-NetworkingTrendsReport-NB_rpten018612_5.pdf
- Clemm, A., Zhani, M. F., & Boutaba, R. (2020). Network Management 2030: Operations and Control of Network 2030 Services. *Journal of Network and Systems Management*, 28(4), 721–750. <https://doi.org/10.1007/s10922-020-09517-0>
- Espinel Villalobos, R. I., Ardila Triana, E., Zarate Ceballos, H., & Ortiz Triviño, J. E. (2021). Design and Implementation of Network Monitoring System for Campus Infrastructure Using Software Agents. *Ingeniería e Investigación*, 42(1), e87564. <https://doi.org/10.15446/ing.investig.v42n1.87564>
- Fitzek, F. H. P., Granelli, F., & Seeling, P. (Eds.). (2020). *Computing in communication networks: From theory to practice*. Academic Press, an imprint of Elsevier.
- GNS3. (s/f). *Getting Started with GNS3 | GNS3 Documentation*. Recuperado el 22 de febrero de 2021, de <https://docs.gns3.com/docs/>
- Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., Vasilakos, A. V., & Marsono, M. N. (2021). A comprehensive survey of load balancing techniques in software-defined network. *Journal of Network and Computer Applications*, 174, 102856. <https://doi.org/10.1016/j.jnca.2020.102856>
- Hernández-Sampieri, D. R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill Interamericana de España S.L.

- Hong, C.-H., & Varghese, B. (2019). Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. *ACM Computing Surveys*, 52(5), 97:1-97:37. <https://doi.org/10.1145/3326066>
- Hurwitz, J. S., & Kirsch, D. (2020). *Cloud Computing For Dummies* (2a ed.). Wiley.
- Ilyés, E. (2019). *Create your own agile methodology for your research and development team*. 823–829. <https://doi.org/10.15439/2019F209>
- Inter-American Development Bank. (2020). *Convivir con el coronavirus: ¿Cómo dar continuidad a la educación? | Publications*. <https://publications.iadb.org/publications/spanish/document/Convivir-con-el-coronavirus-Como-dar-continuidad-a-la-educacion.pdf>
- Jain, V., Yatri, V., Kanchan, & Kapoor, C. (2019). Software defined networking: State-of-the-art. *Journal of High Speed Networks*, 25(1), 1–40. <https://doi.org/10.3233/JHS-190601>
- Jiménez, M. B., Fernández, D., Rivadeneira, J. E., Bellido, L., & Cárdenas, A. (2021). A Survey of the Main Security Issues and Solutions for the SDN Architecture. *IEEE Access*, 9, 122016–122038. <https://doi.org/10.1109/ACCESS.2021.3109564>
- KathrynEE. (2021, noviembre 9). *Understand Scrum process work items types & workflow—Azure Boards*. <https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/scrum-process-workflow>
- Kavitha, G., Kavitha, R., & A.V, A. G. (2019). Network Monitoring on Cloud Environment using SDN. *International Journal of Engineering and Advanced Technology*, 8(6S2), 171–173. <https://doi.org/10.35940/ijeat.F1044.0886S219>
- Keshari, S. K., Kansal, V., & Kumar, S. (2021). A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN). *Wireless Personal*

Communications, 116(3), 2593–2614. <https://doi.org/10.1007/s11277-020-07812-2>

Khan, N., Salleh, R. bin, Koubaa, A., Khan, Z., Khan, M. K., & Ali, I. (2023). Data plane failure and its recovery techniques in SDN: A systematic literature review. *Journal of King Saud University - Computer and Information Sciences*, 35(3), 176–201. <https://doi.org/10.1016/j.jksuci.2023.02.001>

Lange, S., Reinhart, L., Zinner, T., Hock, D., Gray, N., & Tran-Gia, P. (2018). Integrating network management information into the SDN control plane. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 1–9. <https://doi.org/10.1109/NOMS.2018.8406228>

Layton, M. C. (2022). *Scrum for dummies* (3rd ed.). John Wiley & Sons Inc.

Lin, Y.-H., Yang, C.-W., Chuang, T.-C., Liu, M., & Chang, M.-C. (2019). An Integrated Network Monitoring System for SDN VPN. *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 1–4. <https://doi.org/10.23919/APNOMS.2019.8892841>

MINEDU. (2020). *Resolución Vice-Ministerial 095-2020-MINEDU*. https://cdn.www.gob.pe/uploads/document/file/671513/RESOLUCION_VICE_MINISTERIAL-00095-2020-MINEDU.pdf

MINEDU. (2021). *Decreto Supremo-N° 002-2021-MINEDU*. <http://busquedas.elperuano.pe/normaslegales/decreto-supremo-que-establece-los-criterios-para-la-determin-decreto-supremo-n-002-2021-minedu-1921545-5/>

Mininet. (s/f). *Introduction to Mininet · mininet/mininet Wiki*. Recuperado el 22 de febrero de 2021, de <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>

Mininet. (2018). *Cbench*. GitHub. <https://github.com/mininet/oflops>

Mohammed, A. K., El Zoghby, H. M., & Elmesalawy, M. M. (2020). Remote Controlled Laboratory Experiments for Engineering Education in the Post-COVID-19 Era: Concept and Example. *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 629–634. <https://doi.org/10.1109/NILES50944.2020.9257888>

Montoya-Munoz, A. I., Casas-Velasco, D., Estrada-Solano, F., Caicedo Rendon, O. M., & Saldanha Da Fonseca, N. L. (2021). An approach based on Yet Another Next Generation for software-defined networking management. *International Journal of Communication Systems*, 34(11). <https://doi.org/10.1002/dac.4855>

MTC. (2021). *MTC mejorará los servicios de telecomunicaciones*. <https://elperuano.pe/noticia/119879-mtc-mejorara-los-servicios-de-telecomunicaciones>

Mukhopadhyay, S., Booth, A. L., Calkins, S. M., Doxtader, E. E., Fine, S. W., Gardner, J. M., Gonzalez, R. S., Mirza, K. M., & Jiang, X. (Sara). (2020). Leveraging Technology for Remote Learning in the Era of COVID-19 and Social Distancing. *Archives of Pathology & Laboratory Medicine*, 144(9), 1027–1036. <https://doi.org/10.5858/arpa.2020-0201-ED>

Ndiaye, M., Abu-Mahfouz, A. M., & Hancke, G. P. (2020). SDNMM—A Generic SDN-Based Modular Management System for Wireless Sensor Networks. *IEEE Systems Journal*, 14(2), 2347–2357. <https://doi.org/10.1109/JSYST.2019.2927946>

NSTAC. (2020, agosto). *NSTAC REPORT TO THE PRESIDENT on Software-Defined Networking*.

<https://www.cisa.gov/sites/default/files/publications/NSTAC%20SDN%20Report%20%288-12-20%29.pdf>

ONOS. (2020). *ONOS*. <https://wiki.onosproject.org/display/ONOS/ONOS>

OpenDayLight. (2021). *OpenDaylight—Overview*. <https://wiki.opendaylight.org/view/>

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

Panek, C. (2019). *Networking fundamentals*. John Wiley & Sons, Inc.

PCM. (2020). *DECRETO SUPREMO-N° 117-2020-PCM*. El Peruano. <http://busquedas.elperuano.pe/normaslegales/decreto-supremo-que-aprueba-la-fase-3-de-la-reanudacion-de-a-decreto-supremo-n-117-2020-pcm-1869317-1/>

Robles-Gómez, A., Tobarra, L., Pastor-Vargas, R., Hernández, R., & Cano, J. (2020). *Emulating and Evaluating Virtual Remote Laboratories for Cybersecurity*. *Sensors*, 20(11), 3011. <https://doi.org/10.3390/s20113011>

Ryu SDN Framework. (s/f). Recuperado el 21 de febrero de 2021, de <https://ryu-sdn.org/>

SeoulTech. (2020). *SeoulTech and Peru Universidad Nacional de Ingeniería (UNI) Create a Cyber Security Major and Support Local Community Education Programs*.

<https://en.seoultech.ac.kr/news/news/?do=commonview&searchtext=&searchtype=-1&nowpage=1&bnum=57087&bidx=490069&cate=23>

Srivastava, A. K., Venkataramanan, V., & Hauser, C. (2023). *Cyber Infrastructure for the Smart Electric Grid*.

- Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2018). Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks. *2018 4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018*, 462–469. <https://doi.org/10.1109/NETSOFT.2018.8460090>
- The Institution of Engineering and Technology. (2018). Big Data and Software Defined Networks. En J. Taheri (Ed.), *Big Data and Software Defined Networks*. Institution of Engineering and Technology. <https://doi.org/10.1049/pbpc015e>
- Tran, H. M., Le, S. T., Nguyen, S. V., & Le, H.-D. (2019). A Web-Based Management System for Software Defined Network Controllers. *2019 International Conference on Advanced Computing and Applications (ACOMP)*, 1–6. <https://doi.org/10.1109/ACOMP.2019.00008>
- Tsai, P.-W., Tsai, C.-W., Hsu, C.-W., & Yang, C.-S. (2018). Network Monitoring in Software-Defined Networking: A Review. *IEEE Systems Journal*, 12(4), 3958–3969. <https://doi.org/10.1109/JSYST.2018.2798060>
- UNESCO. (2020, junio 25). *La campaña “La Nueva Normalidad” de la UNESCO*. UNESCO. <https://es.unesco.org/campaign/nextnormal>
- Usman, M., Risdianto, A. C., Han, J., & Kim, J. (2019). Interactive Visualization of SDN-Enabled Multisite Cloud Playgrounds Leveraging SmartX MultiView Visibility Framework. *The Computer Journal*, 62(6), 838–854. <https://doi.org/10.1093/comjnl/bxy103>
- Vela, A. P., Gifre, Li., De Dios, O. G., Ruiz, M., & Velasco, L. (2018). CASTOR: A Monitoring and Data Analytics Framework to Help Operators Understand what is Going on in their Networks. *2018 European Conference on Optical Communication (ECOC)*, 1–3. <https://doi.org/10.1109/ECOC.2018.8535500>

- Villota, W., Gironza, M., Ordoñez, A., & Caicedo Rendon, O. M. (2018). On the Feasibility of Using Hierarchical Task Networks and Network Functions Virtualization for Managing Software-Defined Networks. *IEEE Access*, 6, 38026–38040. <https://doi.org/10.1109/ACCESS.2018.2852649>
- Vollbrecht, P. J., Porter-Stransky, K. A., & Lackey-Cornelison, W. L. (2020). Lessons learned while creating an effective emergency remote learning environment for students during the COVID-19 pandemic. *Advances in Physiology Education*, 44(4), 722–725. <https://doi.org/10.1152/advan.00140.2020>
- Wang, L., Sun, M., & Tang, S. (2019). SCSCDaylight: Network Monitoring Tools for Software-Defined Networks Based on Opendaylight. *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, 320–323. <https://doi.org/10.1109/ICICAS48597.2019.00075>
- WORLD BANK. (2020a). *How countries are using edtech (including online learning, radio, television, texting) to support access to remote learning during the COVID-19 pandemic* [Text/HTML]. World Bank. <https://www.worldbank.org/en/topic/edutech/brief/how-countries-are-using-edtech-to-support-remote-learning-during-the-covid-19-pandemic>
- WORLD BANK. (2020b). *Remote Learning response to COVID-19 Knowledge Pack*. <https://pubdocs.worldbank.org/en/925611587160522864/KnowledgePack-COVID19-RemoteLearning-LowResource-EdTech.pdf>
- Yang, C.-T., Chen, S.-T., Liu, J.-C., Yang, Y.-Y., Mitra, K., & Ranjan, R. (2019). Implementation of a real-time network traffic monitoring service with network functions virtualization. *Future Generation Computer Systems*, 93, 687–701. <https://doi.org/10.1016/j.future.2018.08.050>

- Yu, Y., Li, X., Leng, X., Song, L., Bu, K., Chen, Y., Yang, J., Zhang, L., Cheng, K., & Xiao, X. (2019). Fault Management in Software-Defined Networking: A Survey. *IEEE Communications Surveys & Tutorials*, 21(1), 349–392. <https://doi.org/10.1109/COMST.2018.2868922>
- Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., & Sun, Y. (2019). A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning. *IEEE Access*, 7, 95397–95417. <https://doi.org/10.1109/ACCESS.2019.2928564>

ANEXOS

Anexo 1: Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES INDICADORES	E	METODOLOGÍA
PROBLEMA GENERAL ¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de Redes Definidas por Software?	OBJETIVO GENERAL Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de Redes Definidas por Software.	HIPÓTESIS GENERAL Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el proceso de monitoreo de redes definidas por software.	Variable independiente (X) •Sistema de Monitoreo de Red de detección de fallos de enlace Variable dependiente (Y) •Proceso de monitoreo de redes definidas por software Indicadores <input type="checkbox"/> Tiempo de visualización de la topología. <input type="checkbox"/> Cantidad de Uso de Memoria. <input type="checkbox"/> Cantidad de Uso de CPU.		Tipo de Investigación Esta investigación es del tipo: <i>Aplicada</i> Nivel de Investigación Esta investigación es de nivel: <i>Explicativa</i> Diseño de Investigación Esta investigación tiene un diseño: Experimental - <i>Pre experimental</i>
PROBLEMA ESPECÍFICO ¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología?	OBJETIVO ESPECÍFICO Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología.	HIPÓTESIS ESPECÍFICO Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en el tiempo de visualización de la topología.			
¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria?	Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.	Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.			
¿En qué medida el uso de un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria?	Determinar en qué medida un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.	Un Sistema de Monitoreo de Red-NMS de detección de fallos de enlace influye en la cantidad de uso de memoria.			

de fallos de enlace influye de fallos de enlace influye en la cantidad de uso de
en la cantidad de uso de la cantidad de uso de CPU.
CPU? CPU.

Anexo 2: Matriz de operacionalización de variables

VARIABLE	DEFINICION CONCEPTUAL DE VARIABLE	TIPO DE INDICADOR	DIMENSIONES	DEFINICION CONCEPTUAL DE DIMENSION	INDICADORES	ESCALA DE LOS INDICADORES	TÉCNICA	INSTRUMENTO	UNIDAD DE MEDIDA
Proceso de monitoreo de redes definidas por software	Según Tsai et al. (2018) enfatiza la importancia de mejorar la eficiencia del monitoreo de redes al reducir la sobrecarga mediante la medición de estadísticas y el estado de la red. Esta tarea es crucial para lograr un monitoreo más efectivo y eficiente. Además, la importancia de la	Cuantitativo	Presentación	La visualización o presentación de información es la representación visual e interactiva de datos que soportadas computacionalmente permiten la generación de información y conocimiento para apoyar las tareas de gestión. En la red, la visualización de la topología es un parte fundamental en el análisis de los administradores.	Tiempo de visualización de la topología.	Intervalo	Observación	Ficha de observación	Milisegundos (ms)

presentación en el proceso de monitoreo de red recae en la detección de anomalías de red en tiempo real. Es por ello que el tiempo de presentación de los datos de red deben ser tomados en cuenta en el desarrollo de las investigaciones.

Dicha visualización requiere de un tiempo de respuesta en los dispositivos para ser mostrados (Chen et al., 2021).

Eficiencia	La eficiencia de la gestión y monitoreo de las redes definidas por software (SDN) puede verse afectada por diferentes parámetros de red. Para evaluar y medir dicha eficiencia, se utilizan métricas, como el uso de CPU y memoria, permitiendo analizar el impacto computacional generado por el procesamiento de información adicional.	Cantidad de Uso de Memoria.	Intervalo	Observación	Ficha de observación	KiloBytes (KB)
		Cantidad de Uso de CPU.	Intervalo	Observación	Ficha de observación	Porcentaje (%)

(Lange et al.,
2018).

Anexo 3: Validación de Expertos

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO A TRAVES DE JUICIO DE EXPERTO

Título de la Investigación	SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE
Autor (es)	Andres Junior Aparcana Tasayco

N°	DIMENSIONES / Ítems		Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
			Si	No	Si	No	Si	No	
1	DIMENSIÓN 1: PRESENTACIÓN		X		X		X		-
	Tiempo de visualización de la topología.								
2	DIMENSIÓN 2: EFICIENCIA		X		X		X		-
	Uso de memoria.								
3	Uso de CPU.		X		X		X		-

Observaciones (precisar si hay suficiencia): Ninguna

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador. **Mg.** /Dr.: Daniel Díaz Ataucuri DNI: 07139361

Especialidad del validador: Ingeniero Electrónico

¹Pertinencia: El ítem corresponde al concepto teórico formulado.

²Relevancia: El ítem es apropiado para representar al componente o dimensión específica del constructo

³Claridad: Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

27 de Julio del 2023


Firma del Experto Informante.

(a) Mg. Daniel Díaz Ataucuri

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO A TRAVES DE JUICIO DE EXPERTO

Título de la Investigación	SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE
Autor (es)	Andres Junior Aparcana Tasayco

N°	DIMENSIONES / Ítems		Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
			Si	No	Si	No	Si	No	
	DIMENSIÓN 1:								
1	PRESENTACIÓN	Tiempo de visualización de la topología.	X		X		X		
	DIMENSIÓN 2:		Si	No	Si	No	Si	No	
2	EFICIENCIA	Uso de memoria.	X		X		X		
3		Uso de CPU.	X		X		X		

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: **Aplicable** [X] **Aplicable después de corregir** [] **No aplicable** []

Apellidos y nombres del juez validador: Mg. Dr.: Javier Arturo Gamboa Cruzado **DNI:** 17906323

Especialidad del validador: Ingeniería de Sistemas

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

20 de Julio del 2023



Firma del Experto Informante

(b) Dr. Javier Gamboa Cruzado

CERTIFICADO DE VALIDEZ DE CONTENIDO DEL INSTRUMENTO A TRAVES DE JUICIO DE EXPERTO

Título de la Investigación	SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE
Autor (es)	Andres Junior Aparcana Tasayco

N°	DIMENSIONES / Ítems	Pertinencia ¹		Relevancia ²		Claridad ³		Sugerencias
		Si	No	Si	No	Si	No	
	DIMENSIÓN 1:							
1	PRESENTACIÓN Tiempo de visualización de la topología.	X		X		X		
	DIMENSIÓN 2:							
2	EFICIENCIA Uso de memoria.	X		X		X		
3	Uso de CPU.	X		X		X		

Observaciones (precisar si hay suficiencia): _____

Opinión de aplicabilidad: Aplicable [X] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador: Mg. Dr.: CELIS HENRY OCHOA JAYO DNI: 41647498

Especialidad del validador: INGENIERIA DE SISTEMAS

Link del CTI VITAE: https://ctivitae.concytec.gob.pe/appDirectorioCTI/VerDatosInvestigador.do?id_investigador=194296

¹**Pertinencia:** El ítem corresponde al concepto teórico formulado.

²**Relevancia:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

1 de Agosto del 2023



CELIS HENRY OCHOA JAYO
INGENIERO DE SISTEMAS
Reg. CIP N° 136649

Firma del Experto Informante

(c) Mg. Henry Celis Ochoa

Anexo 4: Matriz de Resultados

Procesos de Control	Switches	Enlaces	Ancho de Banda		Uso de Memoria		Uso de CPU		Tiempo de Visualización de Topología	
			Preprueba	Posprueba	Preprueba	Posprueba	Preprueba	Posprueba	Preprueba	Posprueba
1	1	1	6310.79	6463.19	2440772.61	2710866.90	0.69	1.65	0.05	54.34
2	2	3	13698.04	13325.64	2459054.49	2742786.53	0.67	1.86	17.54	157.13
3	4	7	21742.69	21815.51	2468490.54	2771211.38	0.77	2.25	49.97	254.16
4	6	11	24797.84	25791.46	2470269.70	2756938.00	0.91	2.33	97.21	412.44
5	8	15	28214.28	31702.48	2110239.24	2763549.72	1.00	2.57	170.50	542.44
6	10	19	33918.07	35604.22	1916097.42	2829717.62	1.29	3.00	222.95	745.45
7	12	23	37751.07	37683.08	1908315.43	2951566.86	1.58	3.19	286.43	810.11
8	14	27	38302.40	39041.18	1924640.61	3040286.70	1.60	3.50	353.23	925.53
9	16	31	40900.44	42364.77	1953020.77	3161631.29	1.83	3.75	428.47	898.96
10	18	35	43670.30	44054.48	2099359.52	3291889.03	1.94	4.10	504.30	1157.10
11	20	39	46104.48	45894.45	2186843.09	3449966.30	2.05	4.38	575.36	1362.69
12	22	43	46794.72	47761.71	2242487.77	3833411.24	2.24	4.68	630.31	1315.94
13	24	47	40218.29	48558.77	2320028.10	4068138.33	2.48	4.84	779.73	1586.96
14	26	51	48132.39	49822.14	2414071.31	4238088.78	2.59	5.37	845.68	1493.75
15	28	55	51160.23	52234.03	2507256.58	4534359.95	2.95	5.64	920.93	1998.09
16	30	59	47118.85	52883.38	2627507.20	4611070.04	3.06	5.93	986.72	1887.63
17	32	63	51568.09	56261.64	2738751.76	4747727.28	3.31	6.31	1088.17	1839.07
18	34	67	52996.76	55494.45	2753660.40	5253633.59	3.44	6.71	1166.95	1979.84
19	36	71	57624.35	55947.69	2758926.56	5263336.95	3.83	7.43	1284.07	2175.95
20	38	75	56568.34	59102.58	2778800.45	5301716.10	5.64	7.57	1439.07	2212.07
21	40	79	54963.50	57662.61	2825757.86	5315068.43	4.27	8.21	1544.74	2609.12
22	42	83	56509.21	58231.53	2863842.94	5324454.00	5.25	8.35	1578.19	2524.96
23	44	87	58968.89	57542.70	2873479.75	5416904.70	5.36	8.82	1687.76	2830.99

24	46	91	59728.70	58599.92	2896760.29	5482986.91	8.22	10.45	1669.67	2782.95
25	48	95	58862.92	59164.67	2889821.34	5498581.86	5.56	9.80	1780.38	2883.99
26	50	99	60760.31	60723.51	2965156.00	5519391.27	5.77	10.09	1867.06	3105.27
27	52	103	60841.10	61406.30	2981903.56	5556575.51	5.94	10.50	1959.82	3378.35
28	54	107	60877.31	59275.40	2998349.69	5607008.21	6.92	11.27	2225.50	3389.44
29	56	111	62713.08	59290.12	2925903.03	5649771.20	6.56	11.34	2207.58	3956.93
30	58	115	63889.80	60101.96	2987202.04	5804375.86	10.04	14.15	2288.64	3566.23

Anexo 5: Resultado de Turniti

Reporte de similitud

NOMBRE DEL TRABAJO

2023_1_2_Tesis.pdf

AUTOR

Andres Aparcana

RECUENTO DE PALABRAS

27891 Words

RECUENTO DE CARACTERES

146795 Characters

RECUENTO DE PÁGINAS

164 Pages

TAMAÑO DEL ARCHIVO

7.3MB

FECHA DE ENTREGA

Jan 12, 2024 10:17 AM GMT-5

FECHA DEL INFORME

Jan 12, 2024 10:19 AM GMT-5

● 12% de similitud general

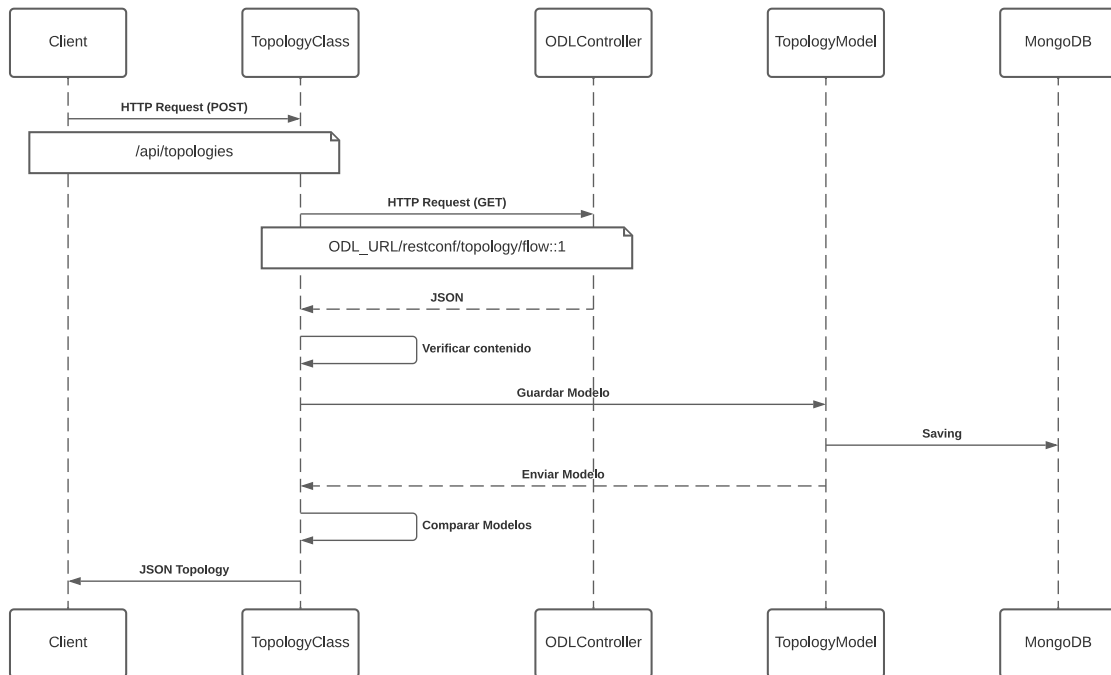
El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base c

- 12% Base de datos de Internet
- Base de datos de Crossref
- 4% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossr

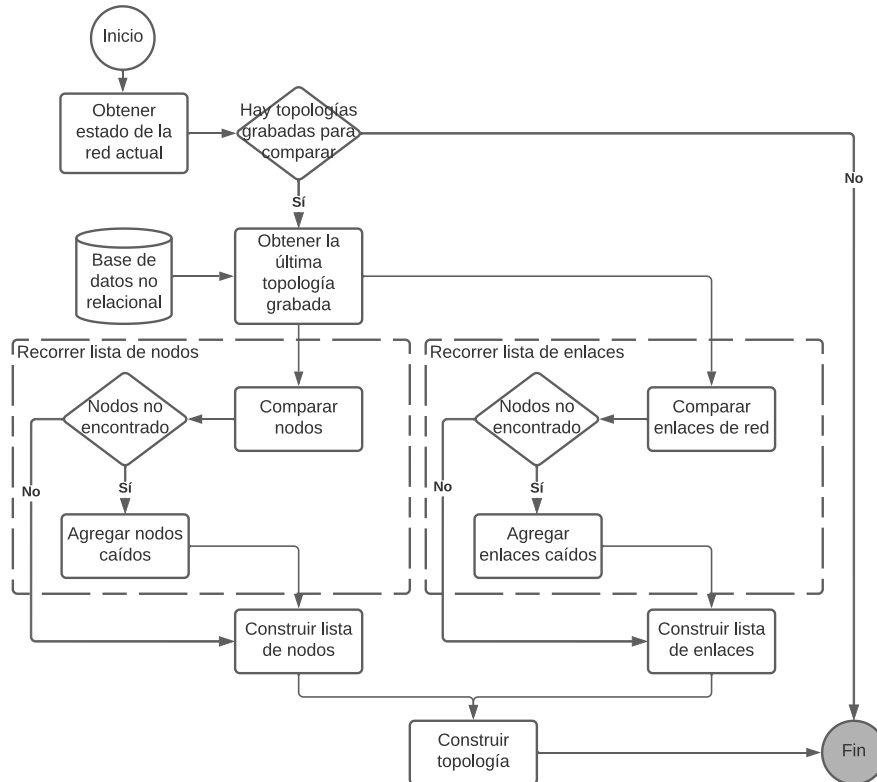
● Excluir del Reporte de Similitud

- Base de datos de trabajos entregados
- Material citado
- Coincidencia baja (menos de 8 palabras)
- Material bibliográfico
- Material citado
- Bloques de texto excluidos manualmente

Anexo 6: Diagramas de detección de fallos de enlace



(a) Diagrama de Actividades de recolección de estados de red



(b) Diagrama de flujo de detección de fallos de enlace

Anexo 7: Carta de Autorización



UNIVERSIDAD NACIONAL DE INGENIERIA

**Instituto Nacional de Investigación y
Capacitación de Telecomunicaciones**



“Año del Fortalecimiento de la Soberanía Nacional”

CARTA DE AUTORIZACIÓN

Lima, 22 de Noviembre del 2022

El que subscribe, en representación de **UNIDAD EJECUTORA 002 – INICTEL-UNI** con RUC N° **20514761826**.

Autoriza:

Al Sr. (a) **ANDRES JUNIOR APARCANA TASAYCO**, identificado con DNIN° **72527217**, estudiante de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Autónoma del Perú, para el uso y publicación de resultados para su proyecto de tesis: “**SISTEMA DE MONITOREO DE RED-NMS DE DETECCIÓN DE FALLOS DE ENLACE EN EL PROCESO DE MONITOREO DE REDES DEFINIDAS POR SOFTWARE**” en el laboratorio de redes del **INICTEL-UNI**.

Se emite esta carta para fines académicos para el desarrollo de su tesis.

Atentamente,

Ing. Daniel Díaz Ataucuri
Director Ejecutivo (e)
INICTEL-UNI