

UN AMBIENTE DE PRUEBA VSDN SOBRE LA PLATAFORMA TELCO 2.0



Trabajo de Grado

Cristian Felipe Velasco Molano
Jonathan Dannilo Arcila Muñoz

Director: PhD. Oscar Mauricio Caicedo Rendón

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022

UN AMBIENTE DE PRUEBA VSDN SOBRE LA PLATAFORMA TELCO 2.0

Cristian Felipe Velasco Molano
Jonathan Dannilo Arcila Muñoz

Trabajo de Grado presentado a la Facultad de Ingeniería Electrónica
y Telecomunicaciones de la Universidad del Cauca para obtener el
título de Ingeniero en Electrónica y Telecomunicaciones

Director: Ph.D. Oscar Mauricio Caicedo Rendón

*Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022*

Agradecimientos

En primer lugar, queremos agradecer y expresar nuestra profunda gratitud a nuestros padres, abuelos y familiares en general por su apoyo incondicional y acompañamiento en el cumplimiento de nuestras metas. Para ellos, que han hecho posible nuestra formación personal y profesional, les dedicamos el presente trabajo de grado. Agradecemos también a nuestros amigos con los que compartimos esta etapa de nuestra vida, los que siempre estuvieron presentes en los momentos más necesarios y nos dieron la fuerza y la confianza necesaria para terminar este trabajo. Agradecemos a nuestros profesores que nos acompañaron en este proceso y que aportaron lo mejor de ellos en su enseñanza profesional. Por último, agradecemos a nuestro asesor académico, el Profesor Ph.D. Oscar Mauricio Caicedo Rendón, por su apoyo en la dirección de este trabajo de grado.

Resumen

La constante evolución de herramientas *hardware* y *software* brindan diferentes utilidades para el acceso al consumo de servicios de maneras más asequible. Las arquitecturas de red, se mantienen en constante crecimiento, tanto en sus tecnologías como en sus implementaciones, llegando a converger en conceptos de virtualización para dar como resultado nuevas arquitecturas de red.

A medida que las redes informáticas crecen y se vuelven más complejas, existe la necesidad de un nuevo tipo de enfoque simple para configurarlas. Las redes definidas por *software* se han convertido en una arquitectura de red prometedora, eliminando el plano de control de los nodos individuales y centralizando el control de la red mediante la gestión de tráfico basada en el flujo. Las *vSDN* son arquitecturas de red programables que pretenden disminuir los gastos operativos y de capital.

La necesidad de gestión de un entorno configurable de las arquitecturas de red visualizadas, es un objetivo resaltante, por ello la creación de un ambiente de pruebas *vSND* se convierte en un desarrollo factible a realizar, con el fin de garantizar el despliegue de diferentes topologías de red e implementar las diferentes utilidades para el control del tráfico y revisión de la valorización de desempeño de las arquitecturas deseadas. Este documento presenta un estudio, diseño, construcción e implementación de un ambiente de pruebas *vSDN*.

Tabla de Contenido

Lista de Figuras	vi
Lista de Tablas	ix
Lista de Abreviaturas	x
1 Introducción	1
1.1 Planteamiento del problema	1
1.2 Objetivos	5
1.2.1 General	5
1.2.2 Específicos	5
1.3 Contribuciones a la investigación	6
1.4 Estructura del documento	7
2 Marco Teórico	8
2.1 Redes definidas por <i>software</i>	8
2.1.1 Controlador de red	10
2.2 Virtualización de funciones de red	11
2.3 Redes Definidas por software virtualizadas	14
2.4 Telco 2.0	14
2.5 SCRUM	15
2.5.1 Equipos Scrum	16
2.5.2 Eventos o <i>Sprint</i>	17
2.5.3 Artefactos de Scrum	17
3 Trabajos Relacionados	19

3.1	Ambientes de prueba para redes definidas por <i>software</i>	19
3.2	Ambientes de prueba para redes definidas por <i>software</i> soportados sobre <i>OpenStack</i>	21
3.3	Ambientes de prueba para redes definidas por <i>software</i> virtualizadas .	23
3.4	Brechas	25
4	Diseño	27
4.1	Escenario de Motivación	27
4.2	Diseño del Ambiente de pruebas para redes definidas por <i>software</i> virtualizadas	28
4.2.1	Etapa <i>i</i> - Inicio	29
4.2.2	Etapa <i>ii</i> - Planificación y estimación	34
4.2.3	Etapa <i>iii</i> - Implementación	43
4.2.4	Etapa <i>iv</i> - Revisión y retrospectiva	50
4.2.5	Etapa <i>v</i> - Lanzamiento	53
5	Evaluación y análisis	60
5.1	Entorno de trabajo <i>Web</i>	60
5.2	Emulador de Red	61
5.3	Analizador de protocolos	62
5.4	Generador de tráfico	62
5.5	Ejecución y pruebas	64
5.5.1	Rendimiento de la red	65
5.5.2	Análisis de resultados	66
6	Conclusión y Trabajo Futuro	75
6.1	Conclusiones	75
6.2	Trabajo Futuro	77
	Bibliografía	77
A	Historias de usuario	1
B	Bosquejos de las interfaces de usuario	8

TABLA DE CONTENIDO

v

C Interfaces Gráficas de Usuario	18
D Repositorio Telco IMS	27
E Guías de usuario	28

Lista de Figuras

2.1	Arquitectura de las redes definidas por <i>software</i> (adaptado de [32]).	9
2.2	Separación lógica (adaptada de [59]).	12
2.3	Despliegue de un servicio (adaptado de [61]).	12
2.4	Arquitectura de la Virtualización de funciones de red (adaptado de [63]).	13
2.5	Infraestructura Telco 2.0	15
2.6	Metodología <i>Scrum</i>	16
4.1	Lista <i>Sprint</i>	39
4.2	Caso de uso del ambiente de pruebas	44
4.3	Diagrama de clases Telco IMS	46
4.4	Diagrama de componentes ambiente pruebas <i>vSDN</i>	48
4.5	Diagrama de secuencias del ambiente pruebas <i>vSDN</i>	49
4.6	Interfaz de inicio	50
4.7	Interfaz de inicio y diseño libre	51
4.8	Interfaz para el diseño de topologías preestablecidas	52
4.9	Interfaz para la configuración de tráfico de la red	52
4.10	Interfaz para el despliegue de las métricas de desempeño	53
4.11	Interfaz de acceso al ambiente de pruebas <i>vSDN</i>	54
4.12	Interfaz de inicio y de diseño libre	55
4.13	Interfaz de diseño de topologías predeterminadas	55
4.14	Interfaz de notificación del estado de la emulación	56
4.15	Interfaz de configuración del tráfico	56
4.16	Interfaz de despliegue de resultados	57

5.1	prototipo final del ambiente de pruebas <i>vSDN</i>	63
5.2	Métricas de desempeño	65
5.3	Topología <i>SDN</i> en configuración simple de 20 <i>hosts</i>	67
5.4	Vista general del desempeño de la red <i>SDN</i>	67
5.5	Desempeño de la red <i>SDN</i> vista desde el cliente	68
5.6	Desempeño de la red <i>SDN</i> vista desde el servidor	69
5.7	Vista general del desempeño de la red <i>SDN</i>	70
5.8	Desempeño de la red <i>SDN</i> vista desde el servidor	70
5.9	Desempeño de la red <i>SDN</i> vista desde el cliente	71
5.10	Validación del prototipo final	74
A.1	Acceso al ambiente de pruebas <i>vSDN</i>	1
A.2	Herramienta para el diseño libre	2
A.3	Herramienta para el diseño predeterminado	2
A.4	Ejecución emulación de red	3
A.5	Detención emulación de red	3
A.6	Configuración de parámetros de tráfico	3
A.7	Resultados de tráfico	4
A.8	Controladores de red	4
A.9	Métricas de desempeño individuales	5
A.10	Estado de emulación	5
A.11	Herramientas de acción	5
A.12	Información	6
A.13	Parametrización elementos de red	6
A.14	Parametrización elementos de red	7
B.1	Interfaz de menú de archivo	9
B.2	Interfaz de menú de edición	9
B.3	Interfaz de menú de emulación	10
B.4	Interfaz de menú de ayuda	10
B.5	Interfaz de configuración elemento <i>host</i>	11
B.6	Interfaz de configuración elemento controlador	11
B.7	Interfaz de configuración elemento conmutador <i>SDN</i>	12
B.8	Interfaz de configuración elemento puerto de red	12

B.9	Interfaz de configuración elemento enlace	13
B.10	Interfaz de inicio de emulación	13
B.11	Interfaz de inicio de emulación	14
B.12	Interfaz de notificación de estado de la emulación	14
B.13	Interfaz de notificación de estado de la emulación	15
B.14	Interfaz de analizador de protocolos	15
B.15	Interfaz del controlador <i>SDN</i> externo	16
B.16	Interfaz de documentación - repositorio del ambiente	16
B.17	Interfaz de documentación - guía de usuario del ambiente	17
B.18	Interfaz de documentación - descripción del ambiente	17
C.1	Interfaz de inicio del ambiente de pruebas <i>vSDN</i>	18
C.2	Interfaz de selección de topologías <i>vSDN</i> prediseñadas.	19
C.3	Interfaz de menú de edición.	19
C.4	Interfaz de menú de emulación.	20
C.5	Interfaz de menú de ayuda.	20
C.6	Interfaz menú de configuración del controlador.	21
C.7	Interfaz menú de configuración del <i>switch</i>	21
C.8	Interfaz menú de configuración del <i>host</i>	22
C.9	Interfaz menú de configuración del puerto	22
C.10	Interfaz menú de configuración del enlace	23
C.11	Interfaz menú de configuración IP servidor gráfico del cliente	23
C.12	Interfaz de notificación del estado de la emulación	24
C.13	Interfaz de notificación del estado de la emulación	24
C.14	Interfaz del analizador de protocolos	25
C.15	Interfaz gráfica del controlador externo <i>ODL</i>	25
C.16	Interfaz gráfica del controlador externo <i>ONOS</i>	26
E.1	Guías de usuario multimedia del ambiente de pruebas <i>vSDN</i>	28

Lista de Tablas

3.1	Características de los trabajos relacionados	25
4.1	Roles del ambiente de pruebas <i>vSDN</i>	30
4.2	Cuestionario preliminar análisis de requisitos de usuario	31
4.3	Análisis preliminar de requisitos y experiencia de usuarios	32
4.4	Análisis de la literatura, requisitos técnicos de un ambiente de pruebas <i>vSDN</i>	33
4.5	Relación de requisitos	37
5.1	Cuestionario validación de prototipo	73

Lista de Abreviaturas

ICT	Information and Communication Technology
ONF	Open Networking Foundation
SDN	Software Defined Networking
NFV	Network Functions Virtualization
vSDN	Virtual Software Defined Networking
QoS	Quality of Service
GENI	Global Environment for Networking Innovation
API	Application Programming Interfaz
GUI	Graphical User Interface
ONOS	Open Network Operative System
GpENI	Great Plains Environment for Network Innovation
VLAN	Virtual Local Area Network
VXLAN	Virtual Extensible Local Area Network
IP	Internet Protocol
CAPEX	Capital Expenditure
OPEX	Operational Expenditures
VNF	Virtualized Network Function
ETSI	The European Telecommunications Standards Institute

GB	Giga Byte
TB	Tera Byte
MSA	Modular Smart Array
DNS	Domain Name System
GUI	Graphical User Interface
LAN	Local Area Network
OCF	Ofelia Control Framework
OS	Operating System
LXC	LinuX Containers
KVM	kernel-Based Virtual Machine
CORE	Common Open Research Emulator
DC	Data Center
VM	Virtual machine
VMs	Virtual machines
CPU	Central Processing Unit
RAM	Random Access Memory
RTT	Round Trip Time
RTT	Round Trip Time Variation
PMTU	Path Maximum Transmission Unit
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
D-IGT	(Distributed Internet Traffic Generator
TSP	(Traffic Specification Protocol

Capítulo 1

Introducción

1.1 Planteamiento del problema

Las arquitecturas de red tradicionales a pesar de solventar las necesidades actuales (creación y manejo de entornos de comunicaciones) son muy complejas y poco flexibles [1]. La velocidad de evolución de nuevas aplicaciones y tecnologías avanza, conllevando al desarrollo de nuevas arquitecturas de red enfocadas en suplir dichas necesidades y crear una visión de nuevos servicios [2].

Gracias a un amplio esfuerzo de la industria de las tecnologías de la información y las comunicaciones (*ICT - Information and Communication Technology*) [3], encabezado por la fundación de redes abiertas (*ONF - Open Networking Foundation*) [4], nacen las redes definidas por *software* (*SDN - Software Defined Networking*) como respuesta a la necesidad de una nueva arquitectura de red flexible, eficiente y escalable, proyectando la construcción de redes altamente programables y adaptables a las necesidades cambiantes de los servicios de red [5]. En *SDN*, los dispositivos de red son responsables de encaminar físicamente los paquetes, mientras que el control del enrutamiento es realizado por medio de *software*. Al centralizar la inteligencia de la red en controladores (visión global de la red), la *ONF* estandarizó el protocolo *OpenFlow* [6] como uno de los protocolos encargados de estructurar la comunicación entre los planos de control y datos de una arquitectura *SDN* [7].

Por otro lado, la virtualización de funciones de red (*NFV*, *Network Functions Virtualization*) aborda las limitaciones de las arquitecturas de red no flexibles, aprovechando la tecnología de virtualización que ofrece nuevas formas de diseño, despliegue y gestión de servicios de red [8]. La tecnología *NFV* mejora la velocidad, la automatización y la gestión de redes, mediante la independencia de las funciones y servicios de red (proxies, balanceadores de carga, cortafuegos, etc.) de los dispositivos convencionales y migrándolos a elementos *software*. Al estar las tecnologías *NFV* y *SDN* profundamente relacionadas con la convergencia de redes virtualizadas, y resaltando que las *SDN* benefician el diseño y despliegue de redes [9], es pertinente destacar la oportunidad de cambiar la forma de concebir y construir redes, aumentar los beneficios, reducir su complejidad y posibilitar la implementación de redes inteligentes a través de las redes definidas por *software* virtualizadas (*vSDN*) [10].

Para la investigación, desarrollo e innovación (I+D+i) de *SDN* diferentes entidades han aprovechado los ambientes de prueba [11] como plataformas para el estudio, el fortalecimiento, la experimentación de conceptos y el prototipado de servicios [12] [13]. *NFV* posibilita el despliegue de ambientes de prueba para *SDN* virtualizando los componentes y emuladores de esta arquitectura en entornos dispuestos en la nube [14], estas funciones permiten así la evaluación de calidad de servicio (*QoS*, *Quality of Service*) de las redes [15]. Es destacable que *vSDN* plantea la idea de redes inteligentes, distribuidas y dinámicas capaces de prestar un mejor servicio a las aplicaciones del cliente y la asignación de recursos según las necesidades requeridas, este tipo de redes virtualizadas aportan un entorno flexible a la I+D+i mediante la experimentación de proyectos *ICT* a través de la simulación o emulación [16].

A nivel Internacional existen ambientes de prueba *SDN* a gran escala como *GENI* [17], *GpENI* [18], *OpenStackEmu* [19] y *Chameleon Cloud* [20], estos entornos son implementados bajo tecnologías y protocolos de red emergentes como *NFV*, *SDN* y *OpenFlow*; los escenarios nombrados están constituidos por infraestructuras de alta velocidad y nodos programables. Los ambientes mencionados ofrecen recursos de red para la experimentación a través de una interfaz de programación de aplicación (*API* - *Application Programming Interface*) de acceso abierto a investigadores registrados [21]. Los entornos como *GENI* y *GpENI* que están enfocados en la innovación y experimentación de la arquitectura de internet y su administración autónoma,

permiten el diseño y evaluación de protocolos y servicios distribuidos en conjunto de la administración de contenido y el despliegue de servicios de red. No obstante, los ambientes citados presentan una limitación principal debido a la infraestructura que poseen, la cual está basada en equipos propietarios [22] que dificultan e incrementan los costos de acceso. El ambiente de pruebas *Chameleon cloud* implementa el diseño, el desarrollo y la experimentación de tecnologías innovadoras de virtualización y arquitectura en nube [23] a través de un ecosistema experimental de redes altamente programables (Redes ruteadas, *VLAN*, *VXLAN* y *OpenFlow*). Sin embargo, las configuraciones del plano de control (Protocolos de enrutamiento, señalización y/o resolución de direcciones *IP*, etc.) del ambiente está limitado a parámetros preestablecidos, por lo cual, los experimentadores y sus implementaciones están condicionadas a las redes predisuestas por el ambiente. El ambiente *OpenStackEmu* propone una plataforma de emulación a gran escala con el objetivo de potenciar la emulación de red en nube tan simple como la simulación de red; basándose en el sistema operativo para la administración de centros de datos, como lo es *OpenStack* [24], el ambiente de prueba virtualiza eficientemente varios nodos de emulación física en cientos de nodos de emulación virtual e implementa de forma síncrona, dinámica y precisa la emulación de red en tiempo real, este tipo de despliegue conlleva a nuevas estrategias de enrutamiento y control de congestión de redes en nube [25]. A nivel nacional es importante resaltar que no se cuenta con ambientes de prueba para el estudio, experimentación e innovación de entornos *SDN*. Es destacable que la carencia de estos ambientes influye negativamente en el aprendizaje y formación de los estudiantes en las temáticas enfocadas en nuevas tecnologías de red, como lo son las *SDN*.

La facultad de ingeniería electrónica y telecomunicaciones (FIET) de la Universidad del Cauca actualmente cuenta con el laboratorio de telecomunicaciones Telco 2.0 (equipos de procesamiento y almacenamiento de alta capacidad), asimismo, esta facultad tiene la necesidad de disponer en sus instalaciones de un ambiente de pruebas *vSDN* que permita el fortalecimiento de los procesos académicos e investigativos en este campo de formación e investigación. La carencia de este tipo de escenarios en la FIET impide la familiarización y profundización en los conocimientos necesarios para los desafíos de las nuevas tecnologías, este despliegue impera actual-

mente en las redes de centros de datos que soportan los servicios en nube. Por tanto, es indispensable disponer de un entorno en el que los usuarios puedan acceder a todas las funcionalidades que los ambientes descritos proporcionan, con la capacidad de crear, experimentar y analizar las posibilidades que esta arquitectura *vSDN* presenta. Dadas las condiciones anteriormente mencionadas, el presente trabajo de grado es planteado para dar respuesta al interrogante:

¿Cómo ofrecer un ambiente de prueba *vSDN* en la Plataforma Telco 2.0 de la Facultad de Ingeniería Electrónica de la Universidad del Cauca?

Para encontrar respuesta a la pregunta planteada, el presente informe final presenta el diseño, configuración y emulación de un ambiente de pruebas *vSDN*, enfocado en la I+D+i de servicios y aplicaciones en redes convergentes sobre la plataforma Telco 2.0.

1.2 Objetivos

1.2.1 General

- Proporcionar un ambiente de prueba para *SDN* virtualizadas sobre Telco 2.0.

1.2.2 Específicos

- Diseñar la arquitectura de un ambiente de prueba para *vSDN*.
- Implementar el ambiente de prueba diseñado sobre la plataforma Telco 2.0.
- Evaluar el ambiente de pruebas implementado en términos de métricas de desempeño de la red.

1.3 Contribuciones a la investigación

Las principales contribuciones aportadas en este trabajo de grado son las siguientes:

- Desarrollo de un ambiente de pruebas *vSDN* soportado en TELCO 2.0.
- Evaluación del desempeño de arquitecturas *vSDN* configurables, con adaptabilidad en la gestión del tráfico.

1.4 Estructura del documento

Este documento está dividido en los capítulos descritos a continuación.

- El capítulo 1 presenta la **Introducción**, este capítulo incluye el planteamiento del problema, los objetivos, las contribuciones a la Investigación y la estructura del presente documento.
- El capítulo 2 despliega el **Marco Teórico**, el cual expone la recopilación, revisión e identificación de la información más relevante que soporta nuestro trabajo de grado. Estos temas incluyen la conceptualización de las redes definidas por *software*, la virtualización de funciones de red, las redes definidas por *software* virtualizadas y la plataforma TELCO 2.0.
- El capítulo 3 describe las contribuciones de los trabajos más relevantes encontrados en la literatura, esta selección es realizada respecto a la implementación de un ambiente de pruebas para la investigación, el desarrollo tecnológico y la innovación de las redes definidas por *software* virtualizadas.
- El capítulo 4 implementa el **Diseño de la arquitectura del ambiente de pruebas vSDN** propuesto. Esta sección contiene el escenario motivador del presente trabajo de grado, así como también el diseño de la arquitectura del ambiente de prueba a partir del uso de la metodología de diseño de software ágil *Scrum*.
- El capítulo 5 comenta la **Implementación del ambiente de prueba vSDN diseñado** basada en la virtualización, configuración y despliegue de la interfaz gráfica del ambiente, el emulador de red *Mininet*, el generador de tráfico *Iperf3*, el analizador de protocolos *Wireshark* y los controladores externos *OpenDaylight* y *ONOS*. Asimismo, esta parte describe el diseño de las pruebas realizadas, como también la evaluación y el análisis del rendimiento de las emulaciones ejecutadas en el ambiente de pruebas implementado.
- El capítulo 6 resalta las **Conclusiones** obtenidas en el presente trabajo de grado, así como también las implicaciones dispuestas para el desarrollo de **trabajos futuros**.

Capítulo 2

Marco Teórico

Este capítulo presenta los antecedentes que contribuyen al desarrollo del presente trabajo de grado para la implementación de un ambiente de pruebas *SDN* virtualizado. En primer lugar, el paradigma *SDN* es descrito, así como también sus elementos constitutivos, segundo, el concepto de *NFV* es desplegado con base en la instanciación de los componentes de la arquitectura *SDN* en máquinas virtuales, tercero, el concepto de *vSDN* es introducido. Por último, la infraestructura física de Telco 2.0 es detallada con el fin de implementar el entorno de pruebas *SDN* virtualizado.

2.1 Redes definidas por *software*

SDN es una arquitectura de red fundamentada en la separación de la lógica del control de la red de los enrutadores y conmutadores subyacentes que envían el tráfico [26] [27]. La separación de los planos de control y datos redefinen la concepción de conmutadores de red, especificándolos como dispositivos de reenvío, controlados lógicamente desde un *software* centralizado, simplificando la aplicación de políticas, (re)configuración y evolución de la red [28] [29].

La arquitectura *SDN* puede ser dividida en diferentes planos lógicos e interfaces que facilitan la creación e introducción de nuevas abstracciones en la red. Resaltando los planos de aplicación, gestión, control y datos, e interfaces de frontera norte y sur como sus principales elementos [30] [31] [32] (Ver figura 2.1).

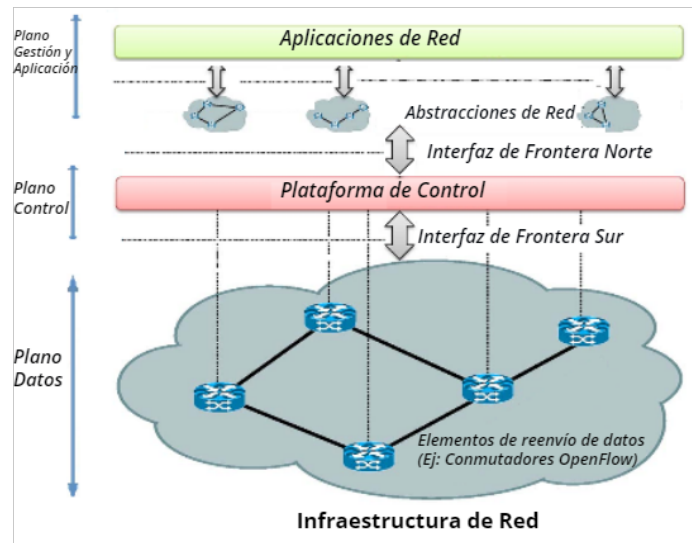


Figura 2.1: Arquitectura de las redes definidas por *software* (adaptado de [32]).

Plano de datos: Conjunto de medios de transmisión y dispositivos de red (repetidor, conmutador, enrutadores, entre otros) encargados del envío y procesamiento de datos de la red [33] [34].

Interfaz de frontera sur: Formaliza la interacción entre los planos de control y datos a través de un protocolo de comunicación entre el controlador de la red y los dispositivos de reenvío establece las reglas de procesamiento de datos [34] [35].

Plano de control: Entidad lógica compuesta por uno o más controladores de red o sistemas operativos *SDN*, capaces de gestionar la red (organizar, actualizar y eliminar las reglas de flujo abierto) basándose en la lógica definida por las aplicaciones de red en ejecución [36] [37].

Interfaz de frontera norte: Ecosistema de *APIs* enfocadas en el desarrollo e interpretación de instrucciones de alto nivel de red utilizada para programar los dispositivos de reenvío [36] [37].

Plano de gestión: Conjunto aplicaciones soportadas por el controlador de red o el sistema operativo *SDN*, capaces de comunicar los comportamientos y recursos necesarios para la administración de la red. Estas aplicaciones están enfocadas en la gestión de calidad de funcionamiento o desempeño, fallas o

averías, configuración, seguridad, contabilidad y facturación [38] [39].

Plano de aplicación: Conjunto de aplicaciones de red que permiten implementar el control de la red y la lógica de operación. Las principales aplicaciones *SDN* son: Aplicaciones de redes híbridas, aplicaciones de ingeniería de tráfico, aplicaciones de redes de centros de datos, aplicaciones de movilidad e inalámbricas y, por último, aplicaciones de seguridad de redes, entre otras [39] [40].

2.1.1 Controlador de red

En la arquitectura *SDN*, los controladores son la parte esencial del plano de control, dado que son los encargados de abstraer la visión global de la red y proveer a las aplicaciones información de los eventos o comportamientos que suceden en la misma. Estos dispositivos permiten el control de todos los elementos *SDN* a través de la ejecución de distintas políticas y la toma de decisiones óptimas en la red. Normalmente, los controladores vienen compuestos por un conjunto de aplicaciones por defecto, como por ejemplo: un *switch* para el aprendizaje y descubrimiento de la red, un *router*, un *firewall* o un balanceador de carga, entre otros [41].

En la actualidad los controladores están caracterizados por las siguientes cualidades [42]:

- Virtualización.
- Soporte de plataforma.
- Código abierto o propietarios.
- Cantidad de *API* y protocolos soportados.
- Manejo de interfaces gráficas de usuario (*GUI*, *Graphical User Interface*).
- Capacidad de procesamiento (Miles de flujos/s o hasta millones de flujos/s).
- Modo de funcionamiento reactivo (La carga de la red aumenta con el número de nodos a los que tengan que enviarse la información).
- Modo de funcionamiento proactivo (La instancia de control anticipa y envía la información nueva y las modificaciones a todos los nodos participantes en

la red).

Sin embargo, dado que controladores están basados en *software*, su ejecución es caracteriza por la facilidad en su desarrollo y despliegue, por lo cual están muy ligados a sus lenguajes de programación (*Java*, *Python*, *C*, *C++*, etc.). Por lo tanto, la elección del lenguaje de programación influye en la seguridad, portabilidad y rendimiento del mismo. Actualmente, los controladores más destacados son: *Beacon* [43], *NOX* [44], *POX* [45], *Floodlight* [46], *Ryu* [47] [48], *OpenDaylight* [49] y *ONOS* [50].

Con base en la literatura [51] [52] [53] [54], es inferible que los controladores *SDN* *ONOS* y *OpenDaylight* son los más destacados debido a que funcionan en varias plataformas, presentan una gran modularidad, implementan una buena función de *GUI* y están bajo la asociación de conocidos proveedores de redes y comunidades de investigación, que permiten un plan de desarrollo claro y una buena documentación.

- **ONOS:** Diseñado principalmente para redes de operadores, es un proyecto modular basado en estándares abiertos para el desarrollo e implementación de programas y bibliotecas de *software* *Open Services Gateway Initiative (OSGi)* [55] y desarrollado en lenguaje *Java*. La arquitectura de *ONOS* es de tipo distribuida y está diseñada para mantener redes de alta velocidad y gran escala. Su principal característica distintiva es la implementación orientada a la administración, configuración y despliegue de nuevos servicios y compatibilidad con las redes híbridas [56].
- **OpenDaylight:** Es un controlador de código abierto escrito en lenguaje *Java*, diseñado principalmente para centro de datos. Entre sus características principales es destacable la arquitectura de tipo distribuido, que permite una alta disponibilidad, además, de una gran capacidad de desarrollo de funciones y aplicaciones debido a que soporta múltiples protocolos de comunicación [57].

2.2 Virtualización de funciones de red

NFV es una tecnología de red capaz de proporcionar agilidad y flexibilidad al despliegue de servicios de red, reduciendo las inversiones de Capital (*CAPEX*, *cap-*

tal expenditure) y gastos operativos (*OPEX, operational expenditures*) [58]. *NFV* desvincula las funciones de red (cortafuegos, inspectores de paquetes, balanceadores de carga, entre otros) del hardware subyacente a través de componentes virtuales llamados funciones de red virtuales (*Virtualized Network Function, VNF*) [59] (Ver Figura 2.2).

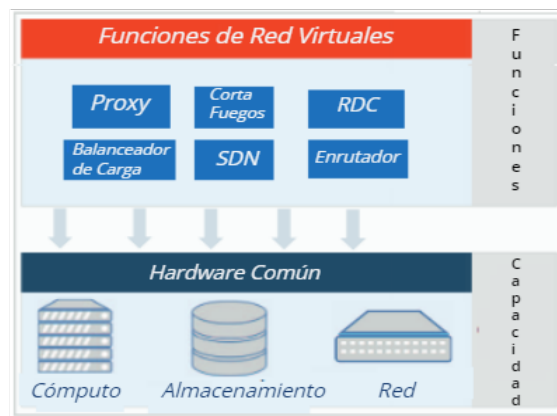


Figura 2.2: Separación lógica (adaptada de [59]).

NFV establece una cadena de servicio como una secuencia de múltiples *VNF* ejecutándose en un orden específico para entregar un servicio [60]. La figura 2.3 muestra un ejemplo de un sistema de seguridad compuesto por tres *VNF* que pueden ser desplegados en una o más máquinas virtuales [61].

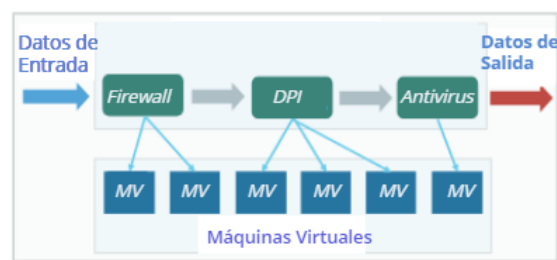


Figura 2.3: Despliegue de un servicio (adaptado de [61]).

El Instituto Europeo de Estándares de Telecomunicaciones (*ETSI, The European Telecommunications Standards Institute*) [62] define al gestor de infraestructura, la capa de virtualización y al orquestador de *NFV* y *VNF* como pilares fundamentales de la arquitectura *NFV* [63] (Ver Figura 2.4).

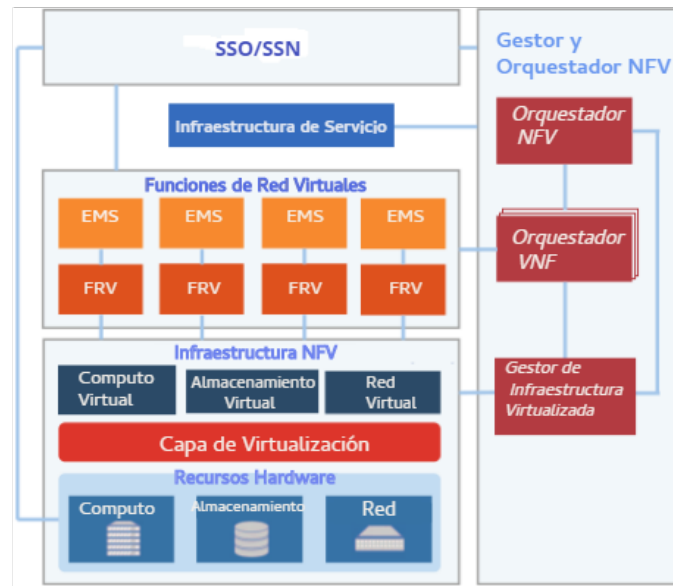


Figura 2.4: Arquitectura de la Virtualización de funciones de red (adaptado de [63]).

Orquestador NNFV: Entidad responsable de gestionar y orquestar los recursos *software* e infraestructura *hardware* virtualizada con el fin de proporcionar servicios de red [64] [65] [66].

Orquestador VNF: Entidad responsable de la instanciación, escalado, terminación y actualización de los eventos de cada *VNF*, ejecutados durante un período determinado [64] [65] [66].

Gestor de Infraestructura: Entidad encargada de virtualizar, gestionar o controlar los recursos configurables de computación, red y almacenamiento necesarios para su interacción con las *VNF* [64] [65] [66].

Capa de Virtualización: Entidad encargada de proporcionar una asignación funcional entre los recursos físicos y la infraestructura virtualizada para cada *VNF* [64] [65] [66].

2.3 Redes Definidas por software virtualizadas

SDN y *NFV* son consideradas tecnologías totalmente independientes, pero inherentemente complementarias, tanto así que su integración *NFV/SDN* o *vSDN* proporcionan muchos beneficios al área de las telecomunicaciones resaltando la transformación de nodos de red en micro centros de datos (siendo estos capaces de alojar *VNF* y aplicaciones del cliente). Las *vSDN* logran concebir la red como plataformas distribuidas y dinámicas capaces de prestar un mejor servicio a las aplicaciones del cliente y la asignación de recursos según las necesidades [67].

Desde la perspectiva de *NFV*, las *vSDN* están destinadas a optimizar el despliegue de funciones de red (cortafuegos, *DNS*, balanceadores de carga, etc.), como también mejorar la eficiencia y flexibilidad de los servicios del plano de control [68]. Desde la perspectiva *SDN*, las *vSDN* centralizan la optimización de las redes subyacentes al tiempo que garantizan la entrega y calidad del tráfico entre las funciones virtualizadas [69]. Por lo tanto, la integración de *NFV* y *SDN* permite a las *vSDN* tomar decisiones de control de red inteligente, logrando un comportamiento de red dinámico y una mejora en el uso de recursos de la red tanto físicos como virtuales [70]. Las *vSDN* han evolucionado y son el componente principal para el diseño, desarrollo y despliegue de varios ambientes de prueba, alentando a numerosos investigadores a construir sus prototipos y experimentos o configurar novedosas arquitecturas y servicios de red de próxima generación [71].

2.4 Telco 2.0

Telco 2.0 es una infraestructura de la FIET de la Universidad del Cauca para la investigación e innovación de servicios de Telecomunicaciones en la nube, cuyo propósito es convertirse en uno de los centros de investigación tecnológico más importantes de la región. En particular, esta plataforma brinda servicios de gestión, almacenamiento y procesamiento de máquinas virtuales.

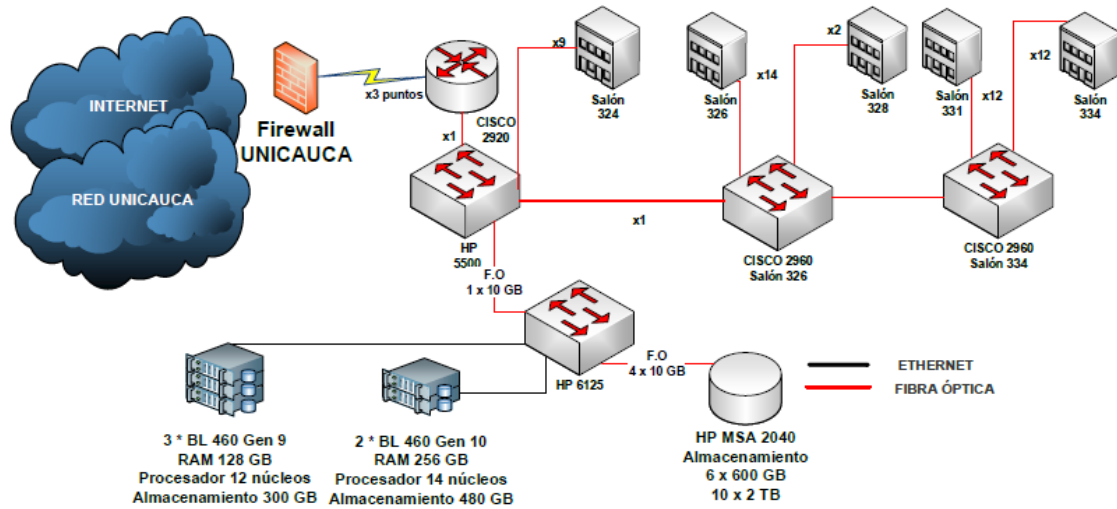


Figura 2.5: Infraestructura Telco 2.0

Actualmente, el laboratorio Telco 2.0 está compuesto por cinco servidores *Blade Hewlett Packard*, tres de ellos con 128 GB de RAM, 2 procesadores de 12 núcleos cada uno y almacenamiento de 300 GB. Asimismo, dos servidores con 256 GB de RAM, 2 procesadores de 14 núcleos cada uno y almacenamiento de 480 GB. Igualmente, cuenta con unidades de almacenamiento *MSA* de un total de 23,6 TB. En la figura 2.5 se puede observar la infraestructura física de la plataforma Telco 2.0.

2.5 SCRUM

Scrum es un marco de trabajo fundamentado en los conceptos de transparencia, inspección y adaptación [72], garantizando la visibilidad y detección de variaciones indeseables en el proceso de desarrollo, como también, la realización de los ajustes pertinentes que permitan minimizar el impacto de las mismas sobre el resultado [73]. En pocas palabras, *Scrum* ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables a problemas complejos [73].

Esta metodología logra la colaboración eficaz de equipos en proyectos mediante la definición de un conjunto de reglas, artefactos y roles que generan la estructura necesaria para su correcto funcionamiento [74]. Como lo indica la figura 2.6, *Scrum*

está constituido por los equipos *Scrum*, los eventos o *Sprint* y los artefactos *Scrum* [75].

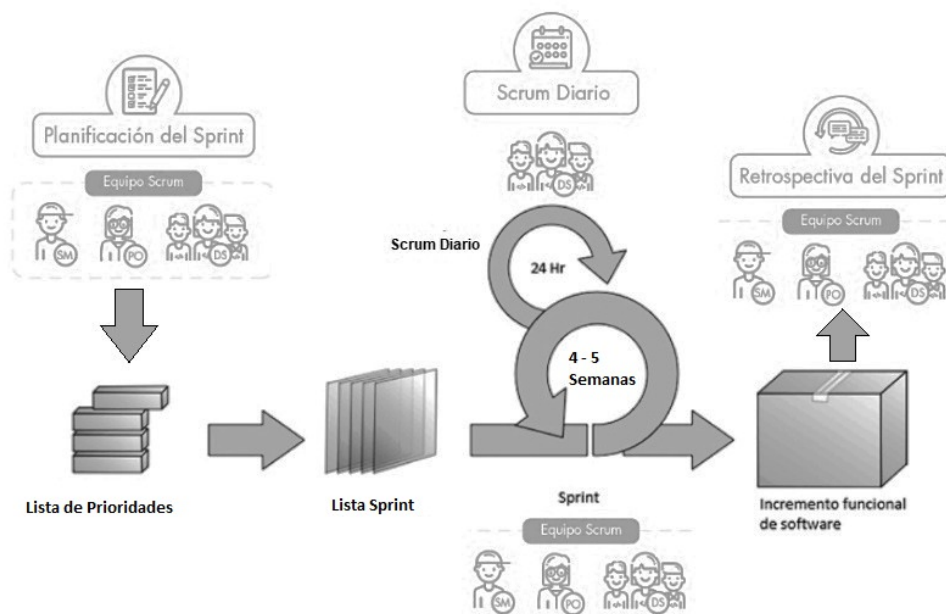


Figura 2.6: Metodología *Scrum*

2.5.1 Equipos Scrum

Los llamados equipos *Scrum* son caracterizados por la multifunción, la autogestión y el trabajo iterativo como sus pilares fundamentales [76]. La multifunción permite a los integrantes estar capacitados en todos los conocimientos necesarios para llevar a cabo cada tarea; la autogestión permite seleccionar la mejor forma de desarrollar el proyecto sin seguir lineamientos de terceros no pertenecientes al equipo que carezcan de contexto; y el trabajo iterativo permite la entrega del producto en iteraciones y cada iteración crea nuevas funcionalidades o modifica las que el dueño del producto requiera. Los equipos Scrum están constituidos por tres roles diferentes: Líder *Scrum*, propietario del producto y desarrolladores [77].

- **Líder:** Es el encargado de asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el jefe del equipo, pero no gestiona el desarrollo [78].

- **Propietario del producto:** Es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo, entre sus funciones tiene la gestión de la lista ordenada y priorizar las funcionalidades requeridas por el cliente [79].
- **Desarrolladores:** Tienen como responsabilidad convertir los requerimientos del cliente en iteraciones funcionales del producto. El tamaño óptimo del equipo está entre tres y nueve personas.

2.5.2 Eventos o *Sprint*

Scrum define un evento o *Sprint* como una ventana de tiempo donde es creada una versión utilizable del producto, cada *Sprint* es considerado como un proyecto independiente y su duración máxima es de un mes [80]. Los *Sprint* están definidos por las siguientes etapas: Planeación, *Scrum* diario, revisión y retrospectiva [81].

- **Planeación:** Define el plan de trabajo, es decir, el diseño del sistema y la estimación de la cantidad de trabajo [81].
- ***Scrum* diario:** Etapa realizada por los desarrolladores cada día, con el fin de describir las metas alcanzadas desde la última reunión, esta tarea es realizada antes de la siguiente y en ellas son destacados los obstáculos presentados [82].
- **Revisión:** Etapa de gran importancia para los siguientes *Sprint*, debido a que en esta, el propietario del producto revisa los requisitos realizados, identifica los que no se realizaron y discute acerca de los atrasos del producto. El equipo de desarrollo presenta los problemas encontrados, la manera en que fueron resueltos, finalmente, expone el producto y su funcionamiento [82].

2.5.3 Artefactos de Scrum

Los artefactos *Scrum* son derivados de las actividades del marco de trabajo que brindan dirección y transparencia al equipo. *Scrum* los define como: Lista de prioridades, lista *Sprint*, monitoreo de progreso e incremento [82] [83].

- **Lista de prioridades:** Es una lista total de requerimientos definida, actualiza y ordenada por el dueño del producto. El orden de estas relaciones dependen

del valor, riesgo, prioridad y necesidad del requerimiento [83].

- **Lista *Sprint*:** Es un subconjunto de requerimientos con mayor prioridad de la *lista del producto* y aquellos que quedaron por resolver en el *Sprint* anterior [83] [84].
- **Monitoreo de Progreso:** Es refiere a la suma del trabajo que falta por realizar en el *Sprint*. Está actividad puede presentarse en cualquier momento, permitiendo al dueño del producto evaluar el progreso del desarrollo [84].
- **Incremento:** Recopila la suma de todos los ítems finalizados de la lista *Scrum*. Si existen ítems incompletos son devueltos a la lista del producto con una prioridad alta para que sean incluidos en el siguiente *Sprint* [84].

Resumiendo, al conocer las ventajas de las redes definidas por *software* con la definición de diferentes controladores de red, y teniendo en cuenta los alcances de la virtualización de funciones de red, es evidente que la implementación de *SDN* virtualizadas componen los lineamientos para el desarrollo y despliegue de diferentes ambientes de prueba, con la construcción de prototipos y experimentos de arquitecturas y servicios de red de siguiente generación. Por otra parte, *Scrum* expone un marco de trabajo adaptativo y ágil para el despliegue de diferentes procesos de desarrollo.

El laboratorio de Telco 2.0 cuenta con las capacidades competentes para la experimentación y correcto despliegue de diferentes proyectos que un investigador desee llevar a cabo.

Capítulo 3

Trabajos Relacionados

Este capítulo describe los trabajos relevantes encontrados en la literatura referentes al despliegue de ambientes de prueba académicos y/o comerciales, los cuales están enfocados en la investigación, desarrollo e innovación de redes virtuales definidas por *software*. En primer lugar, los trabajos relacionados con el uso de *SDN* son presentados, en segundo lugar, los proyectos relacionados con ambientes de prueba *SDN* soportados sobre *OpenStack* son expuestos, en tercer lugar, los desarrollos catalogados como ambientes de prueba *vSDN* son comentados. En cuarto y último lugar son definidas las brechas de los despliegues mencionados y cómo este trabajo de grado cierra esas limitaciones.

3.1 Ambientes de prueba para redes definidas por *software*

El trabajo “*Global Environment for Network Innovations o GENI: A federated test-bed for innovative network experiments*” [17] plantea un ecosistema de ambientes de prueba virtuales distribuidos para la experimentación de *SDN* a gran escala, promoviendo la innovación en ciencias de la computación, seguridad, servicios y aplicaciones de red [85]. *GENI* proporciona una profunda capacidad de programación de red, permitiendo a administradores definir de manera centralizada las políticas de seguridad, gestión y control de la misma; y a los usuarios el acceso a

los recursos requeridos para desplegar sus diferentes experimentaciones a través de las *API* de *GENI* [86] [87]. Entre sus proyectos de investigación más destacados están: *(i)* *Nation Wide OF Network*, proyecto encargado de administrar y enlazar diferentes laboratorios a nivel mundial a través de enlaces *LAN* y *VLAN*, ofreciendo a los usuarios *GENI* redes adaptadas a sus requisitos de experimentación [88]. *(ii)* *Mobility First Future Internet Architecture*, proyecto encargado de desplegar experimentos y pruebas de nuevos algoritmos y protocolos de redes móviles a nivel mundial [89] [90].

El trabajo “*The Great Plains Environment for Network Innovation o GpENI testbed: Network infrastructure, implementation experience, and experimentation*” [18] describe un entorno de red programable centrado en una red óptica regional. Este entorno está conformado por un grupo de nodos interconectados físicamente por un conmutador *Netgear Gigabit Ethernet* [91] con el objetivo de permitir la capacidad de programación en todas las capas de red (aplicación, transporte, red e incluso la óptica) [92]. Los investigadores pueden crear topologías virtuales arbitrarias e iniciar el proceso de automatización de enrutamiento sobre experimentos aleatorios y flexibles [93].

El trabajo “*Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed*” [94] establece un proyecto de investigación europeo destinado a crear un gran ambiente de pruebas distribuido para redes *SDN* basadas en *OpenFlow*, aportando su propia infraestructura pública (basada en la virtualización de servidores y *switches OpenFlow* comerciales) y la versión de código abierto de su marco de trabajo y control (*OCF, OFELIA Control Framework*) [95]. El ambiente permite a los diferentes usuarios realizar sus experimentos en paralelo, además de facilitar la creación de topologías de red virtuales configurables sobre un amplio número de conmutadores físicos *OpenFlow*; el entorno posibilita el instanciamiento de sus propios controladores *SDN* en un conjunto de máquinas virtuales que desempeñan el papel de generadores de tráfico [96].

En resumen, los trabajos [17] [18] [94] describen destacados ambientes de prueba *SDN*; a partir de la diversidad de estilos, tamaños y duración de sus experimentaciones de red en diferentes interfaces abiertas, estos ambientes son capaces de permitir la experimentación separada en diferentes entornos virtualizados, sin embargo, la

principal limitación de estos proyectos es la no adopción de una interfaz de experimentación de usuario única, ya que al no integrar una plataforma en la que converjan todos los entornos necesarios y sus configuraciones, hacen que el experimentador se desplace entre diferentes sistemas manualmente, exponiéndose a la pérdida de información, además el investigador debe conocer los protocolos de interacción entre estos sistemas, obligándolo entre otras cosas, a disponer de tiempo de experimentación por el tiempo de adaptabilidad a cada sistema en los diferentes proyectos que desarrolle, limitando así el cumplimiento de los estándares en la industria y la academia, cabe resaltar, que estos ambientes de prueba no están disponibles a investigadores no federados o miembros, no siendo la FIET la excepción.

3.2 Ambientes de prueba para redes definidas por *software* soportados sobre *OpenStack*

El trabajo “*Using SDN Technology to Mitigate Congestion in the OpenStack Data Center Network*” [97] está enfocado en la experimentación de métodos de mitigación para la congestión del tráfico de la red, mejorando el rendimiento mediante el diseño e implementación de un ambiente de prueba para una arquitectura *SDN*. Utilizando el controlador *Floodlight* y la migración a sistemas de virtualización dinámica *VMLive* [98] como método de mitigación, el ambiente de prueba mejora la conexión de esta red sobre la infraestructura de *OpenStack*, asimismo, permite a los experimentadores monitorizar periódicamente el ancho de banda de todos los conmutadores de la red y detectar la congestión de tráfico de manera oportuna.

“*Next Generation Clouds, The Chameleon Cloud Testbed, and Software Defined Networking (SDN)*” [20] establece un ambiente de prueba *SDN* enfocado en la experimentación e innovación de diferentes prototipos de arquitectura, protocolos y tecnologías de red en la nube [99]. *Chameleon Cloud* es un ambiente de prueba compuesto por 18,000 núcleos de procesamiento en más de 500 nodos de red en la nube, con 5 petabytes de almacenamiento. Para abordar los desafíos de eficiencia, escalabilidad, optimización y seguridad de sistemas operativos adaptativos, *Chameleon Cloud* permite integrar y configurar conjuntos de recursos y atributos en la nube, implementando parametrizaciones de red predefinidas o personalizadas en sus

diferentes experimentaciones [100].

“*Design and deployment of OpenStack-SDN based testbed for EDoS*” [101] proporciona un ambiente de prueba para la experimentación de futuros estándares de red que aprovechan la tecnología *SDN*. Este trabajo plantea los parámetros de configuración e integración de *OpenDayLight* y la plataforma *OpenStack* para el despliegue de *SDN*, además, este ambiente permite a los investigadores analizar el impacto de la Negación Económica de Servicio (*EDoS*, *Economic Denial of Service*) en entornos de producción.

“*Estudio de una solución OpenStack integrada con SDN*” [102] expone el análisis de la tecnología *Cloud* y su integración con el paradigma *SDN*. Este aporte está dividido en dos fases, en la primera realiza el despliegue de *OpenStack* a través de una máquina virtual utilizando tecnologías de virtualización de núcleo (*LXC - LinuX Containers*) y virtualización de contenedores (*KVM - Kernel based Virtual Machine*), la segunda, estudia la tecnología *SDN* y la integración del controlador con *OpenStack*, definiendo las decisiones en diseño, guías de instalación y soluciones a errores encontrados.

“*OpenStack and Software Defined Networking, the enormous potential of open source software collaboration*” [103] plantea como objetivo teórico el estudio de las tecnologías emergentes (*OpenStack*, *SDN*, *OpenDayLight*), la descripción de las arquitecturas y el despliegue de servicios centrales, resaltando el alto crecimiento en las comunidades de código abierto. Los estudios expuestos proporcionan parámetros de diseño e implementación general de distintos ambientes de prueba soportados en *OpenStack* y permiten una mejor estandarización para los ambientes de prueba *SDN*.

En resumen, el trabajo [97] resalta el rendimiento del controlador *Floodlight* como método de mitigación, no obstante, demuestra la disminución del rendimiento cuando el patrón de tráfico en la red sobre la infraestructura de *OpenStack* es robusto, impidiendo que un entorno de experimentación maneje rutas de flujo de alta densidad. El trabajo [20] está enfocado en la integración *OpenStack* y el plano de control de *Chameleon Cloud*, con el fin de gestionar recursos físicos y virtuales sin alterar el rendimiento de la red requerida experimentalmente, sin embargo, los

usuarios carecen de configuraciones flexibles en *OpenStack*, restringiendo procesos fundamentales como la parametrización de políticas, asignación e integración de recursos en distintos periodos de tiempo. Por otro lado, los trabajos [101], [102], [103] resaltan a *OpenStack* como una plataforma favorable para el despliegue de ambientes de prueba *SDN* en la nube, pese a esto, destacan que la integración de *SDN*, *NFV* y *OpenStack* sin un controlador programable, pero presentan problemas de seguridad y confiabilidad durante la entrega de los servicios de los ambientes de prueba. En consecuencia, es evidente denotar que para un correcto despliegue de un ambiente de pruebas *SDN* es necesario contar con una infraestructura de red con capacidades de gestión de recursos (proceso, almacenamiento y redes) a petición de los entornos de experimentación, como lo es la plataforma Telco 2.0 de la FIET.

3.3 Ambientes de prueba para redes definidas por *software* virtualizadas

El trabajo “*OpenStackEmu - A cloud testbed combining network emulation with OpenStack and SDN*” [19] integra la plataforma *OpenStack*, el controlador *OpenDayLight* y el emulador de red a gran escala (*CORE - Common Open Research Emulator*). A fin de desplegar un ambiente de prueba *SDN* en la nube, programable y de alto rendimiento, el entorno permite a los experimentadores evaluar el rendimiento de algoritmos y protocolos novedosos en la nube e implementar y analizar distintos esquemas de enrutamiento y balanceo de carga sobre diferentes topologías de red emuladas.

“*Diseño y simulación de una red de DataCenters basada en una topología FatTree en un ambiente de redes definidas por software*” [104] establece un ambiente de prueba *SDN*. Este ambiente contribuye al estudio de los desafíos presentes en las redes de los centros de datos (*DC, DataCenter*), estos retos son basados en el aumento de la generación de tráfico dado por el creciente número de usuarios con acceso a Internet. El objetivo de este ambiente de prueba es diseñar e implementar una red de *DC* basada en topologías *Fat-Tree*, bajo la orden de un controlador *OpenDayLight* y siendo emulada por *Mininet*, permitiendo la reestructuración de la infraestructura con tal de mantener la calidad en los servicios proporcionados.

“*SDN sobre redes IEEE 802.11: Simulación mediante MiniNet Wifi*” [105] presenta un estudio del estándar IEEE 802.11, la tecnología *SDN*, el controlador *OpenDayLight* y el protocolo *OpenFlow* para el despliegue de un ambiente de prueba para *SDN* inalámbricas. La integración de la herramienta de emulación *MiniNet* y la extensión *MiniNet Wifi* con el controlador del ambiente de prueba, permite a los usuarios analizar el comportamiento del protocolo *OpenFlow* sobre tres topologías de red (*i.* topologías simples, *ii.* topologías con estaciones estáticas inalámbricas y *iii.* topologías con estaciones móviles).

“*Implementación de movilidad en redes 5G: Simulación de movilidad con redes definidas por software*” [106] tiene como objetivo estudiar el comportamiento de un ambiente de prueba *SDN* para una red móvil 5G (arquitectura y protocolos) en escenarios de movilidad. A través de la integración de *MiniNet* y el controlador *OpenDayLight*, el ambiente de prueba permite la emulación de un traspaso de servicios móvil (*Hand-Over*) [107] en una *SDN*. Este escenario de simulación móvil, concede a los usuarios analizar el comportamiento de los flujos de tráfico de la red de los conmutadores *SDN* involucrados en la conexión y medir la pérdida de paquetes efectuada en el proceso.

Los trabajos [19] [104] [105] [106] resaltan el despliegue de un ambiente de prueba *vSDN* soportados sobre *OpenStack*, controlados con *OpenDayLight* e integrados a emuladores de red de pequeña y gran escala. Estas colaboraciones están centradas en la experimentación, despliegue y prototipado de servicios y protocolos de red, los cuales, han sido desarrollados específicamente para ejecutar labores limitadas sobre diferentes topologías de red emuladas. La principal desventaja de los trabajos expuestos, es debida a la disposición de diversas pruebas preestablecidas no configurables, como por ejemplo, el usuario no dispone de herramientas que permitan la parametrización de los diferentes elementos de red, asimismo, el investigador carece de libertad en el despliegue de topologías de red, eligiendo solamente las topologías preestablecidas en tamaño y estandarización de protocolos. Por lo tanto, cabe resaltar que el grado de éxito de cualquier ambiente de pruebas, particularmente compartido por una gran comunidad, depende en gran medida de dos factores claves: la *utilidad* del entorno para una amplia gama de experimentos y su *disponibilidad* para los investigadores (universidades, centros de investigación y operadores de

telecomunicaciones).

3.4 Brechas

La literatura anteriormente descrita plantea el uso de tecnologías de virtualización para el despliegue del ambiente de pruebas *vSDN*, como también el uso de herramientas de generación y captura de métricas de desempeño del tráfico de red para determinar la *QoS*. La tabla 3.1 presenta las características y brechas relevantes encontradas en cada uno de los trabajos descritos respecto al trabajo de grado desarrollado.

Brechas		Trabajos relacionados											
		[17]	[18]	[19]	[20]	[94]	[97]	[101]	[102]	[103]	[104]	[105]	[106]
Entorno experimentación	Única				✓		✓	✓	✓		✓	✓	✓
	Múltiple	✓	✓	✓		✓				✓			
Tráfico soportado	Liviano			✓				✓	✓		✓	✓	✓
	Denso	✓	✓		✓	✓	✓			✓			
Recursos	Físicos		✓					✓	✓				
	Virtuales	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
Interfaz de usuario centralizada					✓	✓		✓	✓				✓
Controlador SDN		✓	✓			✓		✓		✓	✓	✓	✓

Table 3.1: Características de los trabajos relacionados

De la tabla 3.1, es notorio inferir que las principales limitaciones encontradas en los trabajos relacionados son debidas principalmente a la carencia de entornos de experimentación configurables a las necesidades de los investigadores y el control específico de diferentes niveles de granularidad de tráfico en los ambientes de prueba, como lo exponen los trabajos [20], [97], [101], [102], [104], [105], [106], en segundo lugar, la ausencia de integración con controladores de red que permitan el control estratégico de los entornos de experimentación *SDN*, como lo resaltan los trabajos [18], [17], [94], [101], [103], [19], [104], [105] y [106]. Para finalizar, es importante contar con una interfaz de acceso e interacción centralizada en el ambiente de pruebas, que facilite al máximo la transmisión de información, la navegabilidad y la interacción con los investigadores, como lo exponen los trabajos [20], [94], [101] y [102].

En resumen, dadas las brechas anteriormente detalladas, el presente trabajo de grado propone un ambiente de prueba para *vSDN* sobre la plataforma Telco 2.0, que presente un entorno con capacidades de experimentación múltiple, la gestión de diferentes configuraciones de tráfico (Liviano y denso); los recursos a consumir de este despliegue estarán virtualizados sobre la plataforma en mención, además, este ambiente contara con diferentes controladores *SDN*, todo esto dispuesto en una interfaz de usuario única y centralizada. Este trabajo idea el ambiente de pruebas como una herramienta de utilidad en una amplia gama de experimentos (soporte para herramientas de emulación y generación de tráfico de red en un mismo entorno) y disponibilidad para los investigadores (acceso libre a estudiantes y profesores que tengan relación con temáticas afines).

Capítulo 4

Diseño

En primer lugar, este capítulo define el escenario motivador, en segundo, esta sección expone el análisis de requerimientos necesarios para el diseño de la arquitectura del ambiente y por último, el diseño e implementación del ambiente modelado son definidos a partir de diagramas de casos de uso, secuencia y despliegue.

4.1 Escenario de Motivación

En esta sección es descrito un contexto motivador para la implementación de un ambiente de pruebas *vSDN* en Telco 2.0.

Es planteado un escenario en el que la industria de las Telecomunicaciones en Colombia y a nivel internacional, tiene como desafío la adopción de nuevas infraestructuras de red, capaces de soportar la evolución e innovación de servicios y aplicaciones de nueva generación. Las *SDN* son una de las soluciones tecnológicas que se adaptan a las necesidades de red requeridas para la investigación, desarrollo e innovación de nuevos servicios [108].

La carente implementación de las *SDN* en un despliegue de tecnologías de virtualización de funciones de red, no permite extender la arquitectura de estas redes a instancias de *VMs*, limitando al investigador en la gestión y experimentación en las áreas correspondientes. La falta de desarrollo de un ambiente con dichas características impide al investigador el acceso a herramientas de pruebas para la

experimentación e innovación en los panoramas referentes a las *SDN*.

Teniendo en cuenta las diferentes restricciones que un investigador de redes puede presentar en la experimentación de arquitecturas *SDN*, es notorio afirmar que, el despliegue de la arquitectura *SDN* en un ecosistema *NFV* optimiza y facilita el funcionamiento de los servicios de red implementados, no obstante, el rendimiento de un servicio depende del comportamiento de las métricas de calidad, como lo son: el retardo, el ancho de banda y la pérdida de paquetes de la red [109].

Cabe resaltar que ante las demandas de la industria y la academia actuales, los protocolos de red utilizados e implementados en un ambiente de pruebas no difieren en su ejecución de un ambiente real, por lo tanto, evaluar el comportamiento de las *vSDN* y sus estadísticas de desempeño permite monitorizar los parámetros y métricas de rendimiento, al igual que la calidad de servicio.

Luego, el presente trabajo propone el diseño, desarrollo e implementación de un ambiente de pruebas para redes definidas por *software* virtualizadas sobre Telco 2.0.

4.2 Diseño del Ambiente de pruebas para redes definidas por *software* virtualizadas

Con base a la necesidad de encontrar una metodología de desarrollo de *software* adaptativa al proceso de construcción del ambiente de prueba *vSDN* sobre Telco 2.0, es realizada una revisión de la literatura [110] [111] [112] [113] [114] sobre metodologías ágiles y es seleccionada la metodología *Scrum* debido a sus principales características que permiten el desarrollo iterativo e incremental, la simplicidad de la implementación, las entregas frecuentes y la priorización de los requerimientos o características a desarrollar a cargo del cliente [115] [116] [117].

Esta sección expone los procesos realizados para diseñar y construir un ambiente de pruebas *vSDN* innovador e interactivo. A partir del uso de la metodología ágil de desarrollo de software *Scrum* se definen las etapas para el análisis, desarrollo e implementación del proyecto, asimismo, la elaboración de la documentación respectiva tal como lo sugiere esta metodología. Los adelantos del proyecto son gestionados a través de las siguientes etapas: (i) Inicio, (ii) Planificación y estimación, (iii)

Implementación, (iv) Revisión y retrospectiva y (v) Lanzamiento.

En la etapa (i) es creada la visión del proyecto e identificados los roles claves del mismo, como también, es definida la lista de prioridades, base fundamental del plan de lanzamiento y tamaño de los *Sprints*. En la etapa (ii) los *Sprint* son aterrizados en historias de usuarios y se realizan las estimaciones de tiempo y esfuerzo según el valor generado al proyecto, estas estimaciones se traducen en la lista de tareas que deben cumplir los desarrolladores asignados durante la ejecución del *Sprint*. En la etapa (iii) es trabajada la construcción de los entregables requeridos por la lista *Sprint* a través del seguimiento de las actividades llevadas a cabo por los desarrolladores y los problemas enfrentados. Igualmente, es actualizada o revisada la lista de pendientes del producto. En la etapa (iv), son presentados al cliente los avances, dependencias e impedimentos en el desarrollo del proyecto y son articuladas las lecciones aprendidas que pueden aplicarse a futuros *Sprints*. Finalmente, la etapa más esperada por el cliente y el equipo *Scrum*, debido a que es desplegado el proceso de aceptación del entregable y finalización exitosa del *Sprint*, del mismo modo, son realizadas las actividades retrospectivas que permiten establecer mejoras e identificar las vulnerabilidades que a futuros proyectos pueden ser implementados.

4.2.1 Etapa i - Inicio

En esta etapa son retomadas las conclusiones del estudio literario realizado con el fin de proyectar las diferentes herramientas y funcionalidades que pueden brindar las *vSDN* y así ejecutar un despliegue digital de las mismas. A través de grupos de discusión con estudiantes, profesores e investigadores *ICT* de la FIET son identificados los diferentes tipos de roles que participan en el proceso de desarrollo y ejecución de un ambiente de pruebas *vSDN*. Posteriormente, es realizado un análisis de requisitos de usuario policéntrico con los roles identificados mediante técnicas como lluvia de ideas, cuestionarios y tarjetas de usuario para obtener una primera aproximación de las necesidades relacionadas con el ambiente. El análisis de requisitos es complementado mediante la investigación bibliográfica que permite profundizar en los requerimientos técnicos y funcionales del ambiente con relación a las contribuciones y tendencias tecnológicas actuales. Basándonos en la arquitectura y acceso a Telco 2.0 es concluyente que dicha implementación debe ser desplegada en un

entorno web.

Definición de roles para el ambiente de pruebas *vSDN*

En esta sección, son identificados los roles que interactúan con la plataforma *Web* del entorno a desarrollar a fin de obtener una mejor contextualización del problema y sus necesidades de uso consideradas importantes para la implementación del ambiente de pruebas. Los roles identificados participan en todas las fases de desarrollo y evaluación del ambiente de pruebas *vSDN* para garantizar un producto funcional y usable. La identificación de los roles es realizada mediante la adaptación de *Scrum* a este proyecto siendo usuarios activos que emplean la herramienta web del ambiente y son resumidos en la tabla 4.1.

Rol	Descripción	Motivación
Usuario	Estudiante de Ingeniería Electrónica de la Universidad del Cauca que hace uso de los sistemas de Telecomunicaciones presentes en la Universidad para el aprendizaje e investigación.	Realizar pruebas de concepto en <i>vSDN</i> desplegados en tecnologías de virtualización como <i>VM</i> o contenedores. Obtener una herramienta tecnológica que facilite la labor de enseñanza y aprendizaje en la FIET de la Universidad del Cauca en un ambiente de pruebas <i>vSDN</i> .
	Profesor de Ingeniería Electrónica y Telemática de la Universidad del Cauca que orienta cursos relacionados con investigación, desarrollo e innovación en tecnologías de <i>vSDN</i> .	Gestionar y optimizar los recursos de procesamiento y almacenamiento de Telco 2.0 de la FIET en un ambiente de pruebas virtualizado gestionado mediante una <i>GUI</i> en la <i>Web</i> . Garantizar la usabilidad y utilidad del ambiente.
	Personas de posgrados en Telemática y Telecomunicaciones de la Universidad del Cauca enfocados en investigación y desarrollo tecnológico de <i>vSDN</i> .	Acceso a recursos virtuales instanciados en arquitecturas de <i>VM</i> o contenedores utilizados para investigación y desarrollo en <i>vSDN</i> . Sistema de monitorización de recursos que permita evaluar el rendimiento y QoS de una <i>vSDN</i> . Heterogeneidad de tecnologías y protocolos disponibles en un ambiente de pruebas <i>vSDN</i> utilizados para profundización de conceptos, prueba de protocolos, interoperabilidad de sistemas e innovación tecnológica.

Table 4.1: Roles del ambiente de pruebas *vSDN*

Mediante los grupos de discusión llevados a cabo en la FIET ha sido identificado un actor principal que participa activa y decisivamente en el proceso de uso, desarrollo y evaluación del ambiente de pruebas *vSDN*. La función principal de este usuario es brindar información útil sobre las necesidades del ambiente desde diferentes puntos

de vista y evaluar su proceso de desarrollo con el fin de detectar errores tempranos en el diseño perfeccionando iterativamente el despliegue de las interfaces.

El análisis de requisitos es realizado con el actor mediante entrevistas iniciales y un cuestionario visualizado en la tabla 4.2, que permite obtener una primera aproximación de los requerimientos para el desarrollo del ambiente. El cuestionario inicial estudia la experiencia de usuario con respecto a la utilización de tecnologías de virtualización en Telco 2.0 que proporciona recursos instanciados en *VM*, este cuestionario conformado por preguntas cerradas evidencia las limitaciones e ideas abiertas sobre la necesidad de desarrollar un ambiente de pruebas *vSDN*. Este análisis permite obtener una amplia gama de información útil sobre las características de usabilidad y utilidad que debe ofrecer el entorno a desarrollar.

Item	Pregunta
P1	Has utilizado alguna vez máquinas virtuales (VM)
P2	Has utilizado alguna vez un ambiente de pruebas virtualizado
P3	Has utilizado herramientas para la virtualización de redes definidas por <i>software</i> . Si tu respuesta es sí, responde.
P3.1	¿Qué funcionalidades destacas de las herramientas utilizadas?.
P3.2	¿Qué funcionalidades presentan carencia en las herramientas utilizadas?.
P3.3	¿Cuál ha sido tu nivel de satisfacción con el uso de estas herramientas?
P4	Si tu respuesta es no, responde.
P4.1	¿Cómo estudias las funcionalidades de las redes definidas por <i>software</i> virtualizadas?
P4.2	¿Cómo desearías acceder a las herramientas de experimentación de redes definidas por <i>software</i> virtualizadas?
P5	¿Te gustaría tener una herramienta de estudio y experimentación para redes definidas por <i>software</i> en Telco 2.0?
P6	¿Cómo y mediante que tecnología te gustaría que fuera la implementación de la herramienta de pruebas para redes definidas por <i>software</i> virtualizadas?

Table 4.2: Cuestionario preliminar análisis de requisitos de usuario

El cuestionario anterior de análisis de requisitos fue realizado a 12 actores por medio de entrevistas en los grupos de discusión, los efectos causados por la encuesta

expuesta en la Tabla 4.2 son sintetizados en la Tabla 4.3, la cual muestra el despliegue de las diferentes respuestas contextualizadas mediante el enfoque deseado, destacando tres tópicos fundamentales: Virtualización, herramientas *vSDN* y nivel educativo.

VIRTUALIZACIÓN								
Encuestados	Máquinas virtuales				Herramientas virtualizadas			
	Alto	Medio	Bajo	Ninguno	Alto	Medio	Bajo	Ninguno
12	2	4	5	1	3	6	2	1
HERRAMIENTAS <i>vSDN</i>								
Encuestados	Herramientas				Satisfacción			
	Interfaz gráfica	Tráfico	Licencia Libre	Configurable	Alta	Mediana	Baja	
7	2	1	2	2	0	3	4	
EDUCACIÓN								
Encuestados	Información			Acceso				
	Academia	Referencias bibliográficas		Academia	Referencias bibliográficas	Aplicación		
5	5	5		3	1	5		
OBSERVACIONES Y SUGERENCIAS								
1	La herramienta debe permitir la implementación de diferentes controladores de red.							
2	Crear una plataforma de Web intuitiva, estable, coherente y fácil de utilizar.							
3	El nivel de accesibilidad y manejo de la herramienta debe ser interactivo.							
4	Permita guardar los estados y resultados de la emulación.							
5	La aplicación debe ser configurable, permita el diseño libre de topologías y la gestión de tráfico de la red.							

Table 4.3: Análisis preliminar de requisitos y experiencia de usuarios

Análisis de requisitos

El análisis realizado permite obtener una primera aproximación de los requisitos de usuario, teniendo en cuenta que en este nivel, los posibles usuarios del sistema no tienen muy claro los objetivos y alcances en el uso de un ambiente de pruebas *vSDN*, por lo tanto, esta etapa es orientada entorno a las necesidades de usuario. Posteriormente, son expuestos los servicios requeridos por el experimentador con el fin de afianzar los conocimientos e investigaciones en tecnologías de redes de nueva generación como lo son las arquitecturas *vSDN*.

En particular, las personas encuestadas en esta etapa solicitan un banco de recursos independientes para desplegar arquitecturas *vSDN* configurables sobre instancias de *VM* que faciliten los procesos de aprendizaje, experimentación e investigación tecnológica en un ambiente de pruebas; además, pueden configurar el tráfico y ejecutarlo en una topología de red. Los servicios del ambiente deben ser gestionados por una *GUI* en un entorno *web* intuitivo, amigable y fácil de usar.

El análisis de requisitos es complementado a partir de las brechas y contribuciones indagadas en la literatura que permiten, profundizar en los requerimientos técnicos para el desarrollo de un ambiente de pruebas *vSDN*. Este enfoque está dividido en dos ejes principales, denominados requisitos técnicos y requisitos de diseño para el desarrollo de un ambiente de pruebas funcional y usable desde el punto de vista de los usuarios. La Tabla 4.4 resume los requisitos técnicos indagados en la literatura con respecto a un entorno para la experimentación e investigación sobre un sistema telemático.

Referencia	Requerimientos
[17] [18] [94]	Emulaciones configurables, tamaño y configuración de sus experimentaciones.
[17] [18] [20]	Interfaz única de gestión de servicios <i>vSDN</i>
[94] [103]	Despliegue de la arquitectura <i>SDN</i> sobre instancias de <i>VM</i>
[102] [104] [106]	Gestión de virtualización para consumo de recursos.
[97] [20]	Monitorización de resultados del procesamiento de tráfico y demás parámetros para la evaluación de las <i>vSDN</i>
[19]	Monitorizar la <i>QoS</i> sobre las topologías emuladas.
[101]	Visualización de gráficas de rendimiento y <i>QoS</i> monitorizadas
[105] [97]	Ejecución de diferentes configuraciones de rendimiento y <i>QoS</i> sobre <i>vSDN</i>
[19]	Gestión de recursos virtualizados para desplegar la arquitectura <i>SDN</i> .
[101]	Interfaz <i>web</i> intuitiva y fácil de utilizar

Table 4.4: Análisis de la literatura, requisitos técnicos de un ambiente de pruebas *vSDN*

La literatura referenciada denota la importancia del uso de tecnologías de virtualización para optimizar, gestionar y asignar recursos con el fin de brindar diferentes servicios a los usuarios.

En un ambiente de pruebas virtualizado los autores sugieren que este entorno permita emulaciones configurables y recomiendan una lista de parámetros a tener en cuenta, como lo son: tiempo de emulación, despliegue de diferentes topologías, gestión del tráfico de la red, entre otros. Además, los servicios consumidos deben reposar en una única interfaz de gestión y dichos procesos deben realizarse mediante la virtualización.

Cabe denotar la ausencia de emuladores *vSDN* que permitan diferentes disposiciones a los usuarios, estos experimentadores no cuentan con la posibilidad de ejecutar variaciones en los parámetros que dan acceso a la evaluación de la red.

Considerando estas necesidades es destacable la oportunidad de desplegar un ambiente de pruebas *vSDN* que permita la ejecución, monitorización y valoración de este tipo de redes, estimar la *QoS* de las mismas ante diferentes ejecuciones y configuraciones de rendimiento. En conclusión, es dispuesto ante tales requerimientos una interfaz *WEB* intuitiva y usable para satisfacer de buen modo las necesidades planteadas.

La información recopilada da lugar a la implementación del diseño ágil centrado en el usuario para el desarrollo de un ambiente de pruebas *vSDN* funcional y usable que llegue a satisfacer las necesidades expuestas.

4.2.2 Etapa *ii* - Planificación y estimación

Esta etapa tiene como objetivo definir los requisitos funcionales y contenidos de información proporcionados en la *GUI* del ambiente de pruebas *vSDN* en Telco 2.0, a partir de la relación y priorización de requisitos investigados policéntricamente con los usuarios y el análisis de la literatura. Asimismo, definen los roles de usuario en el ambiente de pruebas a partir de las consideraciones y requisitos realizados por los experimentados y la literatura. Los requisitos priorizados en esta sección son utilizados para modelar las acciones que debe realizar un usuario en el entorno mediante herramientas de diseño como historias de usuario que componen la cartera de producto de funcionalidades del ambiente de pruebas *vSDN*.

Relación de requisitos por análisis de discusión

En esta parte, los requisitos investigados con los usuarios son relacionados y priorizados con el fin de definir los requerimientos funcionales y contenidos de información de la plataforma *Web* del ambiente de pruebas en Telco 2.0. La siguiente lista sintetiza la relación y la jerarquía de requisitos realizada con los usuarios mediante grupos de discusión.

Requisitos funcionales

- Creación y configuración de elementos de red.
- Creación y configuración de topologías de red en arquitecturas de diseño predefinidas.
- Creación y configuración de topologías de red en arquitecturas de diseño libre.

- Emulación de la topología creada.
- Parametrización de los criterios de tráfico a implementar.
- Monitorizar el rendimiento y *QoS* de las redes emuladas desde el servidor.
- Monitorizar el rendimiento y *QoS* de las redes emuladas desde el cliente.
- Monitorizar el rendimiento y *QoS* de las redes emuladas desde una vista general de la red.
- Visualizar gráficos de monitorización.
- Guardar estados de la topología diseñada.
- Guardar resultados del tráfico generado en la red emulada con respecto al tráfico generado desde el servidor.
- Guardar resultados del tráfico generado en la red emulada con respecto al tráfico generado desde el cliente.
- Guardar resultados del tráfico generado en la red emulada con respecto al tráfico generado desde una vista general.

Contenidos de información

- Plataforma *Web* del ambiente montada sobre un dominio *URL* de la Universidad del Cauca.
- Ver información sobre los servicios del ambiente de pruebas *vSDN*.
- Interfaz de gestión en la *Web* del ambiente estable, usable y escalable.
- Contenidos de la plataforma *Web* del ambiente coherentes y precisos.
- Documentación de las funcionalidades del ambiente de pruebas *vSDN*.
- Guías de usuario.

La relación de requisitos permite observar la equivalencia de los requerimientos indagados con los usuarios para así modelar sus funcionalidades mediante historias de usuario. Las configuraciones destacadas son proyectadas a la gestión y configuración de un entorno *vSDN*.

En particular, el ambiente de pruebas *vSDN* debe proporcionar a los usuarios un despliegue de topologías de red configurable, como también herramientas de monitorización y control para la evaluación de la red mediante la parametrización y gestión del tráfico deseado. Asimismo, este entorno debe permitir guardar los estados y resultados de la emulación realizada, a fin, de permitirle al usuario evaluar el rendimiento de una arquitectura *vSDN*. Todas estas utilidades estarán gestionadas a través de *GUI* en la *Web* con un despliegue intuitivo, amigable y fácil de usar.

Relación de requisitos por análisis literario

Los requisitos son priorizados con el fin de definir el alcance del proyecto y establecer las funciones de utilidad y usabilidad que debe proporcionar el ambiente de pruebas *vSDN* sobre Telco 2.0. Los resultados obtenidos son visualizados en la Figura 4.5.

Los requerimientos destacados en el análisis, enfatizan la importancia del uso de un ambiente de pruebas desplegado en un escenario *Web*, que permita la creación y la gestión de *vSDN* y la parametrización de diferentes valores de tráfico. Los recursos asignados a cada emulación son limitados, ya que dependen de las configuraciones dadas a cada máquina virtual donde están virtualizados. Asimismo, esta interfaz debe permitir ejecutar pruebas, parametrizar valores de tráfico, monitorizar el rendimiento y la *QoS* de la topología emulada.

La caracterización obtenida por el análisis dados por la experiencia de usuario y literario son priorizados con el fin de definir el alcance del proyecto y establecer las funciones de utilidad y usabilidad que debe proporcionar el ambiente de pruebas *vSDN*. Los resultados obtenidos son visualizados en este orden en la Figura 4.5.

Requisitos análisis literario	Requisitos análisis de discusión
<p>Emulación de la topología creada.</p> <p>Despliegue de la arquitectura <i>SDN</i> sobre instancias de <i>VM</i></p>	<p>Despliegue de la arquitectura <i>SDN</i> sobre instancias de <i>VM</i>.</p> <p>Gestión de virtualización para consumo de recursos.</p> <p>Gestión de recursos virtualizados para desplegar la arquitectura <i>SDN</i>.</p> <p>Ejecución de diferentes configuraciones de rendimiento y <i>QoS</i> sobre <i>vSDN</i>.</p>
<p>Ver información sobre los servicios del ambiente de pruebas <i>vSDN</i>.</p> <p>Visualizar gráficos de monitorización.</p> <p>Interfaz de gestión en la <i>Web</i> del ambiente estable, usable y escalable.</p> <p>Contenidos de la plataforma <i>Web</i> del ambiente coherentes y precisos.</p> <p>Plataforma <i>Web</i> del ambiente montada sobre un dominio <i>URL</i> de la Universidad del Cauca.</p> <p>Documentación de las funcionalidades del ambiente de pruebas <i>vSDN</i>.</p> <p>Guías de usuario.</p>	<p>Interfaz única de gestión de servicios <i>vSDN</i>.</p> <p>Interfaz <i>web</i> intuitiva y fácil de utilizar.</p> <p>Visualización de gráficas de rendimiento y <i>QoS</i> monitorizadas</p>
<p>Creación y configuración de elementos de red.</p> <p>Creación y configuración de topologías de red en arquitecturas de diseño predeterminadas.</p> <p>Creación y configuración de topologías de red en arquitecturas de diseño libre.</p> <p>Guardar estados de la topología diseñada.</p> <p>Guardar resultados del tráfico generado en la red emulada con respecto al tráfico generado desde el servidor.</p> <p>Guardar resultados del tráfico generado en la red emulada con respecto al tráfico generado desde el cliente.</p>	<p>Emulaciones configurables, tamaño y configuración de sus experimentaciones.</p>
<p>Monitorizar el rendimiento y <i>QoS</i> de las redes emuladas desde el servidor.</p> <p>Monitorizar el rendimiento y <i>QoS</i> de las redes emuladas desde el cliente.</p> <p>Monitorizar el rendimiento y <i>QoS</i> de las redes emuladas desde una vista general de la red.</p>	<p>Monitorización de resultados del procesamiento de tráfico y demás parámetros para la evaluación de las <i>vSDN</i></p> <p>Monitorizar la <i>QoS</i> sobre las topologías emuladas.</p>

Table 4.5: Relación de requisitos

Historias de usuario

El ambiente de pruebas *vSDN* tiene un único usuario para la utilización de las herramientas que ofrece esta aplicación, este usuario abarca tres diferentes perfiles (Profesores, estudiantes e investigadores) pero con el mismo nivel de acceso al ambiente. A este usuario lo denominaremos experimentador.

Cada historia de usuario diseñada establece un nivel de prioridad para el desarrollo del proyecto, así como también, definen la estimación de progreso en semanas y los criterios de aceptación. Las historias de usuario definidas para el desarrollo del ambiente de pruebas *vSDN* en Telco 2.0 son definidas y detalladas en el anexo A.

Lista *Sprint*

En la figura 4.1 se listan las historias de usuario, teniendo en cuenta el orden de prioridad, el tiempo de ejecución, los actores encargados relacionados con el desarrollo del ambiente de pruebas *vSDN* sobre Telco 2.0. Los criterios tenidos en cuenta para priorizar la ejecución de estas historias han sido evaluados en función al valor aportado al proyecto y la complejidad de desarrollo, asimismo, cabe resaltar que algunas historias de usuario son prerequisites de otras, por lo tanto, deben ser desarrolladas antes de las dependientes. Finalmente, el progreso del producto es evaluado iterativamente en *Sprints* periódicos realizados con los diversos actores del ambiente de pruebas mediante grupos de discusión.

Sprint	Historia de Usuario		Estimación semanas	Prioridad
1	HU01	Acceso al ambiente de pruebas vSDN.	5	Alta
2	HU02	Herramientas para el diseño libre.	5	Alta
3	HU03	Herramientas para el diseño predeterminado.	5	Alta
4	HU13	Parametrización de los elementos de red.	3	Media
	HU11	Herramientas de acción.	2	Baja
5	HU04	Ejecución y emulación de red.	4	Alta
6	HU05	Detención emulación de red.	2	Alta
	HU06	Configuración de parámetros de tráfico.	2	Alta
7	HU07	Reultados de tráfico.	3	Alta
	HU10	Guardar estado de la emulación.	2	Media
8	HU08	Configurar controladores de red.	4	Alta
9	HU09	Métricas de desempeño individuales.	4	Alta
10	HU12	Visualización de documentación y ayuda.	2	Baja

Figura 4.1: Lista *Sprint*

Las actividades a desarrollar en cada *Sprint* basado en las historias de usuario correspondientes, se despliegan en la siguiente lista.

- HU01
 1. Configuración de un dominio de red gestionado por Telco 2.0.
 2. Selección e instalación de las herramientas de desarrollo escogidas para un despliegue de una aplicación Web.
 3. Desarrollo de una interfaz gráfica que permita el acceso al ambiente de pruebas vSDN y sus herramientas.
- HU02
 1. Creación de un espacio de trabajo en el que puedan ser desplegados los diferentes elementos de red.
 2. Creación de un panel de selección de elementos de red.

3. Creación de reglas y lineamientos para la correcta interacción con los elementos de red.

- HU03

1. Creación de menú de selección de topologías de red.
2. Desarrollo de interfaz de configuración para el despliegue de diferentes topologías de red.
3. Creación de reglas para la implementación y despliegue de la topología minimal.
4. Creación de reglas para la implementación y despliegue de la topología simple.
5. Creación de reglas para la implementación y despliegue de la topología lineal.
6. Creación de reglas para la implementación y despliegue de la topología en anillo.

- HU04

1. Configuración del motor de infraestructura de red en el servidor.
2. Creación y comunicación de la interfaz de usuario del ambiente de pruebas vSDN con el servidor de infraestructura virtualizada.
3. Envío de la configuración de la red diseñada al servidor.
4. Ejecución de la información de la red a emular.

- HU05

1. Envío de la orden de detención al servidor.
2. Ejecución de detención de los servicios de emulación brindados por el motor de infraestructura virtualizada.

- HU06
 1. Creación del menú de configuración de los diferentes parámetros del tráfico que se desea ejecutar.
 2. Envío de parámetros de tráfico al servidor.
 3. Ejecución de los parámetros de tráfico en la herramienta de generación de tráfico alojada en el servidor.

- HU07
 1. Envío de la información arrojada por la herramienta generadora de tráfico a la interfaz de usuario del ambiente de pruebas vSDN.
 2. Interpretación de los datos recibidos desde el servidor por la herramienta de generación de tráfico.
 3. Creación de una interfaz que permita visualizar gráficamente las métricas de desempeño del tráfico.
 4. Despliegue de resultados de métricas de desempeño del tráfico vistas desde el servidor.
 5. Despliegue de resultados de métricas de desempeño del tráfico vistas desde el cliente.
 6. Despliegue de resultados de métricas de desempeño del tráfico vistas desde la red en general.
 7. Despliegue de la interfaz de un monitor de tráfico predefinido.

- HU08
 1. Configuración del controlador *OpenDaylight* en Telco 2.0.
 2. Configuración del controlador *ONOS* en Telco 2.0.
 3. Creación de menú de selección del controlador SDN.
 4. Habilitación de acceso de cada controlador VSDN.

- HU09
 1. Creación de una interfaz de despliegue de datos con valoración de métricas de desempeño de tráfico de cada elemento tipo host de la red desde el cliente.
 2. Creación de una interfaz de despliegue de datos con valoración de métricas de desempeño de tráfico de cada elemento tipo host de la red desde el servidor.
- HU10
 1. Creación de un archivo que contenga la configuración de la red diseñada y que pueda ser descargado por el usuario.
 2. Creación de un archivo que contenga la configuración de las métricas de desempeño del tráfico generado en la red emulada.
 3. Creación de un archivo que contenga los resultados gráficos de las métricas de desempeño de la red vSDN.
- HU11
 1. Creación de la herramienta rehacer y deshacer.
 2. Creación de la herramienta copiar y pegar.
 3. Creación de la herramienta borrar.
- HU12
 1. Creación de herramientas multimedia para la documentación del uso de la aplicación.
 2. Creación de la información en temas de ayuda en las diferentes funcionalidades y herramientas del ambiente de pruebas.
- HU13
 1. Creación de menú de configuración para el elemento de red tipo host.
 2. Creación de menú de configuración para el elemento de red tipo conmutador.

3. Creación de menú de configuración para el elemento de red tipo controlador.
4. Creación de menú de configuración para el elemento de red tipo puerto.
5. Creación de menú de configuración para el elemento de red tipo enlace.

4.2.3 Etapa *iii* - Implementación

En esta etapa es definido el diseño de interacción, además es descrita la infraestructura del ambiente de pruebas *vSDN* sobre Telco 2.0 a través del desarrollo de las actividades establecidas en los *Sprints*.

Diseño de interacción

El diseño interactivo es realizado con base en la visualización de las funciones, elementos y métodos que contendrá el ambiente de pruebas *vSDN*, el comportamiento lógico y funcional también son destacables en este proceso. Este apartado es modelado mediante el diagrama de casos de uso y el diagrama de clases, logrados a través de la interpretación del requerimiento.

Diagramas de casos de uso del ambiente de pruebas

El diagrama de un caso de uso observado en la Figura 4.2 modela las acciones del usuario experimentador en el ambiente de pruebas *vSDN* sobre Telco 2.0. Este despliegue se obtiene a través de las actividades establecidas en los *Sprint*. El diagrama exhibe la interacción lógica entre el experimentador con los componentes del ambiente, brindando una representación coherente del funcionamiento del mismo.

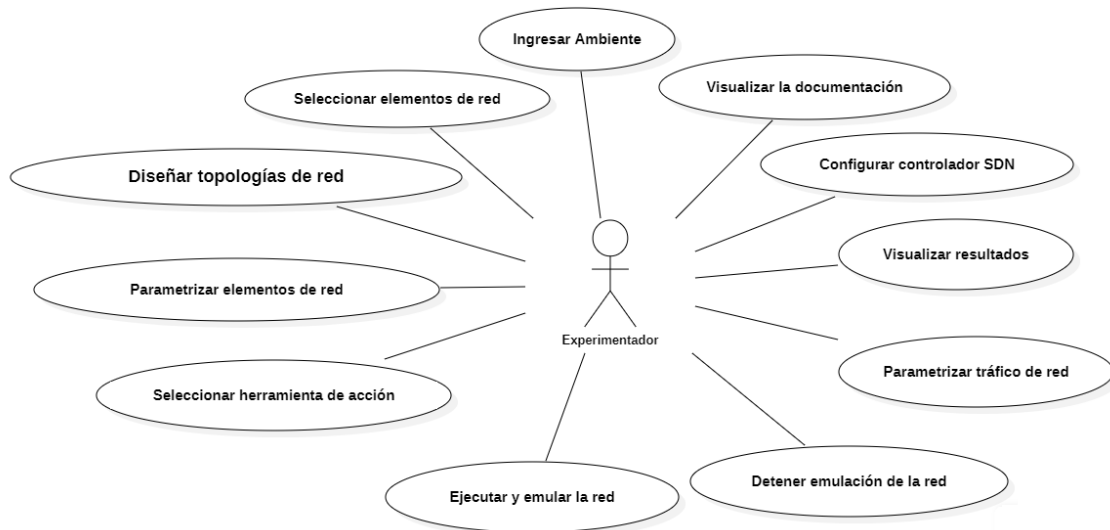


Figura 4.2: Caso de uso del ambiente de pruebas

La figura 4.2, nos induce a que el experimentador puede ingresar al entorno accediendo al dominio reservado para la ejecución del ambiente de pruebas *vSDN* sobre Telco 2.0 sin necesidad de contar con credenciales de acceso o algún permiso especial. En este emulador virtual, el usuario puede disponer de diferentes paneles de herramientas, entre ellos, el panel de selección de elementos de red, en el cual, se puede seleccionar entre cualquier componente y disponer de este en un lienzo o área de trabajo predeterminada.

Todos los elementos de red desplegados son configurables en su totalidad, el entorno permite optar por dos modos de diseño, a través de plantillas que crean topologías de red predeterminadas, pero configurables, o encaminarse por el diseño libre de una topología *vSDN*. El ambiente de pruebas *vSDN* pretende brindar herramientas similares a cualquier aplicación de creación o edición, por lo tanto, es posible ejecutar distintas órdenes como lo pueden ser: seleccionar, borrar, copiar, rehacer, deshacer y pegar.

Como es de esperar, el ambiente permite emular la red diseñada, además, concede la parametrización y gestión de la configuración del tráfico a generar en dicha red. La monitorización de los resultados de la emulación puede realizarse de modo individual en cada uno de los elementos de red (Tipo Host) o con la monitorización conjunta

de la red con tres perspectivas posibles (Cliente, servidor, red general) y disponer de un analizador de protocolos de red como complemento de la valorización de las métricas de desempeño expuestas por los resultados de las experimentaciones.

Una de las funciones adicionales que presenta esta herramienta de emulación es permitir el acceso a la selección de distintos controladores de red, que podrán ser cargados a la parametrización de la red diseñada. Finalmente, la documentación y temas de ayuda están siempre disponibles al usuario en el que momento que requiera de ellas.

Diagrama de clases del ambiente de pruebas

El diagrama de clases visualizado en la figura 4.3, proporciona una estructura básica de los componentes identificados en el diagrama de casos de uso modelados a partir de objetos. Un objeto es una instancia de una clase que define el comportamiento y funcionalidad del ambiente de pruebas *vSDN* mediante métodos y atributos. Los atributos definen las propiedades de un objeto que permiten identificar sus características, mientras que, los métodos especifican las funciones que realiza cada objeto por medio de sus atributos.

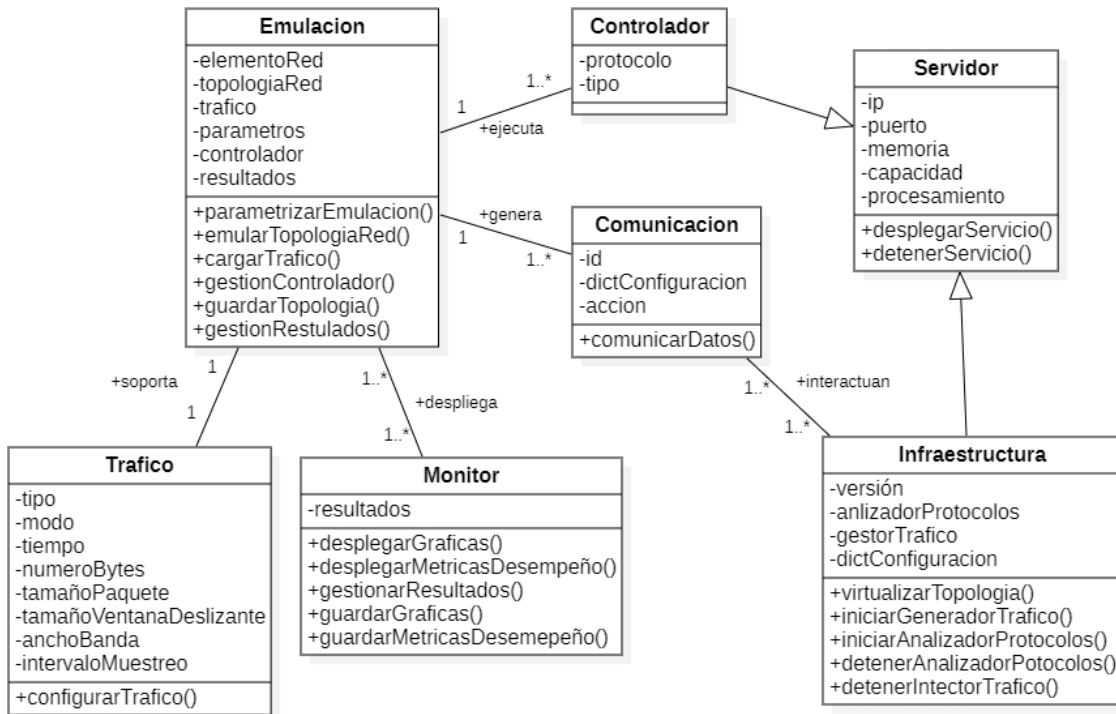


Figura 4.3: Diagrama de clases Telco IMS

La estructura del ambiente de pruebas *vSDN* sobre Telco 2.0 es definida por medio de clases interconectadas entre sí, las clases representan la abstracción de los objetos que prestan los servicios al usuario. El diagrama recoge la información de los requisitos de usuario y los convierte en funciones que cumplen con el objetivo del ambiente. Las clases visualizadas en la Figura 4.3 son explicadas a continuación:

- **Emulación:** Representa el gestor de emulación en el lado del cliente, que orienta y rige los protocolos de las topologías diseñadas y la gestión de los parámetros de tráfico que son ejecutados en una *vSDN*, como también administra el acceso a la configuración de los diferentes controladores externos.
- **Tráfico:** Gestiona los valores de configuración soportados por el motor del generador que se ejecuta en el servidor.
- **Servidor:** Representa un servicio virtualizado que instancia y proporciona herramientas de control y configuración según corresponda. Existen dos tipos de servidores, la infraestructura y de controladores.

- **Comunicación:** Gestiona la información y las acciones que envía y recibe el emulador desde el servidor y viceversa.
- **Infraestructura:** Es un servidor de despliegue que virtualiza las arquitecturas de red, la herramienta de análisis y el generador de tráfico requerido por el ambiente.
- **Controlador:** Es un servidor que gestiona la entidad lógica que controla la red virtualizada.
- **Monitor:** Esta clase permite visualizar las métricas de desempeño de las emulaciones puestas a prueba en el ambiente, asimismo, permite graficar los parámetros de *QoS*.

Arquitectura

La arquitectura en que se fundamenta el ambiente de pruebas *vSDN* está definida de acuerdo a las especificaciones literarias y experiencias de usuario. El experimentador ingresa desde un equipo configurado para el uso del ambiente de pruebas *vSDN* dispuesto en el dominio de Telco 2.0 a través de una interfaz gráfica de usuario.

La *GUI* es la herramienta de interacción con el usuario que le permite diseñar topologías y configurar las diferentes parametrizaciones de la red, el tráfico, y concede la visualización de los resultados que valoran las métricas de desempeño. La interfaz orienta la comunicación con el servidor mediante el intercambio de información, como puede ser, la disposición y parametrización de la topología de red a emular.

El servidor contiene y es capaz de ejecutar la herramienta de infraestructura de red, la cual es la encargada de gestionar la arquitectura de red y emular la *vSDN* que es descrita en la parametrización enviada por la interfaz. Administra además, un generador de tráfico que permite emular la configuración deseada por el usuario. Finalmente, dispone de un analizador de protocolos de red que puede ser invocado desde la *GUI* del lado del experimentador.

El controlador externo es ejecutado desde la *GUI* e invoca la interfaz de configuración por defecto del controlador seleccionado, este servidor pertenece al dominio de Telco 2.0 y es concedido el acceso mediante una *IP* específica que le indica tanto al cliente como al servidor de infraestructura como debe dirigir la interacción de datos, cabe

resaltar, que el controlador gestiona a través de su interfaz propia la configuración de los parámetros de la red a emular.

Lo anterior es descrito en el siguiente diagrama de componentes (Ver figura 4.4) el cual es utilizado para modelar la arquitectura en el despliegue del ambiente de pruebas *vSDN*.

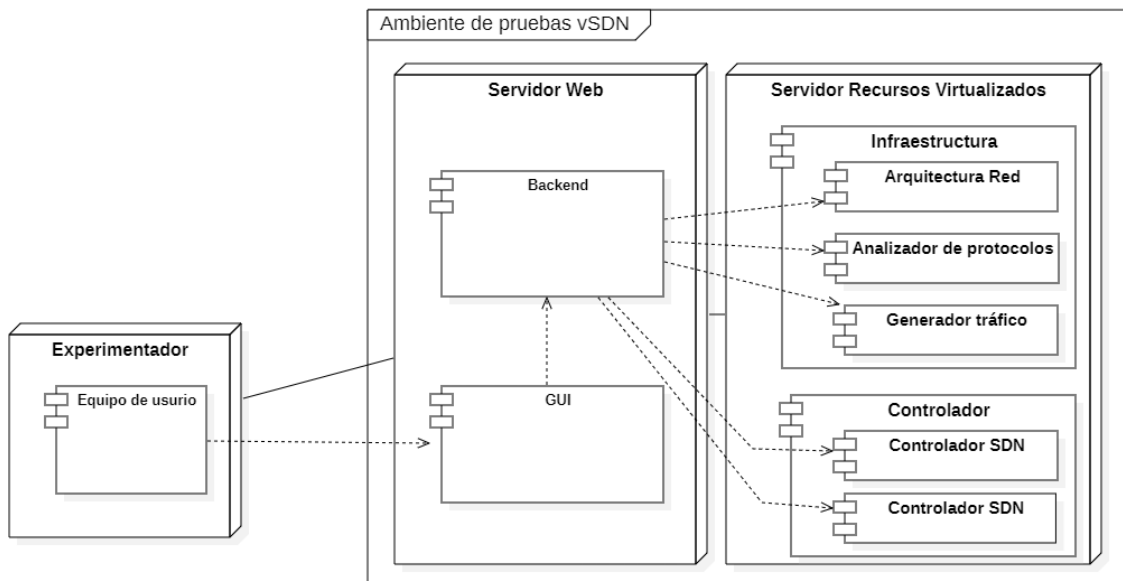


Figura 4.4: Diagrama de componentes ambiente pruebas *vSDN*

El usuario al acceder al dominio de Telco 2.0 designado para el ambiente puede diseñar, parametrizar y construir diferentes topologías de red en la interfaz gráfica de usuario, el ambiente, permite enviar la información de la red al servidor para que este a su vez, ejecute la herramienta de infraestructura de red virtualizada, la cual despliega la parametrización y crea una red *SDN* con base a dichos parámetros. El experimentador puede gestionar en la *GUI* un menú que da valor a las diferentes condiciones en que el tráfico debe ser ejecutado, estos valores son remitidos al servidor y en consecuencia el generador de tráfico realiza las peticiones requeridas, dando como resultado la valorización de las métricas de desempeño que posteriormente puedan ser desplegadas en la interfaz principal del usuario.

Mientras la red este siendo diseñada, el usuario puede disponer y configurar los diferentes controladores externos en su propia interfaz de parametrización, este

controlador seleccionado es dispuesto por el servidor y cargado a la *vSDN* construida para reglamentar el comportamiento del tráfico a generar.

La emulación puede ser detenida en cualquier momento y es permitido para el experimentador descargar tanto la configuración de la red como las métricas de desempeño deseado. El funcionamiento que anteriormente se describe es modelado por el diagrama de secuencias 4.5.

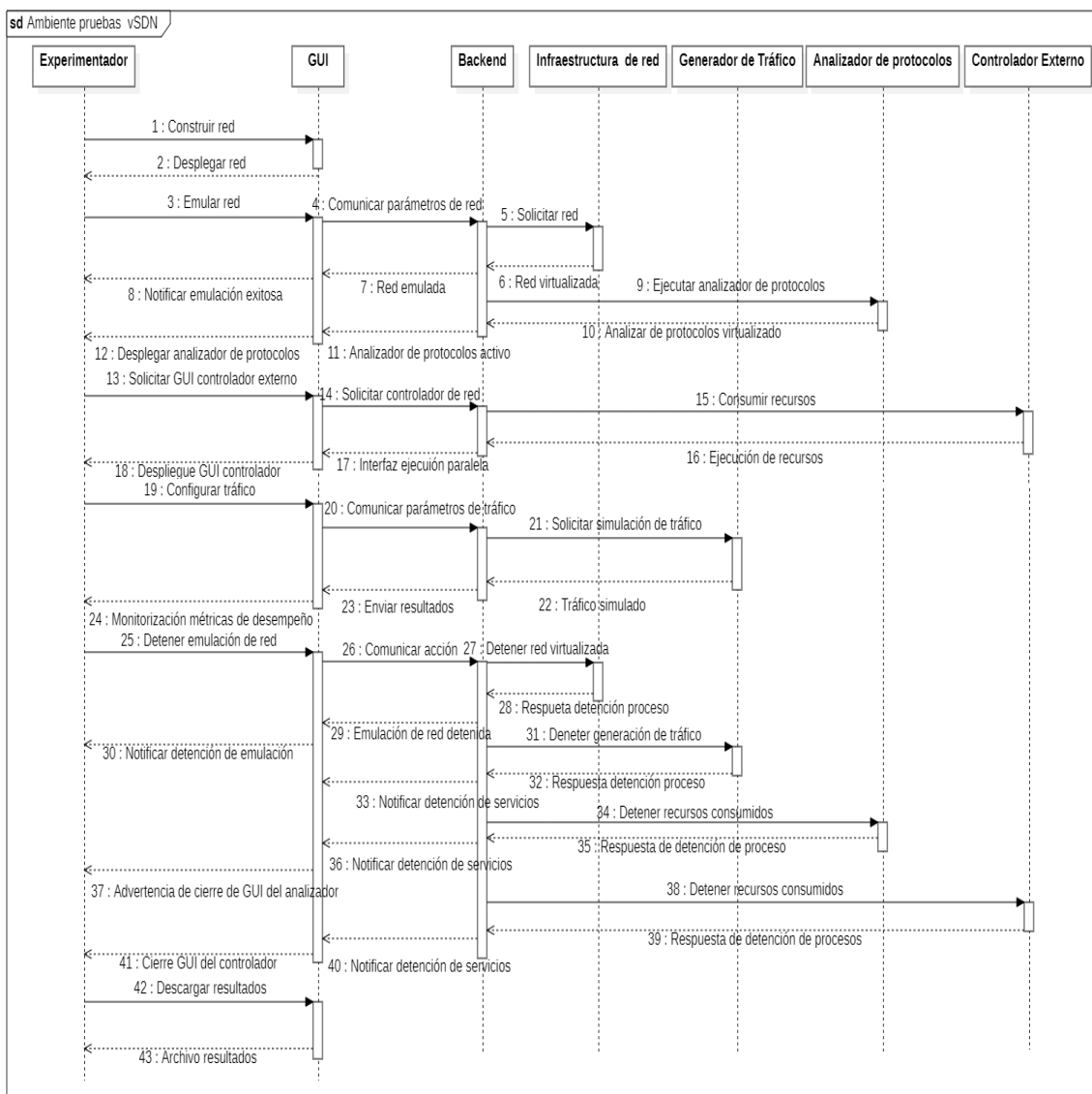


Figura 4.5: Diagrama de secuencias del ambiente pruebas *vSDN*

4.2.4 Etapa *iv* - Revisión y retrospectiva

De acuerdo a la abstracción lograda a través de planeación e interacción con las experiencias de usuario es generada una secuencia de interfaces, que son desplegadas conforme como la literatura y el experimentador sugieren. Mediante los siguientes bosquejos es descrita la propuesta de las principales interfaces gráficas del ambiente de pruebas *vSDN*, las cuales, destacan las cuatro orientaciones básicas que pretende el entorno: acceso, diseño, gestión y resultados, las demás vistas son expuestas en los anexos adjuntos a este trabajo.

Bosquejo interfaz de acceso

La figura 4.6 expone la primera vista del ambiente de pruebas *vSDN*, el cual admite a cualquier usuario que tenga acceso a algún dominio relacionado con Telco 2.0. Esta interfaz despliega en el centro de pantalla el nombre de la aplicación a ingresar y un botón que le permite la entrada, además, en esta vista son mostrados los colores característicos del ambiente cuando este sea implementado.

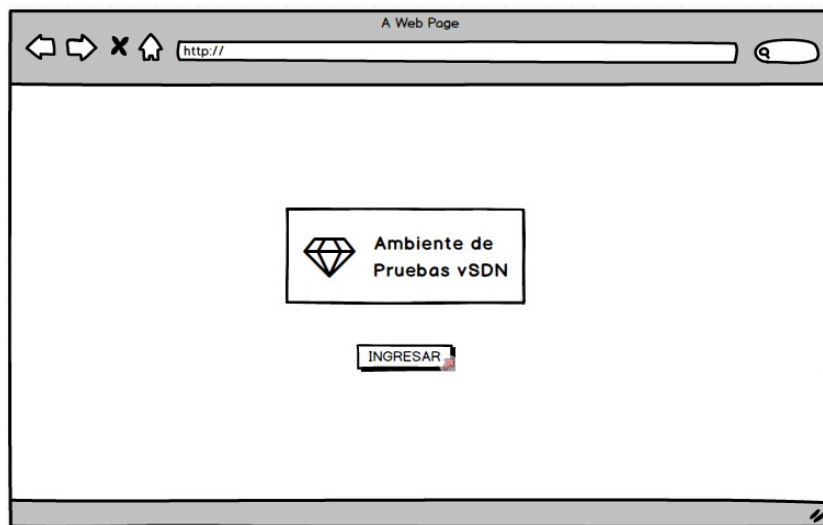


Figura 4.6: Interfaz de inicio

Bosquejo: Interfaz de inicio - diseño libre

El siguiente bosquejo, figura 4.7, ilustra la vista principal del ambiente de pruebas *vSDN* compuesto por tres paneles principales, el primero, un panel de menús:

archivo, edición, emulación y ayuda, los cuales contienen opciones conforme al contexto que orienta su encabezado; el segundo menú, expone el despliegue de los accesos directos a las principales funciones que tiene el ambiente; un panel lateral izquierdo es la tercera disposición de menú encontrada en esta vista, y contiene las herramientas que permiten seleccionar los diferentes elementos de red que componen una *vSDN* y ayudan al usuario en el diseño y parametrización de diferentes topologías que pueden ser dispuestas en el área de trabajo de modo libre. La disposición con mayor cobertura de área es el lienzo o campo de trabajo donde podrán ser desplegados los elementos de red.

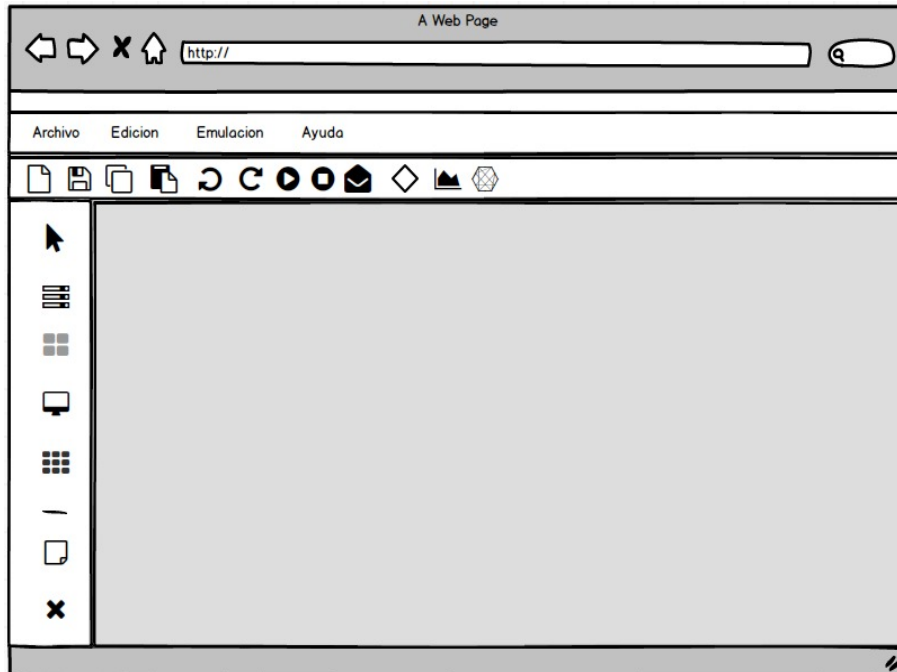


Figura 4.7: Interfaz de inicio y diseño libre

Bosquejo: Interfaz de topologías prediseñadas

La figura 4.8, muestra como el ambiente ofrece una serie de topologías de red predefinidas, que ayudan al usuario en el diseño y la experimentación de una red *SDN*. Las configuraciones de diseño ofrecidas por el ambiente de pruebas son cuatro: topología minimal, topología simple, topología en anillo y topología lineal, al seleccionar cualquiera de estas se solicitará al usuario las características para el despliegue

del diseño escogido. Y esta será desplegado en el lienzo de trabajo.

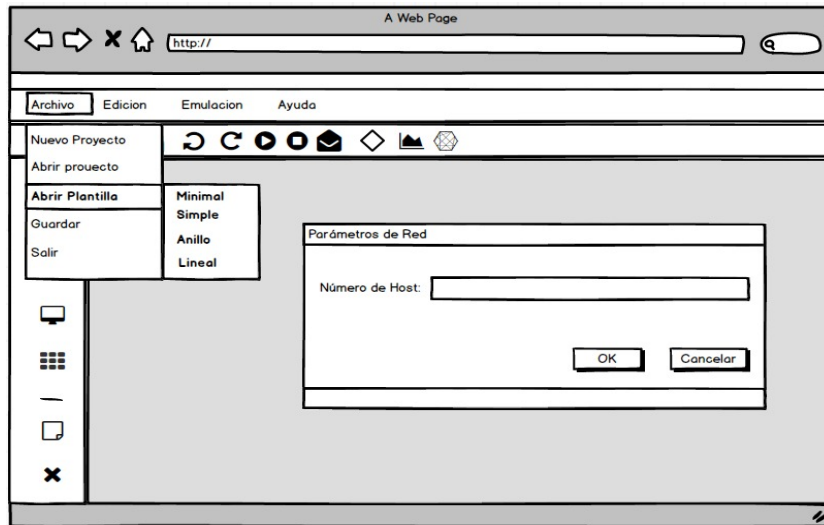


Figura 4.8: Interfaz para el diseño de topologías preestablecidas

Bosquejo: Interfaz de configuración del tráfico

Una de las principales herramientas del ambiente de pruebas es la configuración de los parámetros de tráfico a generar en una red ya emulada. La figura 4.9, lista algunos de los parámetros configurables soportados por el gestor, parametrizando el tipo de tráfico, el modo de operación y distribución.

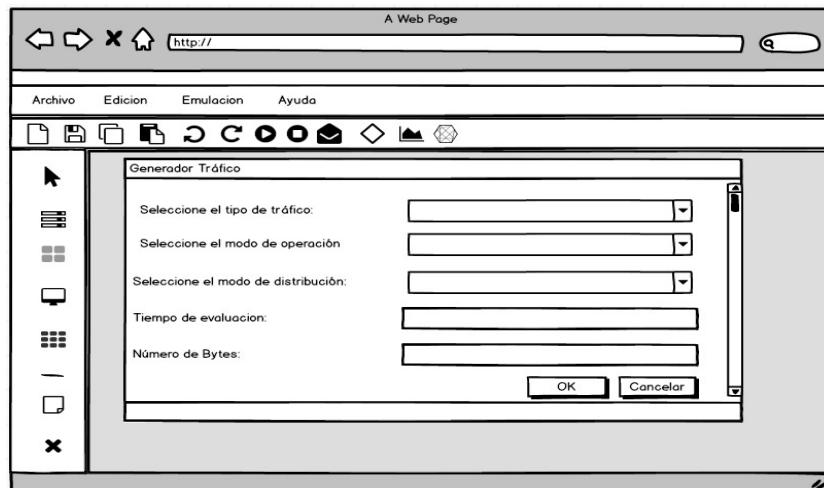


Figura 4.9: Interfaz para la configuración de tráfico de la red

Bosquejo: Interfaz de visualización de resultados

La exposición de resultados es la condición de éxito del desarrollo del proyecto como consecuencia de la emulación ejecutada; cuenta con dos tipos de exposición, la primera, un despliegue gráfico de resultados y la segunda, un panel de métricas de desempeño que valorizan cada parámetro con base a los efectos del comportamiento en la red. El monitor permite al experimentador tres tipos de vista: desde la red en general, desde el servidor y desde el cliente. La figura 4.10 modela la ventana en donde será desplegado el monitor.

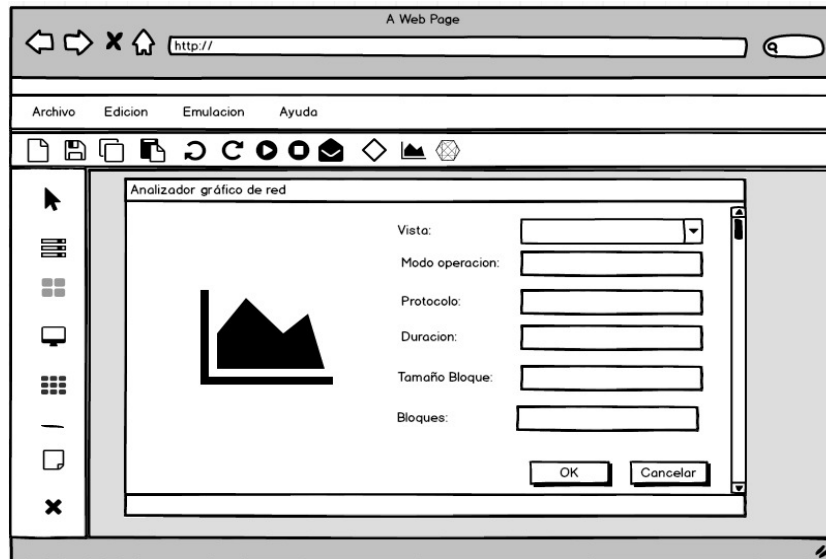


Figura 4.10: Interfaz para el despliegue de las métricas de desempeño

El diseño de los bosquejos implementados pretende cumplir con todos los requerimientos y especificaciones que un ambiente de pruebas *vSDN* requiere. El desarrollo será desplegado en una plataforma *Web*, la cual interactuara con los diferentes servicios virtualizados en las *VM* en *Telco 2.0* para consumir los recursos necesarios que ejecute la emulación.

4.2.5 Etapa *v* - Lanzamiento

En la etapa final es modelada la representación real de la interfaz *Web* del ambiente de pruebas *vSDN*, diseñada conforme a la aceptación de los contenidos y funcional-

dades definidas en las diferentes interacciones realizadas con el cliente. Los elementos y componentes de cada interfaz son desarrolladas y desplegadas con base a los estilos de color, tamaño, tipo y posición de los objetos en la plataforma *Web* del ambiente. Las interfaces representativas de esta etapa son aquellas que permiten al experimentador realizar el diseño de la red a emular, la configuración del tráfico y la monitorización del desempeño de la misma.



Figura 4.11: Interfaz de acceso al ambiente de pruebas *vSDN*

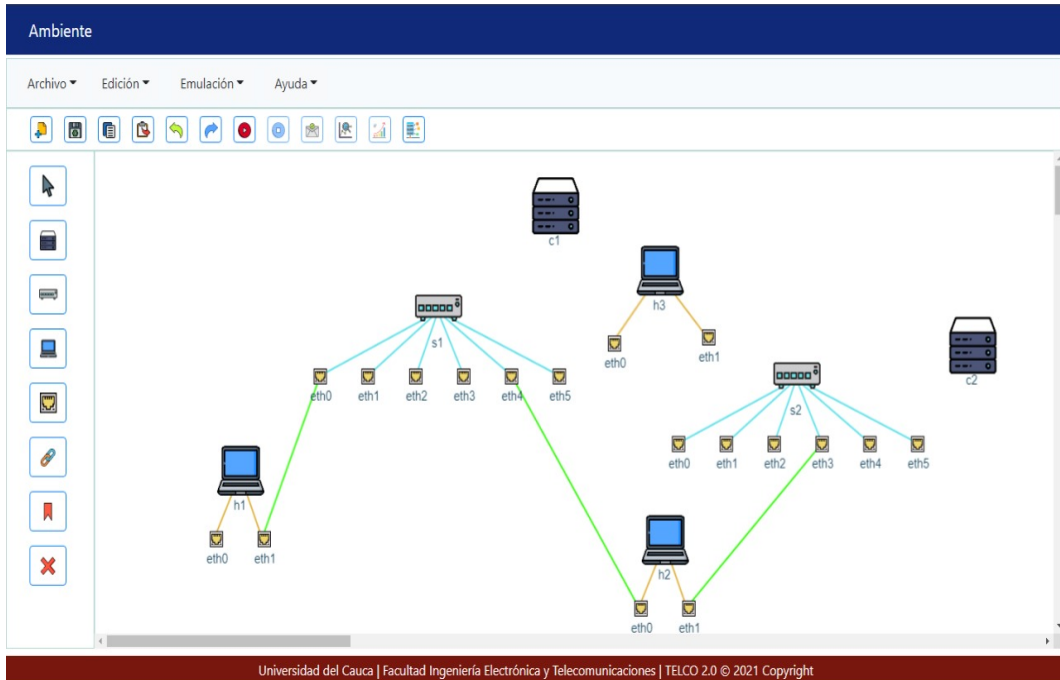


Figura 4.12: Interfaz de inicio y de diseño libre

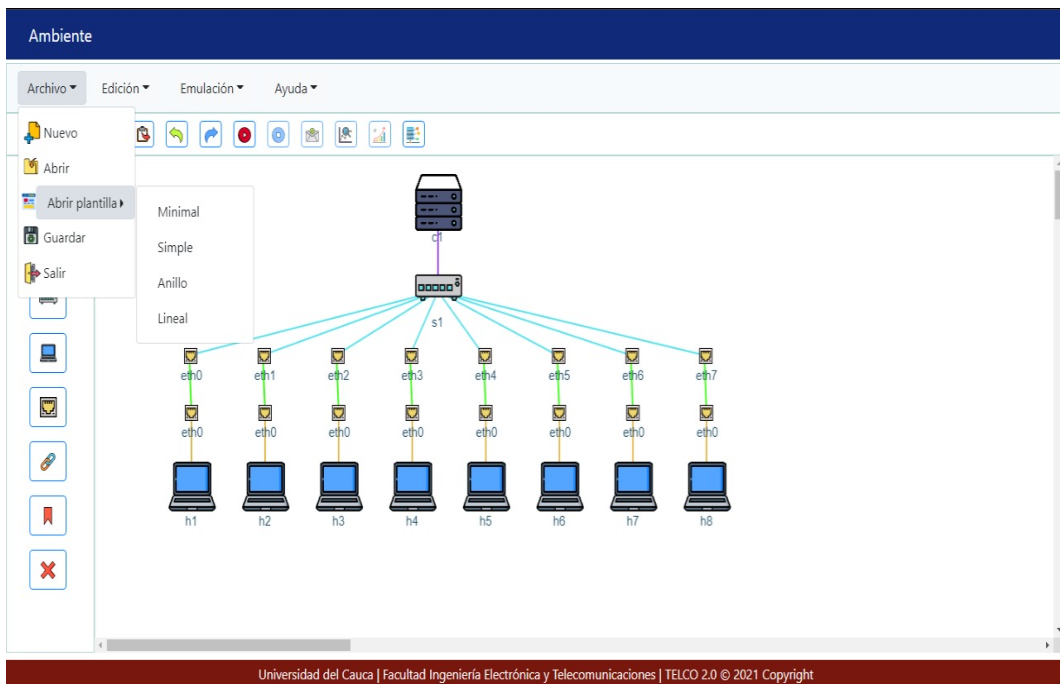


Figura 4.13: Interfaz de diseño de topologías predeterminadas

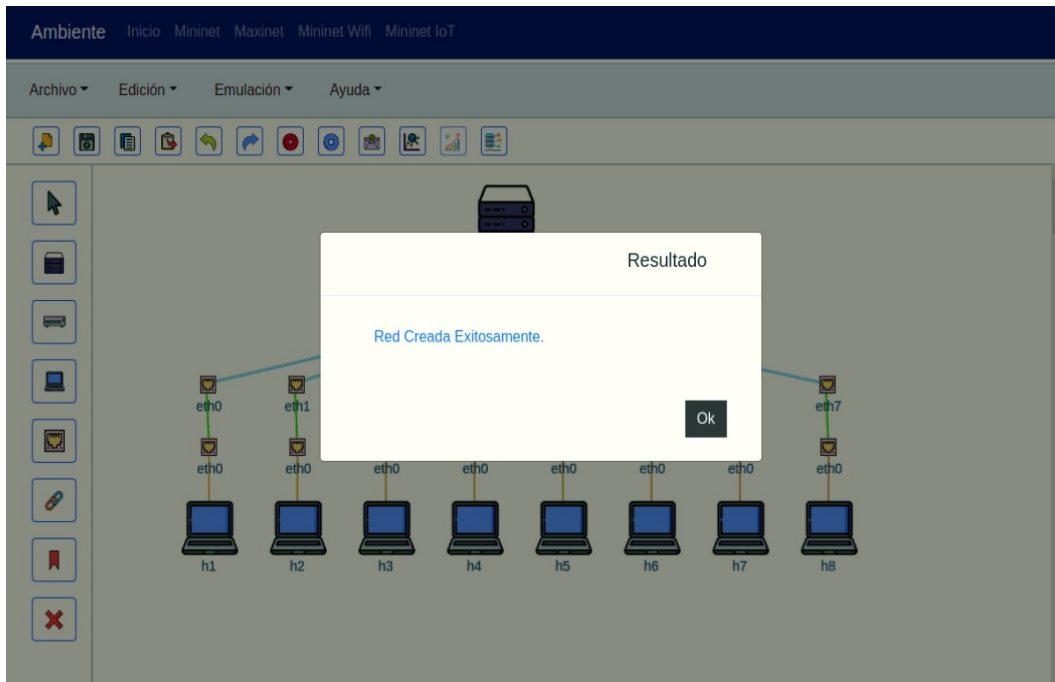


Figura 4.14: Interfaz de notificación del estado de la emulación



Figura 4.15: Interfaz de configuración del tráfico



Figura 4.16: Interfaz de despliegue de resultados

En conclusión, las interfaces del ambiente de pruebas *vSDN* implementadas, definen el prototipo final aprobado por los usuarios mediante la ejecución de diferentes pruebas de usabilidad que permiten evaluar la facilidad de uso del entorno a partir de la realización de las tareas establecidas en las historias de usuario, como lo son, la creación de redes *SDN* a gusto del usuario o preestablecidas por el ambiente y la respectiva parametrización de elementos de red, el establecimiento de cargas y patrones de tráfico necesarios para el análisis y monitorización del rendimiento de la red emulada a través de las métricas de desempeño, la capacidad de descargar un archivo contenedor del estado de diseño de la red y de las métricas de desempeño logradas. Los principales motivos de las pruebas de usabilidad son determinar las funcionalidades del ambiente de pruebas que cumplan con los objetivos de uso requeridos por parte de los experimentadores, así como también, identificar las secciones, etiquetas y botones de los despliegues *Web* que no tengan una descripción adecuada.

Las tareas que los experimentadores deben cumplir en la interfaz de usuario *Web* del ambiente de pruebas *vSDN* son resumidas a continuación:

1. En las **interfaces de inicio y diseño de red** (Figuras 4.12, 4.13) el experimentador realiza la primera interacción con los diferentes paneles disponibles para el diseño, configuración y emulación de redes *SDN*, a través de la selección, parametrización y disposición de los elementos de red que la constituyen o la elección y establecimiento de una topología predeterminada en un área de trabajo dispuesta.
2. En la **interfaz de configuración del generador de tráfico de red** (Figura 4.15) el usuario puede establecer las características del tráfico a generar con base al tipo (*UDP* y *TCP*), los modos de operación (Transmisión de paquetes simultánea entre todos los *host*, transmisión de paquetes entre dos *hosts* establecidos y transmisión de paquetes entre el primer y último *host* de la red) y distribución (Inicia el tráfico en un *host* sin esperar que acabe dicha ejecución inicia la ejecución en el siguiente *host* hasta culminar el arreglo de la red, o inicia el tráfico en un *host* y espera que culmine este proceso e inicia el siguiente *host*, así hasta culminar el arreglo de la red.), y los parámetros referentes al tiempo de emulación, el intervalo de muestreo, el ancho de banda, la longitud de la información y el tamaño del paquete y ventana deslizante.
3. En la **Interfaz para el despliegue de resultados** (Figura 4.16) el experimentador selecciona los modos de operación de los *hosts* de la red emulada (general, servidor y/o cliente), a fin de visualizar el desempeño de la misma con base a la valorización de las medidas de calidad, tales como: total y promedio de bytes transmitidos, el promedio de la velocidad de transmisión, el promedio bytes retransmitidos, el tamaño de la ventana de congestión, el promedio y variación del tiempo que tarda un paquete de datos enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor del destino (*RTT*, *Round Trip Time*) (*RTTVAR*, *Round Trip Time Variation*), el promedio máximo de la unidad de transmisión en la ruta de red entre 2 *hosts* (*PMTU*, *Path Maximum Transmission Unit*), la variación en el tiempo de llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. (*jitter*).

En resumen, el capítulo cuatro presenta el desarrollo del ambiente de pruebas *vSDN* sobre Telco 2.0, a través del modelo de diseño propuesto por *Scrum* y el despliegue en una interfaz *Web*, este modelo permite obtener un entorno práctico y usable desde el punto de vista de los usuarios a través del análisis iterativo de requisitos. Asimismo, destaca el modelo del comportamiento y arquitectura del ambiente desde un enfoque funcional. El proceso de desarrollo es evaluado iterativamente en todas sus etapas por los experimentadores a partir de entregables funcionales y no funcionales. Por último, el prototipo final será evaluado mediante una prueba de usabilidad y rendimiento desarrollada en el capítulo 5 a continuación. Esta prueba permite definir la usabilidad y utilidad del ambiente.

Capítulo 5

Evaluación y análisis

En este capítulo es presentado y descrito el proceso de implementación y valorización del ambiente de pruebas *vSDN* sobre Telco 2.0, a partir del desarrollo, integración y despliegue de los nodos diseñados en el diagrama de componentes y las vistas de la interfaz realizadas mediante los *mockups* en un prototipo funcional, en segundo lugar, el modelo desarrollado es evaluado iterativamente con los usuarios por medio de una prueba de usabilidad con el fin de detectar y corregir errores de diseño que dificulten el uso y no correspondan a la utilidad del ambiente de prueba. Asimismo, el ambiente entorno *vSDN* es probado en rendimiento con el propósito de evaluar la capacidad del servidor *Web*.

Para lograr el despliegue del ambiente virtual *SDN* en un entorno *Web* que tuviera la capacidad de brindarle al usuario las herramientas necesarias para el acceso, emulación, gestión de tráfico y resultados, fue necesario la implementación de diferentes técnicas y utilidades que ejecutadas entre sí concluyen el resultado esperado. Las herramientas mencionadas se listan y describen a continuación.

5.1 Entorno de trabajo *Web*

El ambiente de pruebas *vSDN* requiere ser desplegado en un entorno *Web*, por lo tanto, es necesario el uso de una herramienta de construcción de aplicaciones de este tipo. Con base en la experiencia fueron tenidos en cuenta diferentes entornos

de desarrollo y así poder seleccionar asertivamente el constructor a utilizar. Para facilitar esta implementación es usado un entorno de trabajo *web* intuitivo y fácil de aplicar [118] [119] [120].

Vue: Es un área de trabajo que requiere niveles de conocimiento intermedios en lenguajes programación como lo son *HTML (HyperText Markup Language)*, *CSS (Cascading Style Sheets)* y *JS (JavaScript)*, es progresivo, capaz de construir interfaces de usuario con métodos incrementales impulsando aplicaciones sofisticadas de una sola página, en combinación con herramientas modernas y librerías de apoyo [121].

Flask: Es una área de trabajo escrita en *Python* que permite la creación de aplicaciones de forma rápida y con un número de instrucciones bajo. Simplifica y hace más fácil la creación de un servidor *web* bajo el patrón de modelo vista controlador [122]. Este entorno facilita la comunicación con las diferentes herramientas instanciadas en el servidor gracias al lenguaje nativo en que está implementado [123].

5.2 Emulador de Red

En los procesos de *I+D+i* de entornos *SDN*, el desempeño de diferentes tipos de red basados en características específicas requeridas es indispensable. Además, contar con un sistema real para la experimentación es costoso o virtualmente imposible; por ende, para realizar estudios específicos los emuladores de red son necesarios, permitiendo diseñar experimentos en condiciones controladas, garantizando homogeneidad y confiabilidad en las respuestas de los estudios [124]. A continuación, es descrito el emulador de red que constituye el ambiente de prueba *vSDN*:

Mininet: Posibilita el despliegue de un ambiente de pruebas virtual y un entorno de desarrollo para *SDN* sobre los recursos básicos (procesamiento, unidades de entrada y salida de datos) de un ordenador o máquina virtual [125]. Virtualizando los *hosts*, *switches* y enlaces, *Mininet* permite el prototipado rápido de redes y la experimentación de diferentes topologías sin necesidad disponer de una red física [126].

5.3 Analizador de protocolos

Como complemento para el análisis y evaluación de una red frente a la generación de tráfico es necesario el uso de una herramienta que permita analizar y depurar protocolos y aplicaciones de red, capturando diversas tramas de red para analizarlas [127].

Wireshark: Es una herramienta didáctica que permite el análisis de datos y protocolos a través de la captura de tráfico de cualquier red, ya sea física o virtualizada [128]. Permite monitorizar todos los paquetes de red que pasan por un equipo sugerido, conteniendo la información detalla que ayuda al estudio de la red diseñada y así poder orientar el desempeño y prevenir posibles problemas en la misma [129]. La gran capacidad de filtración y utilidades que despliega *wiresark* lo hace una utilidad que puede ser acoplada con facilidad al ambiente de pruebas *vSDN*.

5.4 Generador de tráfico

Entre las herramientas sugeridas por el experimentador y la literatura [130] [131] es resaltable la necesidad de configurar y ejecutar tráfico en una red, por lo tanto, es imperativo el uso de una utilidad como esta, permitiendo la experimentación aplicada para evaluar el rendimiento de las redes y la infraestructura, modelando y simulando paquetes de comunicación y cargas útiles desplegadas en una red.

IPERF: Es una herramienta de código abierto escrita en lenguaje *C++* que es usada ampliamente para medir el rendimiento máximo alcanzable entre conexiones punto a punto. *IPERF* permite parametrizar pruebas en una red creando flujos de datos *TCP* y *UDP* con la capacidad de medir el rendimiento entre dos extremos de la comunicación, unidireccional o bidireccionalmente, contiene un informe con marcas de tiempo con la cantidad de datos transmitidos y el rendimiento medido [132].

Los módulos descritos anteriormente son integrados y configurados con el fin de proveer la funcionalidad y usabilidad del ambiente de pruebas *vSDN* sobre Telco 2.0. La implementación e integración de cada módulo permite obtener un prototipo final

funcional cuya facilidad de uso y utilidad es evaluado con los usuarios del ambiente mediante una prueba de usabilidad; además, de una prueba de funcionamiento que permita considerar el rendimiento de las redes diseñadas a través de diferentes cargas de trabajo. La Figura 5.1 visualiza la arquitectura del prototipo final del ambiente de pruebas *vSDN* implementado mediante un diagrama de despliegue.

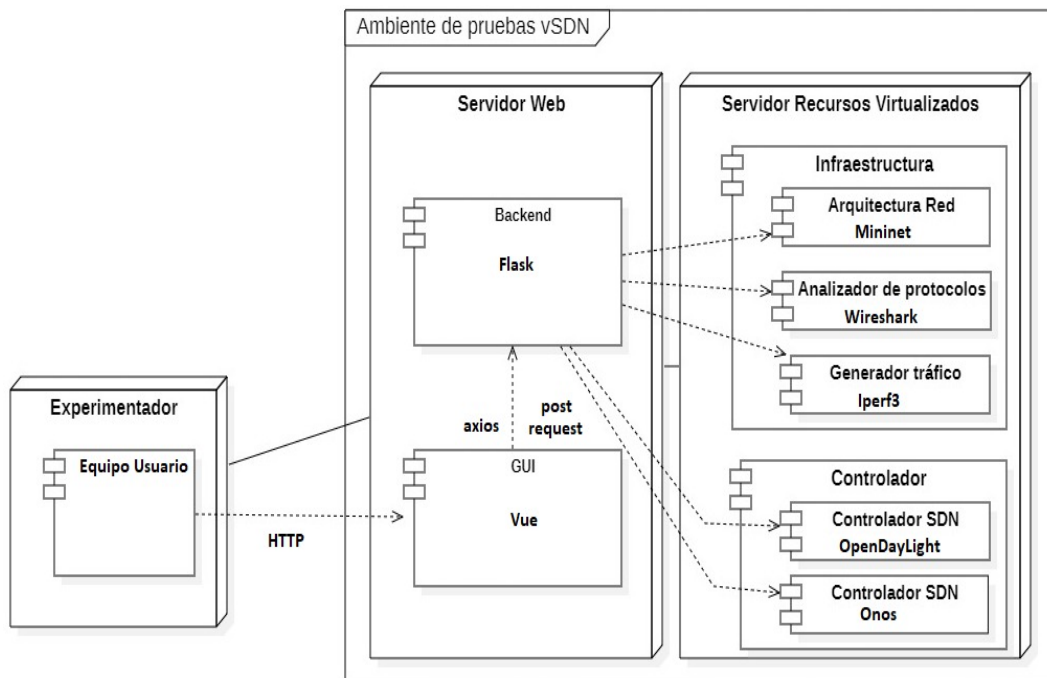


Figura 5.1: prototipo final del ambiente de pruebas *vSDN*

Según la modularidad vista en la figura 5.1 el ambiente de pruebas *vSDN* es desplegado con la siguiente configuración. Para conectarse a la interfaz principal de la aplicación, el equipo de usuario debe estar dispuesto en la red de Telco 2.0; el experimentador ingresará a la interfaz de gráfica de usuario del ambiente a través de un navegador *Web* accediendo a la siguiente dirección 10.55.6.186 y puerto 8080; las solicitudes realizadas desde la interfaz de usuario al servidor pueden estar dirigidas a dos entornos diferentes que pueden ser infraestructura y controladores, que a su vez son divididos en dos dominios más, si la solicitud es dirigida al servidor de infraestructura esta comunicación debe dirigirse al dominio 10.55.5.188 y puerto 5000, ya que en esta dirección está alojado el servidor mencionado; para el acceso servidores de los controladores el direccionamiento debe ser dirigido a los dominios

10.55.6.160:8181 (*ONOS*) y 10.55.6.159:8181 (*OpenDayLight*).

La configuración física de las *VMs* utilizadas para el despliegue de los servidores descritos cuentan con una capacidad de 8 núcleos de procesamiento, 16 GB de *RAM* y 100 GB de almacenamiento con un sistema operativo *Linux* de distribución *Ubuntu* 16.

5.5 Ejecución y pruebas

Esta sección evalúa el funcionamiento y usabilidad del ambiente de pruebas *vSDN* sobre Telco 2.0. El funcionamiento es determinado mediante el generador *IPERF*, permitiendo gestionar tráfico **TCP** y **UDP** en diferentes configuraciones de red emuladas por los usuarios del ambiente. Por otro lado, la usabilidad es calculada mediante una encuesta de experiencia de usuario realizada sobre las interfaces *Web* del entorno. Esta encuesta es diseñada a partir de heurísticas de usabilidad enfocadas en validar las funcionalidades y facilidad de uso del ambiente.

La evaluación y la ejecución de las pruebas de funcionamiento serán tomadas mediante el despliegue de las topologías de red, el tráfico gestionado a través de ella y los resultados que valorizan las métricas de desempeño de la red ante este tráfico.

Debido a que el entorno del ambiente de pruebas *vSDN* está desplegado en Telco 2.0 y cuenta con recursos limitados, las pruebas a ejecutar son dirigidas para garantizar la menor probabilidad de pérdida de información en respuesta a las solicitudes realizadas desde la *GUI*. Los escenarios son mostrados a continuación y comprenden tres diseños diferentes de la topología de red simple, representando despliegues de redes pequeñas, medianas y grandes, estas topologías serán sometidas a diferentes configuraciones de cargas de tráfico sugeridas en tres modos: Carga por defecto del generador, definición del tiempo de emulación o el número de *bytes* que desean ser transmitidos en la ejecución, además, de contar con diferentes parametrizaciones que alimentan la configuración del tráfico.

5.5.1 Rendimiento de la red

En general, el rendimiento de una red es determinado por la Calidad de los Servicios (*QoS*) y el comportamiento de sus componentes, puesto que cada red es diferente en su naturaleza y diseño [133]. En lo que respecta a las *virtual SDN*, la cantidad de tareas logradas por los componentes de la red en comparación con el tiempo y recursos que se utilizan son referidos [134]. Facilitando la detección, cuantificación y solución de anomalías presentes en la red, como la cantidad de interacciones entre las aplicaciones y las cargas de trabajo que comparten la infraestructura física, permiten distribuir y procesar de forma flexible grandes flujos de tráfico, eliminar cuellos de botella y reservar recursos de red, a fin de evitar la sobrecarga [135]. Por lo tanto, es de inferir que tanto el tráfico en las *vSDN* como en las redes tradicionales de conmutación de paquetes refieren a flujos aleatorios de las aplicaciones de red [136].

La evaluación de métricas como la disponibilidad, pérdida, retraso y utilización, permiten tasar el desempeño de la red y la confiabilidad de sus componentes individuales. Por lo tanto, es necesario conocer la valorización planteada en la Figura 5.2.

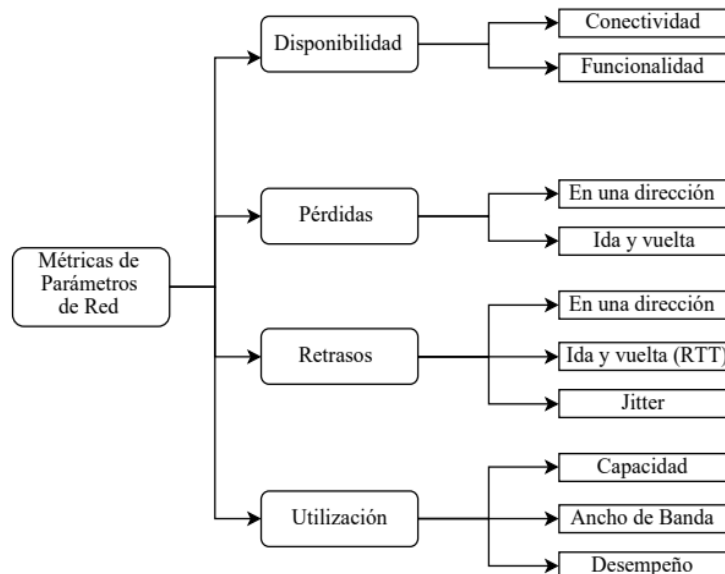


Figura 5.2: Métricas de desempeño

- **Disponibilidad:** Referente a la funcionalidad de la red, es decir, los dispositivos de red funcionan o no.
- **Pérdidas:** Son la cantidad de paquetes que no llegan a su destino dentro de un intervalo de tiempo, es un valor porcentual y puede ser de dos tipos: pérdidas de paquetes en una dirección o pérdida de paquetes en ida y vuelta.
- **Retraso:** Es el tiempo necesario para que un paquete viaje en una dirección o en varias (ida y vuelta). La variación del retraso se conoce como *jitter*.
- **Utilización:** Es el rendimiento del enlace expresado como un porcentaje de su capacidad.

5.5.2 Análisis de resultados

Para llevar a cabo la prueba de funcionamiento, el diseño de red es desplegado en una configuración simple de 20 *host*, sometida a cargas de trabajo parametrizadas en tiempo de emulación y bytes a transmitir mediante peticiones simultáneas de tráfico TCP entre todos los elementos de red. Los resultados de valores como el promedio total de *bytes* transmitidos, *bytes* retransmitidos, la velocidad de transmisión y el tiempo de retardo proporcionan los datos estadísticos de análisis del funcionamiento de la red.

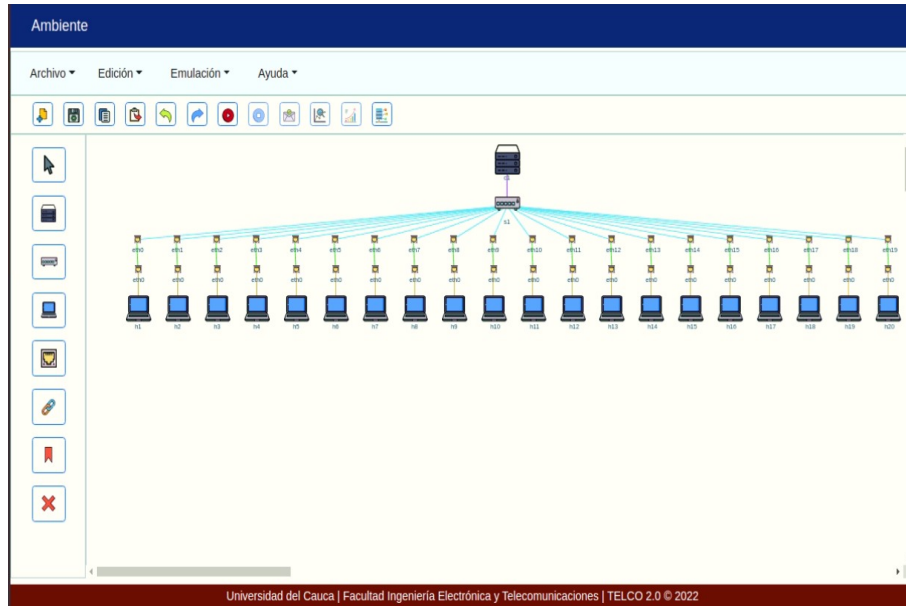


Figura 5.3: Topología *SDN* en configuración simple de 20 *hosts*

En la figura 5.3 es presentada la red diseñada, la cual es cargada con tráfico durante 60 segundos con la configuración por defecto que contiene el generador *IPERF3*, la cual consta de, un tamaño de bloque de 131.72 *bytes* con un tamaño máximo del segmento *TCP* de 1440 *bytes*, un buffer de envío y recepción de 87.380 *bytes*.

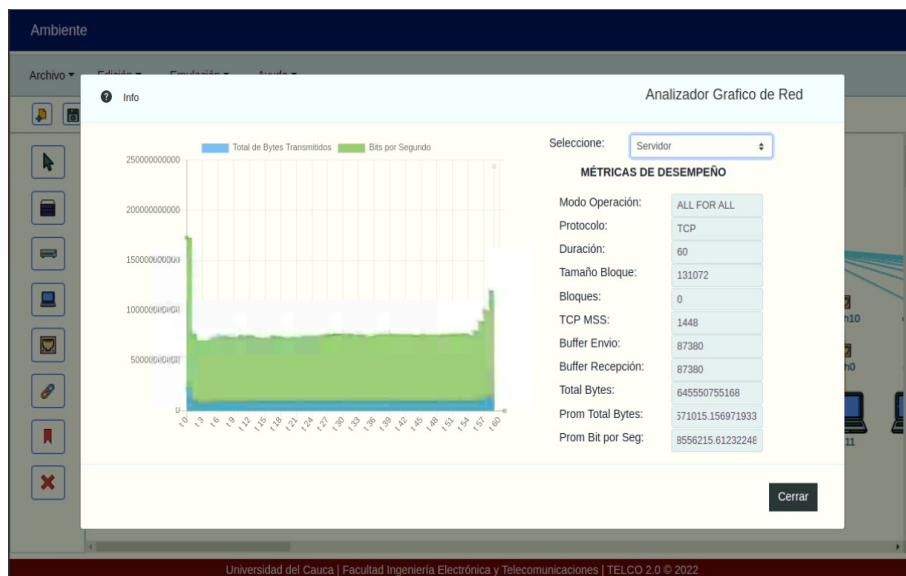


Figura 5.4: Vista general del desempeño de la red *SDN*

Lo anterior da como resultado tres diferentes métricas de desempeño para tres diferentes vistas, la figura 5.4 presenta los valores de las medidas obtenidas desde la vista general de la red, la figura 5.5 presenta los valores de las medidas obtenidas desde la vista desde el cliente y la figura 5.6 presenta los valores de las medidas obtenidas desde la vista desde el servidor, cabe denotar que, durante 60 segundos con las especificaciones dadas la red logro transmitir alrededor de 1.204 GB de información entre todos los *hosts*, en sus dos modos de funcionamiento (servidor, cliente). En la vista cliente es posible ver que en una transmisión de alrededor de 648 GB el promedio de paquete de retransmitidos es de 4.480, con promedio de tiempo de tardanza del mismo desde un emisor pasando por el receptor y volver hasta el mismo es de 2280.84 ms.



Figura 5.5: Desempeño de la red *SDN* vista desde el cliente

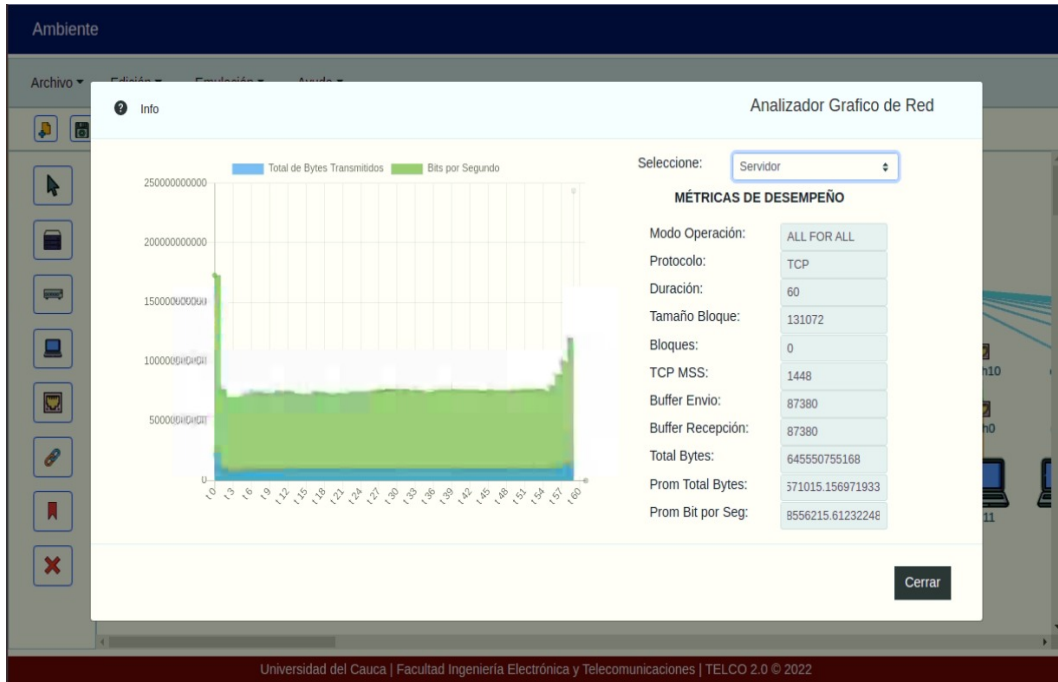


Figura 5.6: Desempeño de la red *SDN* vista desde el servidor

Para la segunda prueba, es necesario recordar que *IPERF3* envía tráfico en una cantidad de tiempo establecida, por lo cual, al establecer 500 *MB* como el número de *buffers* a transmitir, este generador envía una matriz de longitud de *bytes* ciertas veces sin importar cuanto tiempo tome.

El despliegue de resultados de las métricas de desempeño denotan la valorización en diferentes parámetros de medición, el total de *bytes* percibidos desde la vista general de la red son alrededor de 395 *GB*, a una velocidad de transmisión promedio total de 19.5 *MB/s*; el desempeño visto desde el cliente tiene un promedio de tiempo *RTT* igual a 1532.7 *ms* y un promedio de 5.5 paquetes retransmitidos.

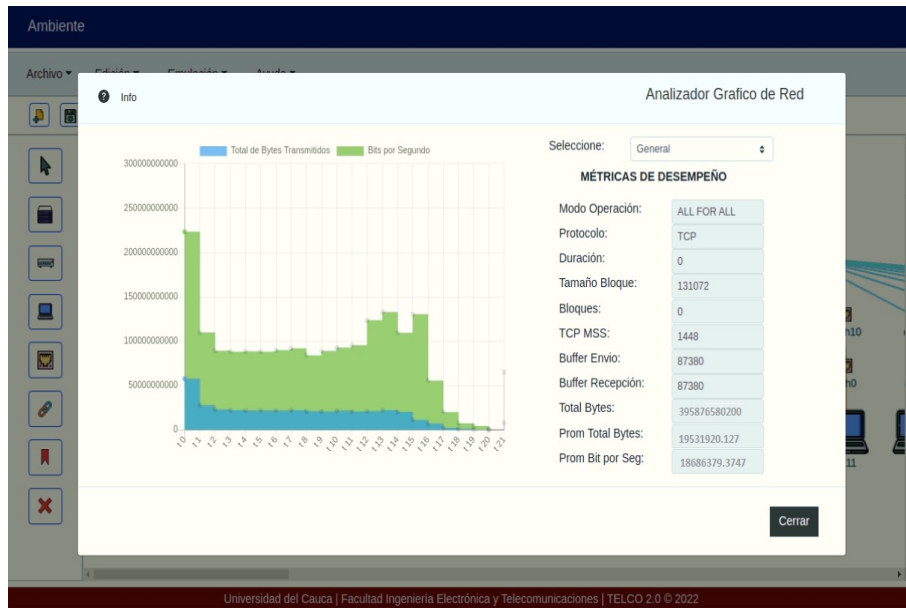
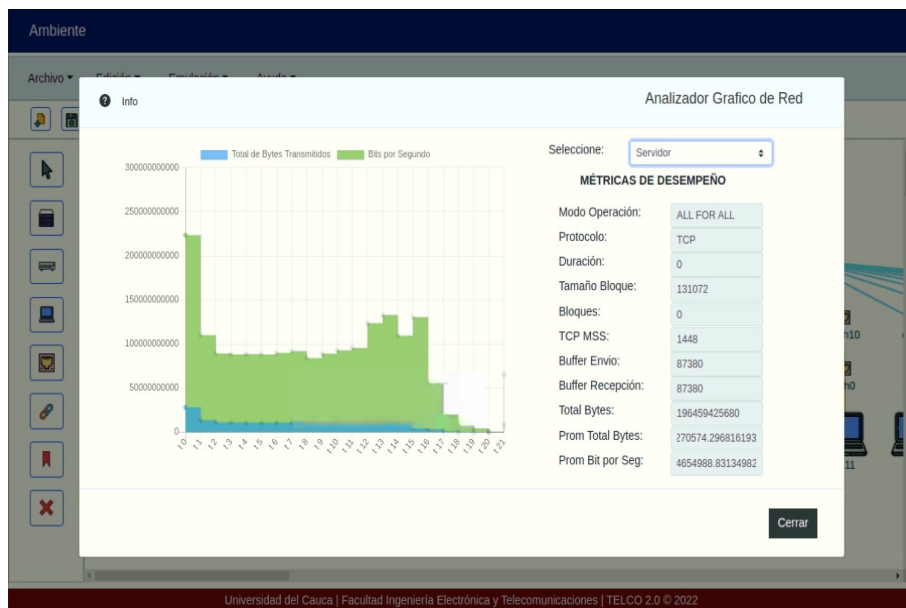
Figura 5.7: Vista general del desempeño de la red *SDN*Figura 5.8: Desempeño de la red *SDN* vista desde el servidor



Figura 5.9: Desempeño de la red *SDN* vista desde el cliente

A pesar de que las dos ejecuciones sobre la misma topología de red, una por tiempo de emulación (60 segundos) y una por número de *bytes* transmitidos (500 MB) son emuladas con los mismos parámetros de configuración por defecto del generador, es denotable que su desempeño es proporcional al tamaño de carga del tráfico y la velocidad de transmisión, debido a que, mientras la carga de tráfico aumenta, los tiempos de respuesta de los *host* servidores a las peticiones realizadas por los *host* clientes también aumentan. Por lo tanto, podemos inferir que el desempeño de la red es aproximadamente proporcional al tamaño de la carga de trabajo aplicado sobre la misma, concluyendo:

- Las peticiones de tráfico en cola disminuyen la capacidad de respuesta de los elementos de red, por lo cual el ambiente de pruebas aumentando los tiempos de emulación.
- El rendimiento de las redes diseñadas en el ambiente de pruebas *vSDN* depende de los tiempos de respuesta del servidor ejecutado en los elementos de red tipo *host*.

Evaluar la utilidad y usabilidad es fundamental para garantizar la facilidad de uso y satisfacción del usuario con las interfaces y funcionalidad de una plataforma. Por ello, el ambiente de pruebas *vSDN* sobre Telco 2.0 ha sido validado y refinado iterativamente con los usuarios del entorno en todas las diferentes etapas del proyecto mediante prototipos no funcionales. En la etapa de desarrollo, las herramientas de *software* y *hardware* son integradas en una plataforma *Web* encargada de garantizar la funcionalidad del ambiente en un prototipo funcional. Este prototipo es evaluado mediante una prueba de experiencia de usuario enfocada validación del proyecto y usabilidad del ambiente.

Para validar esta prueba, es evaluada la utilidad y el nivel de dificultad para utilizar las funciones y herramientas del ambiente de pruebas *vSDN*. La prueba es realizada mediante un cuestionario diseñado a partir de criterios de evaluación definidos en las historias de usuario que especifican funciones a realizar sobre la plataforma *Web* del entorno como ingresar, diseñar topologías de red en forma libre o predeterminada, configurar y ejecutar tráfico y monitorizar las métricas de desempeño. La Tabla 5.1 muestra el cuestionario usado en la prueba de validación de prototipo.

Ítem	Criterios
1	Ingresar al dominio de la <i>GUI</i> en Telco 2.0.
2	Ingresar a la interfaz de edición y creación del ambiente de pruebas.
3	Creación de una topología de red en el modo de diseño libre.
4	Creación de una topología de red en el modo prediseñado.
5	Configuración de los elementos de red.
6	Emular de la red <i>vSDN</i> .
7	Configurar y ejecutar la parametrización de tráfico a fin de emular la red diseñada.
8	Establecer los modos de funcionamiento del generador de tráfico en las redes <i>vSDN</i> .
9	Modificar los parámetros de tráfico a fin de volver a emular la red diseñada
10	Acceso al analizador de protocolos.
11	Acceso a la interfaz de configuración del emulador externo <i>OpenDaylight</i> .
12	Acceso a la interfaz de configuración del emulador externo <i>ONOS</i> .
12	Detener la emulación
13	Exportar datos de estados del diseño de la red.
14	Exportar resultados métricas de desempeño de las redes emuladas.

Table 5.1: Cuestionario validación de prototipo

La prueba fue realizada policéntricamente a 12 experimentados encargados de utilizar las funcionalidades proporcionadas por la interfaz *Web* del ambiente de pruebas *vSDN*. Las respuestas del cuestionario de validación de prototipo indican el nivel de dificultad (fácil, medio, difícil) para realizar una tarea sobre el entorno, además de recopilar comentarios y sugerencias para mejorar su utilidad y usabilidad. La figura 5.10 presenta los resultados obtenidos de la prueba realizada.

Criterios	Respuestas Actores Experimentador (12)		
	Fácil	Medio	Difícil
C1	11	1	0
C2	12	0	0
C3	7	4	1
C4	9	3	0
C5	4	7	1
C6	10	2	0
C7	5	4	3
C8	5	4	3
C9	5	4	3
C10	12	0	0
C11	12	0	0
C12	12	0	0
C13	10	2	0
C14	10	2	0

Figura 5.10: Validación del prototipo final

La figura 5.10 detalla el cumplimiento de cada una de las tareas asignadas en la interfaz *Web* ambiente y la facilidad de uso por medio del cuestionario de validación de prototipo. Los resultados obtenidos son explicados a continuación.

- Las respuestas resaltadas en color verde indican que la mayoría de usuarios pudieron realizar con éxito y sin dificultad la funcionalidad del ambiente de pruebas. Estas funcionalidades facilitaron el uso, debido a que la interfaz de despliegue proporciona una secuencia de información lógica para la ejecución.
- Las respuestas en amarillo corresponden a criterios de mayor dificultad en la configuración, y el alto índice, es referido al nivel de conocimientos que obtenidos para la acción a ejecutar.
- Las respuestas resaltadas en color rojizo indican que los usuarios tuvieron cierta dificultad en el cumplimiento de los objetivos, también es debido al nivel de conocimiento que es requerido para la gestión de dicha labor.

Capítulo 6

Conclusión y Trabajo Futuro

Este capítulo presenta las conclusiones obtenidas en el trabajo desarrollado y la evaluación de funcionamiento y usabilidad realizada sobre la plataforma *Web* del ambiente de pruebas *vSDN* sobre Telco 2.0. Finalmente, propone un direccionamiento de trabajos futuros.

6.1 Conclusiones

Este desarrollo responde a la pregunta de investigación **¿Cómo ofrecer un ambiente de prueba vSDN en la Plataforma Telco 2.0 de la Facultad de Ingeniería Electrónica de la Universidad del Cauca?**. Para responder el interrogante, este trabajo de grado propone un modelo de diseño, construcción e implementación basado en *Scrum*, a través del desarrollo iterativo de un prototipo funcional del ambiente, en los cuales el experimentador es el actor en el diseño, implementación y evaluación del entorno. El enfoque policéntrico del modelo permite recopilar información útil (requisitos de usuario) de disposiciones deseadas por el usuario que interactúa con la plataforma *Web* del ambiente de pruebas *vSDN*, a fin de mejorar y complementar las funcionalidades, definiendo el alcance del proyecto con relación a los requisitos del usuario.

La utilización del modelo de diseño basado en *Scrum* en el desarrollo del ambiente permitió obtener interfaces de usuario intuitivas, amigables y fáciles de utilizar

desarrolladas a bajo costo y refinadas iterativamente con los usuarios. Asimismo, permite definir las herramientas de *software* utilizadas en la ejecución del proyecto para garantizar su usabilidad y utilidad.

El prototipo final incluye la implementación de tres servidores de recursos virtualizados, encargados de gestionar las herramientas para el despliegue de arquitecturas de red virtuales, la generación de tráfico y el análisis de protocolos instanciadas en una *VM* y el acceso a controladores externos que pueden parametrizar la red. El prototipo final del ambiente fue evaluado con los usuarios mediante una prueba de usabilidad y una prueba de funcionamiento. La prueba de usabilidad fue diseñada a partir de tres ejes: Primero, prueba de validación de prototipo diseñada a partir de los criterios de evaluación definidos en las historias y utilizada para evaluar la funcionalidad del ambiente de pruebas *vSDN* sobre Telco 2.0 a partir de la interacción de los usuarios con sus interfaces, segundo, cuestionario de validación de contenidos diseñado con base a heurísticas de usabilidad y utilizado para evaluar la facilidad de uso y navegación de los usuarios con las *GUI* del entorno. Tanto la prueba de validación de prototipo, como la prueba de validación de contenidos, fueron incluidos elementos con comentarios, donde los usuarios realizaron recomendaciones, sugerencias y críticas para refinar el prototipo final.

La prueba de satisfacción del usuario realizada por medio de la evaluación de los criterios de aprobación, evalúan la usabilidad del ambiente a partir de las emociones y percepciones de los usuarios al interactuar con las funcionalidades del entorno. El alto índice de factibilidad en las respuestas de los experimentadores indican que el nivel de usabilidad de la plataforma *Web* del ambiente de pruebas es “Bueno”. No obstante, estas mejoras fueron realizadas en las iteraciones finales del desarrollo del ambiente de pruebas.

Finalmente, la pruebas de funcionamiento evalúan la capacidad del entorno de emulación de ejecutar las tareas descritas según su contexto indican, como lo son, el despliegue de diferentes topologías con diferentes parametrizaciones en su implementación (número de elementos de red y la configuración de los mismos que pueden limitar o encaminar de diferentes formas el diseño), la gestión de parametrización del tráfico y el cargue de este en la red emulada, siendo evaluado a través de las métricas de desempeño arrojadas desde el servidor. Cabe resaltar que, las

diferentes peticiones y configuraciones son limitadas conforme al nivel y cantidad de requerimientos pedidos al servidor, específicamente características como lo son una enorme cantidad de elementos de red, gran cantidad de datos de transmisión o tiempo de ejecución sobrecarga proporcionalmente de manera directa a los recursos físicos de las *VM*, concretamente los tiempos de respuesta llegan a ser bastante largos debido a la exigencia de la emulación, y en algunos casos, las solicitudes ejecutadas no tienen una respuesta exitosa debido al colapso del consumo de servicios en el servidor.

6.2 Trabajo Futuro

A partir del diseño e implementación del ambiente de pruebas *vSDN* sobre Telco 2.0 desarrollado en el presente trabajo de grado, presentamos algunas ideas para continuar avanzando en esta investigación en trabajos futuros.

- Adaptar el ambiente de pruebas a otras configuraciones de red, para añadir nuevas funcionalidades y entornos de emulación.
- Añadir e integrar diferentes herramientas de generación de tráfico que permitan al usuario elegir entre más configuraciones que las desplegadas en el desarrollo actual.
- Despliegue de un entorno de gestión de proyectos que permita al experimentador la creación, edición y eliminación de emulaciones para diferentes entornos de emulación gestionado en la nube.
- Utilizar un sistema de monitorización y generación de tráfico en la red para investigación de disminución de consumo de recursos virtualizados utilizados para proveer y utilizar eficientemente los servicios del ambiente.

Bibliografía

- [1] Z. I. Rasool, R. S. A. Ali, and M. M. Abdulzahra, “Network management in software-defined network: A survey,” *IOP Conference Series: Materials Science and Engineering*, vol. 1094, no. 1, p. 012055, feb 2021. [Online]. Available: <https://doi.org/10.1088/1757-899x/1094/1/012055>
- [2] Tomonori Aoyama, “A new generation network beyond ngn,” in *First ITU-T Kaleidoscope Academic Conference - Innovations in NGN: Future Network and Services*, May 2008, pp. 3–10.
- [3] A. Carrillo, M. Sánchez, and J. Villalobos, “Plan nacional de ciencia, tecnología e innovación para el desarrollo del sector de las tecnologías de la información y las comunicaciones (tic),” 2016.
- [4] O. N. F. Open Network Foundation, “Strategic plan,” in *ONF Broad*, 2012. [Online]. Available: <https://www.opennetworking.org>
- [5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, “A survey on software-defined networking,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, Firstquarter 2015.
- [6] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using openflow: A survey,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 493–512, First 2014.
- [7] K. Benzekki, A. El Fergougui, and A. El Belrhiti El Alaoui, “Software-defined networking (sdn): A survey,” in *Security and Communication Networks*, 02 2017.

-
- [8] O. N. F. Open Network Foundation, “Strategic plan,” in *ONF Broad*, 2012. [Online]. Available: <https://www.opennetworking.org>
- [9] G. Yang, C. Shin, Y. Yoo, and C. Yoo, “A case for sdn-based network virtualization,” in *2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2021, pp. 1–8.
- [10] C. Tipantuña and P. Yanchapaxi, “Network functions virtualization: An overview and open-source projects,” in *IEEE Second Ecuador Technical Chapters Meeting (ETCM), Salinas Ecuador*, 2017. [Online]. Available: <https://ieeexplore.ieee.org>
- [11] D. Staessens, “Large-scale rina experimentation on fire deliverable 4.2 report on available experimental infrastructure,” in *ARCFIRE*, vol. 0.1, pp. 1–16, 2016. [Online]. Available: <http://ict-arcfire.eu>
- [12] A. Willner, S. Albrecht, S. Covaci, F. Schreiner, T. Magedanz, S. Avessta, C. Scognamiglio, S. Fdida, and U. Bub, “Fantaastic: Sustainable management of future internet testbed federations,” in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–4, 2014. [Online]. Available: <https://ieeexplore.ieee.org>
- [13] R. P. Loera, V. M. M. Rocha, and J. I. H. Hernández, “Redes definidas por software: beneficios y riesgos de su implementación en universidades,” *Segundo Número del Congreso ANIEI, Especial de TECNOLOGÍA EDUCATIVA – REVISTA CONAIC*, 2015. [Online]. Available: <https://conaic.net>
- [14] S. Patidar, D. Rane, and P. Jain, “A survey paper on cloud computing,” in *Second International Conference on Advanced Computing Communication Technologies*, Jan 2012, pp. 394–398.
- [15] A. O. Adedayo and B. Twala, “Qos functionality in software defined network,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2017, pp. 693–699.
- [16] A. L. V. Caraguay, L. I. B. Lopez, and J. G. Villalba, “Evolution and challenges of software defined networking,” in *IEEE*, pp. 49–55, 2013.

- [Online]. Available: <https://ieeexplore.ieee.org>
- [17] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, “GENI: A federated testbed for innovative network experiments,” *Computer Networks*, vol. 61, no. 0, pp. 5–23, Mar. 2014. [Online]. Available: <https://www.geni.net/>
- [18] D. Medhi, B. Ramamurthy, C. M. Scoglio, J. P. Rohrer, E. K. Çetinkaya, R. Cherukuri, X. Liu, P. Angu, A. C. Bavier, C. Buffington, and J. P. G. Sterbenz, “The gpeni testbed: Network infrastructure, implementation experience, and experimentation,” *Computer Networks*, vol. 61, pp. 51–74, 2014.
- [19] C. H. Benet, R. Nasim, K. A. Noghani, and A. Kassler, “Openstackemu — a cloud testbed combining network emulation with openstack and sdn,” in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2017, pp. 566–568.
- [20] J. Mambretti, J. Chen, and F. Yeh, “Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn),” in *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, Oct 2015, pp. 73–79.
- [21] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, “A survey on large-scale software defined networking (sdn) testbeds: Approaches and challenges,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 891–917, Secondquarter 2017.
- [22] B. Yi, X. Wang, K. Li, S. k Das, and M. Huang, “A comprehensive survey of network function virtualization,” *ELSEVIER*, vol. 133, pp. 212–262, 2018. [Online]. Available: <https://www.sciencedirect.com>
- [23] A. Bauer, N. Herbst, S. Spinner, A. Ali-Eldin, and S. Kounev, “Chameleon: A hybrid, proactive auto-scaling mechanism on a level-playing field,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 800–813, April 2019.

- [24] A. Saghir and T. Masood, "Performance evaluation of openstack networking technologies," in *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, 2019, pp. 1–6.
- [25] H. Li, Z. Huachun, H. Zhang, B. Feng, and W. Shi, "Emustack: An openstack-based dtn network emulation platform (extended version)," *Mobile Information Systems*, vol. 2016, pp. 1–15, 01 2016.
- [26] D. Kreutz, F. Ramos, V. Esteves, E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software defined networking: A comprehensive survey," *Proceeding of the IEEE*, 2015.
- [27] M. Mousa, A. M. Bahaa-Eldin, and M. Sobh, "Software defined networking concepts and challenges," in *2016 11th International Conference on Computer Engineering Systems (ICCES)*, Dec 2016, pp. 79–90.
- [28] K. Benzekki, A. El Fergougui, and A. El Belrhiti El Alaoui, "Software-defined networking (sdn): A survey," *Security and Communication Networks*, vol. 9, 02 2017.
- [29] S. Badotra, "A review paper on software defined networking," *International Journal of Advanced Computer Research*, vol. 8, 03 2017.
- [30] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 1–1, 01 2015.
- [31] M. Barrera Pérez, N. Y. Serrato Losada, E. Rojas Sánchez, and G. Mancilla Gaona, "State of the art in software defined networking (sdn)," *Visión electrónica*, vol. 13, no. 1, p. 178–194, ene. 2019. [Online]. Available: <https://revistas.udistrital.edu.co/index.php/visele/article/view/14424>
- [32] K. Nisar, E. R. Jimson, M. H. A. Hijazi, I. Welch, R. Hassan, A. H. M. Aman, A. H. Sodhro, S. Pirbhulal, and S. Khan, "A survey on the architecture, application, and security of software defined networking: Challenges and open issues," *Internet of Things*, vol. 12, p. 100289, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2542660520301219>

- [33] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti, “A survey on the security of stateful sdn data planes,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1701–1725, 2017.
- [34] C. D. Bermúdez Escobar, J. A. Rodríguez Taborda, and C. A. Dussan Clavijo, “Beneficios de las redes definidas por software y el protocolo openflow,” *Universidad Santiago de Cali*, 2020.
- [35] L. Yang, B. Ng, W. K. Seah, L. Groves, and D. Singh, “A survey on network forwarding in software-defined networking,” *Journal of Network and Computer Applications*, vol. 176, p. 102947, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804520304021>
- [36] A. Panda, W. Zheng, X. Hu, A. Krishnamurthy, and S. Shenker, “SCL: Simplifying distributed SDN control planes,” in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. Boston, MA: USENIX Association, Mar. 2017, pp. 329–345. [Online]. Available: <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/panda-aurojit-scl>
- [37] E. Pérez Tardío, *Un acercamiento a las Redes Definidas por Software. Arquitectura y beneficios.*, 07 2018.
- [38] W. Odom, “Introduction to controller-based networking,” 2019. [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=2995354>
- [39] J. R. Santamaria Sandoval, “La la gestión en redes definidas por software (sdn) desde la perspectiva de fcaps,” *Repertorio Científico*, vol. 23, pp. 1–12, 12 2020.
- [40] J. C. Correa Chica, J. C. Imbachi, and J. F. Botero Vega, “Security in sdn: A comprehensive survey,” *Journal of Network and Computer Applications*, vol. 159, p. 102595, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804520300692>
- [41] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, “Sdn controllers: A comparative study,” in *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, 2016, pp. 1–6.

- [42] V. S. Raju, "Sdn controllers comparison," in *Proceedings of Science Globe International Conference*, 2018.
- [43] A. V. Priya and N. Radhika, "Performance comparison of sdn openflow controllers," *International Journal of Computer Aided Engineering and Technology*, vol. 11, no. 4-5, pp. 467–479, 2019.
- [44] D. Saleh Asadollahi, B. Goswami, and A. M. Gonsai, "Software defined network, controller comparison," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, 2017.
- [45] C. Fancy and M. Pushpalatha, "Performance evaluation of sdn controllers pox and floodlight in mininet emulation environment," in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017, pp. 695–699.
- [46] S. Asadollahi and B. Goswami, "Experimenting with scalability of floodlight controller in software defined networks," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 288–292.
- [47] Y. Li, X. Guo, X. Pang, B. Peng, X. Li, and P. Zhang, "Performance analysis of floodlight and ryu sdn controllers under mininet simulator," in *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 2020, pp. 85–90.
- [48] S. Asadollahi, B. Goswami, and M. Sameer, "Ryu controller's scalability experiment on software defined networks," in *2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, 2018, pp. 1–5.
- [49] P. Bispo, D. Corujo, and R. L. Aguiar, "A qualitative and quantitative assessment of sdn controllers," in *2017 International Young Engineers Forum (YEF-ECE)*, 2017, pp. 6–11.
- [50] A. K. Arahunashi, S. Neethu, and H. V. Ravish Aradhya, "Performance analysis of various sdn controllers in mininet emulator," in *2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)*, 2019, pp. 752–756.

- [51] Y. Zhao, L. Iannone, and M. Riguidel, “On the performance of sdn controllers: A reality check,” in *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, 2015, pp. 79–85.
- [52] L. Mamushiane, A. Lysko, and S. Dlamini, “A comparative evaluation of the performance of popular sdn controllers,” in *2018 Wireless Days (WD)*, 2018, pp. 54–59.
- [53] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, “Comprehensive review of sdn controller placement strategies,” *IEEE Access*, vol. 8, pp. 170 070–170 092, 2020.
- [54] Q. Carlos Julio, “Propuesta metodológica para la selección de controladores de redes SDN a nivel empresarial,” Maestría en Telecomunicaciones y Reguación TIC, Universidad Santo Tomas, Bogota, Colombia, 2020.
- [55] D. Marples and P. Kriens, “The open services gateway initiative: an introductory overview,” *IEEE Communications Magazine*, vol. 39, no. 12, pp. 110–114, 2001.
- [56] V. Kivachuk, “Estudio de la plataforma ONOS para la gestión de asociaciones de seguridad ipsec en entornos sdn,” *Murcia, España: Universidad de Murcia - Facultad de Informática*, 2018.
- [57] C. J. Quimbayo Rodriguez, “Propuesta metodologica para la selección de controladores de redes sdn a nivel empresarial,” *Bogota, Colombia: Universidad Santo Tomas - Facultad de Ingeniería de Telecomunicaciones*, Septiembre 2020.
- [58] M. Karakus and A. Durrezi, “Service cost in software defined networking (sdn),” in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, March 2017, pp. 468–475.
- [59] D. R. Lopez, “What software networks are,” in *Telefonica I+D, Movistar*, 2016. [Online]. Available: <https://eetac.upc.edu/ca>
- [60] J. F. Fischer, M. T. Botero, H. D. M. Beck, and X. Hesselbach, “Virtual network embedding : A survey,” vol. 15. *IEEE Communications Surveys and*

- Tutorials, 2013.
- [61] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, “Network function virtualization : State-of-the- art and research challenges network function virtualization : State-of-the-art and research challenges,” vol. 18, 2015.
- [62] ETSI, “NFV - architectural framework,” vol. 1, pp. 1–21, 2013.
- [63] b. Han and E. Al, “Network function virtualization: Challenges and opportunities for innovations,” in *Communications Magazine, IEEE*, vol. 53.2, pp. 90–97, 2015.
- [64] C. Tipantuña and P. Yanchapaxi, “Network functions virtualization: An overview and open-source projects,” in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 2017, pp. 1–6.
- [65] P. v. Anvith, N. Gunavathi, B. Malarkodi, and B. Rebekka, “A survey on network functions virtualization for telecom paradigm,” in *2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW)*, 2019, pp. 302–306.
- [66] K. Kaur, V. Mangat, and K. Kumar, “Architectural framework, research issues and challenges of network function virtualization,” in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 474–478.
- [67] C. Bouras, A. Kollia, and A. Papazois, “Sdn nfv in 5g: Advancements and challenges,” in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, March 2017, pp. 107–111.
- [68] C. C. Machado, L. Z. Granville, and A. Schaeffer-Filho, “Answer: Combining nfv and sdn features for network resilience strategies,” in *2016 IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 391–396.
- [69] O. Michel and E. Keller, “Sdn in wide-area networks: A survey,” in *2017 Fourth International Conference on Software Defined Systems (SDS)*, May 2017, pp.

- 37–42.
- [70] Z. Michael, D. Allan, Shirazipour, M. Cohn, N. Damouny, C. Koliass, J. Maguire, S. Manning, D. McDysan, E. Roch, and M. Shirazipour, “Openflow-enabled sdn and network functions virtualization,” in *Open Network Foundation*, pp. 1–12, 2014.
- [71] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, “A survey on large-scale software defined networking (sdn) testbeds: Approaches and challenges,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 891–917, 2017.
- [72] A. Srivastava, S. Bhardwaj, and S. Saraswat, “Scrum model for agile methodology,” in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 864–869.
- [73] Y. Khmelevsky, X. Li, and S. Madnick, “Software development using agile and scrum in distributed teams,” in *2017 Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–4.
- [74] F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab, and M. Abbas, “The influence of agile methodology (scrum) on software project management,” in *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2019, pp. 145–149.
- [75] K. Schawber and J. Sutherland, “La guía scrum,” in *La Guía Definitiva de Scrum: Las Reglas del Juego*, 2020.
- [76] R. Kurnia, R. Ferdiana, and S. Wibirama, “Software metrics classification for agile scrum process: A literature review,” in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 174–179.
- [77] K. Bhavsar, V. Shah, and S. Gopalan, “Scrum: An agile process reengineering in software engineering,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, pp. 840–848, 2020.

-
- [78] Y. Shastri, R. Hoda, and R. Amor, “Spearheading agile: The role of the scrum master in agile projects,” *Empirical Software Engineering*, vol. 26, no. 1, pp. 1–31, 2021.
- [79] J. M. Bass, S. Beecham, M. A. Razzak, C. N. Canna, and J. Noll, “An empirical study of the product owner role in scrum,” in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings*, 2018, pp. 123–124.
- [80] M. A. Ramessur and S. D. Nagowah, “Factors affecting sprint effort estimation,” in *Advanced Computing and Intelligent Engineering*. Springer, 2020, pp. 507–518.
- [81] J. Betta, T. Chlebus, D. Kuchta, and A. Skomra, “Applying scrum in new product development process,” in *International Scientific-Technical Conference MANUFACTURING*. Springer, 2019, pp. 190–200.
- [82] L. Gonçalves, “Scrum,” *Controlling & Management Review*, vol. 62, no. 4, pp. 40–42, 2018.
- [83] M. B. Firdaus, I. M. Patulak, A. Tejawati, A. Bryantama, G. M. Putra, and H. S. Pakpahan, “Agile-scrum software development monitoring system,” in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, vol. 6, 2019, pp. 288–293.
- [84] A. Alhazmi and S. Huang, “A decision support system for sprint planning in scrum practice,” in *SoutheastCon 2018*, 2018, pp. 1–9.
- [85] S. Mustafa, P. Dey, and M. Yuksel, “Genix: A geni-based ixp emulation,” 07 2020, pp. 1091–1092.
- [86] L. Dane and D. Gurkan, “Geni with a network processing unit: Enriching sdn application experiments,” in *2014 Third GENI Research and Educational Experiment Workshop*, 2014, pp. 9–14.
- [87] A. Gosain and I. Seskar, “Geni wireless testbed: An open edge ecosystem for ubiquitous computing applications,” in *2017 IEEE International*

- Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 54–56.
- [88] W.-T. Hong, J.-U. Kong, and J.-W. Chung, “Deployment and performance analysis of nation-wide openflow networks over kreonet,” *The KIPS Transactions:PartC*, vol. 18C, 12 2011.
- [89] A. Ullah, X. Chen, and J. Yang, “Design and implementation of mobility first future internet testbed,” in *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, 2020, pp. 170–174.
- [90] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, “Mobilityfirst future internet architecture project,” 11 2011, pp. 1–3.
- [91] X. Liu, P. Juluri, and D. Medhi, “An experimental study on dynamic network reconfiguration in a virtualized network environment using autonomic management,” in *IFIP/IEEE International Symposium On Integrated Network Management: Mini-Conference*, p. 616–622, Mayo 2013.
- [92] X. Liu and D. Medhi, “Dynamic virtual network restoration with optimal standby virtual router selection,” in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 973–978.
- [93] R. Cherukuri, X. Liu, A. Bavieri, J. Sterbenz, and D. Medh, “Network virtualization in gpeni: framework, implementation and integration experience,” in *3rd IEEE/IFIP International Workshop on Management of the Future Internet*, pp. 1212–1219, Mayo 2013.
- [94] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, “Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed,” *Computer Networks*, vol. 57, no. 16, pp. 3207 – 3221, 2013, information Centric Networking. [Online]. Available: <http://www.sciencedirect.com/science/article>
- [95] M. Gerola, R. Doriguzzi Corin, R. Riggio, F. De Pellegrini, E. Salvadori, H. Woesner, T. Rothe, M. Suñe, and L. Bergesio, “Demonstrating inter-testbed network virtualization in ofelia sdn experimental facility,” in *2013*

- IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, 2013, pp. 39–40.
- [96] M. Santuari, R. Doriguzzi-Corin, M. Gerola, E. Salvadori, U. Toseef, A. Zaalouk, K. Dombek, D. Parniewicz, A. Hammad, M. Rashidi-Fard, E. Jacob, and J. Matias, “Leading the ofelia facility beyond openflow 1.0 experimentations,” in *2014 Third European Workshop on Software Defined Networks*, 2014, pp. 119–120.
- [97] H. Chang and S. Wang, “Using sdn technology to mitigate congestion in the openstack data center network,” in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 401–406.
- [98] C. Clark, K. Fraser, S. Hand, J. G. Hansen, EricJul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in *Proceedings of the 2nd conference on Symposium on Networked Systems Design Implementation*, vol. 2, 2015.
- [99] J. Mambretti, J. Chen, and F. Yeh, “Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn),” in *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, 2015, pp. 73–79.
- [100] A. Bauer, N. Herbst, S. Spinner, A. Ali-Eldin, and S. Kounev, “Chameleon: A hybrid, proactive auto-scaling mechanism on a level-playing field,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 800–813, 2019.
- [101] P. Singh and S. Manickam, “Design and deployment of openstack-sdn based test-bed for edos,” in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Sep. 2015, pp. 1–5.
- [102] E. Soca, “Estudio de una solución openstack integrada con sdn,” *Barcelona Tech, Universitat Politecnica de Catalunya*, 2015. [Online]. Available: <https://upcommons.upc.edu>

- [103] H. Le, “Openstack and software-defined networking : The enormous potential of open source software collaboration,” in *University of Applied sciences*, 2017. [Online]. Available: <http://urn.fi>
- [104] A. Avilés, A. Robert, C. Cristóbal, and M. Michell, “Diseño y simulación de una red de datacenters basada en una topología fat-tree en un ambiente de redes definidas por software,” in *Electrónica GYE*, 2015. [Online]. Available: <http://dspace.ups.edu.ec>
- [105] E. G. Sáinz, “Sdn sobre redes ieee 802.11: Simulación mediante mininet wifi,” in *E.T.S. de ingeniería industrial y de telecomunicación, Universidad de Cantabria*, 2016. [Online]. Available: <https://web.unican.es>
- [106] R. G. Callejas, “Implementación de movilidad en redes 5g: Simulación de movilidad con software defined networking,” in *E.T.S. de ingenierías informática y de telecomunicación, Universidad de Granada*, 2017. [Online]. Available: <https://www.ugr.es>
- [107] J. Vehanen, “Handover between lte and 3g radio access technologies: Test measurement challenges and field environment test planning,” G2 Pro gradu, diplomityö, 2011. [Online]. Available: <http://urn.fi/URN:NBN:fi:aalto-201207022686>
- [108] C. D. Bermúdez Escobar, J. A. Rodríguez Taborda, and C. A. Dussan Clavijo, “Beneficios de la redes definidas por software y el protocolo openflow,” *Universidad Santiago de Cali, Facultad de Ingeniería, Tecnología en Sistemas*, no. 1, 2020.
- [109] E. Tardío and H. Cruz-Enriquez, “Evaluación de desempeño de las redes sdn mediante la aplicación de mecanismos de calidad de servicio definidos para redes ip,” Ph.D. dissertation, 10 2019.
- [110] B. M. Montero, H. V. Cevallos, and J. D. Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales revista multidisciplinaria de investigación*, vol. 2, no. 17, pp. 114–121, 2018.
- [111] S. M. Velásquez, J. D. V. Montoya, M. E. G. Adasme, E. J. R. Zapata, A. A. Pino, and S. L. Marín, “Una revisión comparativa de la literatura acerca

- de metodologías tradicionales y modernas de desarrollo de software,” *Revista Cintex*, vol. 24, no. 2, pp. 13–23, 2019.
- [112] J. Salazar, A. Tovar Casallas, J. Linares, A. Lozano, and Y. Valbuena, “Scrum versus xp: similitudes y diferencias,” in *Tecnología Investigación y Academia*, vol. 6, no. 2, December 2018, p. 29–37.
- [113] G. Hernández, Á. Martínez, R. Jiménez, and F. Jiménez, “Scrum y peopeware: elementos clave para la gestión en la construcción de software,” *Revista Ibérica de Sistemas e Tecnologías de Informação*, no. E19, pp. 265–277, 2019.
- [114] M. Morandini, T. A. Coleti, E. Oliveira, and P. L. P. Corrêa, “Considerations about the efficiency and sufficiency of the utilization of the scrum methodology: A survey for analyzing results for development teams,” *Computer Science Review*, vol. 39, p. 100314, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574013720304147>
- [115] S. Sharma and N. Hasteer, “A comprehensive study on state of scrum development,” in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 867–872.
- [116] J. P. Subra and A. Vannieuwenhuyse, *Scrum: un método ágil para sus proyectos*. Ediciones ENI., 2018.
- [117] M. Hron and N. Obwegeser, “Scrum in practice: an overview of scrum adaptations,” *Agile and Lean: Organizations, Products and Development*, 2018. [Online]. Available: <http://hdl.handle.net/10125/50568>
- [118] C. M. Novac, O. C. Novac, R. M. Sferle, M. I. Gordan, G. BUJDOSÓ, and C. M. Dindelegan, “Comparative study of some applications made in the vue.js and react.js frameworks,” in *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, 2021, pp. 1–4.
- [119] R. Baida, M. Andriienko, and M. Plechawska-Wójcik, “Performance analysis of frameworks angular and vue.js,” *Journal of Computer Sciences Institute*, vol. 14, pp. 59–64, Mar. 2020. [Online]. Available: <https://ph.pollub.pl/index.php/jcsi/article/view/1577>

-
- [120] O. C. Novac, D. E. Madar, C. M. Novac, G. Bujdosó, M. Oproescu, and T. Gal, “Comparative study of some applications made in the angular and vue.js frameworks,” in *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, 2021, pp. 1–4.
- [121] P. Pšenák and M. Tibensky, “The usage of vue js framework for web application creation,” *Mesterséges intelligencia*, vol. 2, pp. 61–72, 01 2020.
- [122] M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah, and A. rokhim, “Design an mvc model using python for flask framework development,” in *2019 International Electronics Symposium (IES)*, 2019, pp. 214–219.
- [123] P. Vogel, T. Klooster, V. Andrikopoulos, and M. Lungu, “A low-effort analytics platform for visualizing evolving flask-based python web services,” in *2017 IEEE Working Conference on Software Visualization (VISSOFT)*, 2017, pp. 109–113.
- [124] G. F. Lucio, M. P. Farrera, E. Jammeh, M. Fleury, M. J. Reed, and M. Ghanbari, “Packet by packet analysis in contemporary network simulators,” *IEEE Latin America Transactions*, vol. 4, no. 4, pp. 299–307, June 2006.
- [125] R. L. S. de Oliveira, C. M. Schweitzer, A. A. Shinoda, and Ligia Rodrigues Prete, “Using mininet for emulation and prototyping software-defined networks,” in *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, June 2014, pp. 1–6.
- [126] S. Y. Wang, “Comparison of sdn openflow network simulator and emulators: Estinet vs. mininet,” *IEEE Symposium on Computers and Communications*, p. 29, June 2014.
- [127] G. Pereira and E. Gamess, “Lineamientos para el despliegue de redes sdn/openflow,” in *Revista Venezolana de Computación*, vol. 4, no. 2, 2017, pp. 21–13.
- [128] R. Jawaharan, P. M. Mohan, T. Das, and M. Gurusamy, “Empirical evaluation of sdn controllers using mininet/wireshark and comparison with cbench,” in

- 27th International Conference on Computer Communication and Networks (ICCCN)*, 2018, pp. 1–2.
- [129] A. Yahia and E. Atwell, “Evaluation of the capabilities of wireshark as network intrusion system,” *Journal Global Research in Computer Science*, vol. 9, no. 8, pp. 1–8, 2018.
- [130] J. Zhang, J. Tang, X. Zhang, W. Ouyang, and D. Wang, “A survey of network traffic generation,” in *Third International Conference on Cyberspace Technology (CCT 2015)*, 2015, pp. 1–6.
- [131] M. Swann, J. Rose, G. Bendiab, S. Shiaeles, and N. Savage, “Tools for network traffic generation – a quantitative comparison,” 09 2021.
- [132] S. S. Kolahi, S. Narayan, D. D. Nguyen, and Y. Sunarto, “Performance monitoring of various network traffic generators,” in *2011 UkSim 13th International Conference on Computer Modelling and Simulation*, 2011, pp. 501–506.
- [133] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-Gia, “Modeling and performance evaluation of an openflow architecture,” in *2011 23rd International Teletraffic Congress (ITC)*, 2011, pp. 1–7.
- [134] M. Jarschel, F. Lehrieder, Z. Magyari, and R. Pries, “A flexible openflow-controller benchmark,” in *2012 European Workshop on Software Defined Networking*, 2012, pp. 48–53.
- [135] D. Kreutz, F. Ramos, P. Veríssimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *ArXiv e-prints*, vol. 103, 06 2014.
- [136] S. Muhizi, G. Shamshin, A. Muthanna, R. Kirichek, A. Vladyko, and A. Koucheryavy, “Analysis and performance evaluation of sdn queue model,” 2017, pp. 26–37.

UN AMBIENTE DE PRUEBA VSDN SOBRE LA PLATAFORMA TELCO 2.0



ANEXOS

Cristian Felipe Velasco Molano
Jonathan Dannilo Arcila Muñoz

Director: PhD. Oscar Mauricio Caicedo Rendón

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022

Anexos A

Historias de usuario

Las historias de usuario descritas a continuación describen la caracterización mencionada en el capítulo 4.2, sección historias de usuario.

Historia de Usuario 1	
Número: 1	Usuario: Experimentador
Nombre: Acceso al ambiente de pruebas <i>vSDN</i>	
Prioridad: Alta	Riesgo: Bajo
Estimación: 5 semanas	Iteración Asignada: 1
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede acceder al ambiente de pruebas <i>vSDN</i> mediante una aplicación <i>Web</i> alojada en un dominio de la Universidad del Cauca y gestionada por Telco 2.0.	
Validación: Acceso a la interfaz de interacción con el usuario para el uso de las herramientas ofrecidas en el ambiente de pruebas <i>vSDN</i> .	

Figura A.1: Acceso al ambiente de pruebas *vSDN*

Historia de Usuario 2	
Número: 2	Usuario: Experimentador
Nombre: Herramienta para el diseño libre de topologías vSDN	
Prioridad: Alta	Riesgo: Bajo
Estimación: 5 semanas	Iteración Asignada: 2
Responsable: Cristian Velasco - Jonathan Arcila	
Descripción: El usuario puede hacer uso de un banco de elementos de red que podrá colocar en un lienzo e interactuar entre ellos.	
Validación: Elementos de red configurables, los cuales pueden interactuar entre si en un lienzo en donde se puede crear una red válida en estructuras vSDN	

Figura A.2: Herramienta para el diseño libre

Historia de Usuario 3	
Número: 3	Usuario: Experimentador
Nombre: Herramienta para el despliegue de una red vSDN en topologías predeterminadas	
Prioridad: Alta	Riesgo: Bajo
Estimación: 4 semanas	Iteración Asignada: 3
Responsable: Cristian Velasco - Jonathan Arcila	
Descripción: El usuario puede hacer uso de un banco de topologías predeterminadas, que al seleccionarl alguna, en el lienzo, se despliega la topología deseada.	
Validación: Despliegue de cuatro topologías de vSDN: Minimal, Single, Ring y Lineal, en el lienzo de la aplicación.	

Figura A.3: Herramienta para el diseño predeterminado

Historia de Usuario 4	
Número: 4	Usuario: Experimentador
Nombre: Emulación de red	
Prioridad: Alta	Riesgo: Bajo
Estimación: 4 semanas	Iteración Asignada: 5
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede dar la orden de ejecución para la emulación de la red que se ha diseñado, con las notificaciones pertinentes de éxito o fracaso en el proceso.	
Validación: Comunicación entre la interfaz gráfica del ambiente de pruebas con el servidor, el cual estará ejecutando la emulación del despliegue de la red virtualizada.	

Figura A.4: Ejecución emulación de red

Historia de Usuario 5	
Número: 5	Usuario: Experimentador
Nombre: Detención de la emulación de red	
Prioridad: Alta	Riesgo: Media
Estimación: 2 semanas	Iteración Asignada: 6
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede detener la emulación de la red ejecutada en cualquier momento deseado.	
Validación: Detención de los sistemas virtualizados en el servidor que emulan la red ejecutada desde el cliente, y notificación al usuario del éxito del proceso.	

Figura A.5: Detención emulación de red

Historia de Usuario 6	
Número: 6	Usuario: Experimentador
Nombre: Configuración de los parámetros del tráfico	
Prioridad: Alta	Riesgo: Bajo
Estimación: 2 semanas	Iteración Asignada: 6
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario podrá acceder a una interfaz que le permita establecer los valores de los diferentes parámetros para la gestión del tráfico que desea generar en la red emulada.	
Validación: Verificación de los parámetros establecidos por el usuario para la configuración del tráfico que se envía al servidor ejecutando el motor de tráfico instalado.	

Figura A.6: Configuración de parámetros de tráfico

Historia de Usuario 7	
Número: 7	Usuario: Experimentador
Nombre: Visualización de resultados de tráfico	
Prioridad: Alta	Riesgo: Bajo
Estimación: 3 semanas	Iteración Asignada: 7
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario podrá visualizar los resultados del tráfico de la red emulada, en donde podrá observar el desempeño de la vSDN con la valoración de las métricas de desempeño.	
Validación: Despliegue de la interfaz de visualización de gráficos y valores de los resultados del tráfico de red, donde se observan características del tráfico generado y las métricas de desempeño medidas desde el servidor y el cliente, mostrando además, las métricas y gráficas de desempeño medido en toda la red en general.	

Figura A.7: Resultados de tráfico

Historia de Usuario 8	
Número: 8	Usuario: Experimentador
Nombre: Configurar controladores de red	
Prioridad: Alta	Riesgo: medio
Estimación: 4 semanas	Iteración Asignada: 8
Responsable: Cristian Velasco - Jonathan Arcila	
Descripción: El usuario puede seleccionar y configurar el controlador de red para la gestión de la vSDN a emular.	
Validación: Activación de los diferentes controladores de red y establecimiento del mismo en la configuración de la red a ser emulada.	

Figura A.8: Controladores de red

Historia de Usuario 9	
Número: 9	Usuario: Experimentador
Nombre: Métricas de desempeño individuales de la red	
Prioridad: Alta	Riesgo: Bajo
Estimación: 4 semanas	Iteración Asignada: 9
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede acceder a los resultados individuales de las métricas de desempeño del tráfico ejecutado en el elemento de red tipo <i>host</i> que este configurado en la red emulada.	
Validación: Despliegue de la información de las métricas de desempeño del tráfico de cada <i>host</i> presente en la topología de red emulada.	

Figura A.9: Métricas de desempeño individuales

Historia de Usuario 10	
Número: 10	Usuario: Experimentador
Nombre: Guardar estado de la emulación	
Prioridad: Media	Riesgo: Bajo
Estimación: 2 semanas	Iteración Asignada: 7
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede guardar y descargar la red diseñada, al igual que las gráficas y los resultados de las métricas de desempeño del tráfico ejecutado en las <i>vSDN</i> .	
Validación: Descarga de un archivo con los resultados seleccionados por el usuario.	

Figura A.10: Estado de emulación

Historia de Usuario 11	
Número: 11	Usuario: Experimentador
Nombre: Herramientas de acción para el ambiente de emulación	
Prioridad: Baja	Riesgo: Bajo
Estimación: 2 semanas	Iteración Asignada: 4
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede seleccionar diferentes herramientas de acción como lo son: deshacer, rehacer, seleccionar, borrar, copiar y pegar, en cualquier elemento o elementos de red.	
Validación: Ejecutar acción sobre el o los elementos de red según corresponda.	

Figura A.11: Herramientas de acción

Historia de Usuario 12	
Número: 12	Usuario: Experimentador
Nombre: Visualización de documentación y ayuda de la aplicación	
Prioridad: Baja	Riesgo: Bajo
Estimación: 2 semanas	Iteración Asignada: 10
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede disponer de opciones que le brinden información sobre el uso de la aplicación y temas de ayuda sobre los elementos de red, herramientas del ambiente y ejecuciones de proyectos.	
Validación: Despliegue de información y contenido según corresponda la opción seleccionada por el usuario.	

Figura A.12: Información

Historia de Usuario 13	
Número: 13	Usuario: Experimentador
Nombre: Configuración de los elementos de red	
Prioridad: Alta	Riesgo: Bajo
Estimación: 3 semanas	Iteración Asignada: 4
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede configurar los diferentes parámetros según el elemento de red que desee modificar.	
Validación: Verificación de los parámetros de configuración de los elementos de red y ejecución en la herramienta de infraestructura de red virtualizada.	

Figura A.13: Parametrización elementos de red

Historia de Usuario 13	
Número: 13	Usuario: Experimentador
Nombre: Configuración de los elementos de red	
Prioridad: Alta	Riesgo: Bajo
Estimación: 3 semanas	Iteración Asignada: 4
Responsable: Cristian Velasco - Dannilo Muñoz	
Descripción: El usuario puede configurar los diferentes parámetros según el elemento de red que desee modificar.	
Validación: Verificación de los parámetros de configuración de los elementos de red y ejecución en la herramienta de infraestructura de red virtualizada.	

Figura A.14: Parametrización elementos de red

Anexos B

Bosquejos de las interfaces de usuario

Esta sección define los bosquejos de las vistas de interacción conforme los requerimientos expresados por la literatura y la experiencia de usuario, enfocadas en permitir el correcto despliegue y función del ambiente de pruebas *vSDN*. Los bosquejos propuestos son visualizadas a continuación.

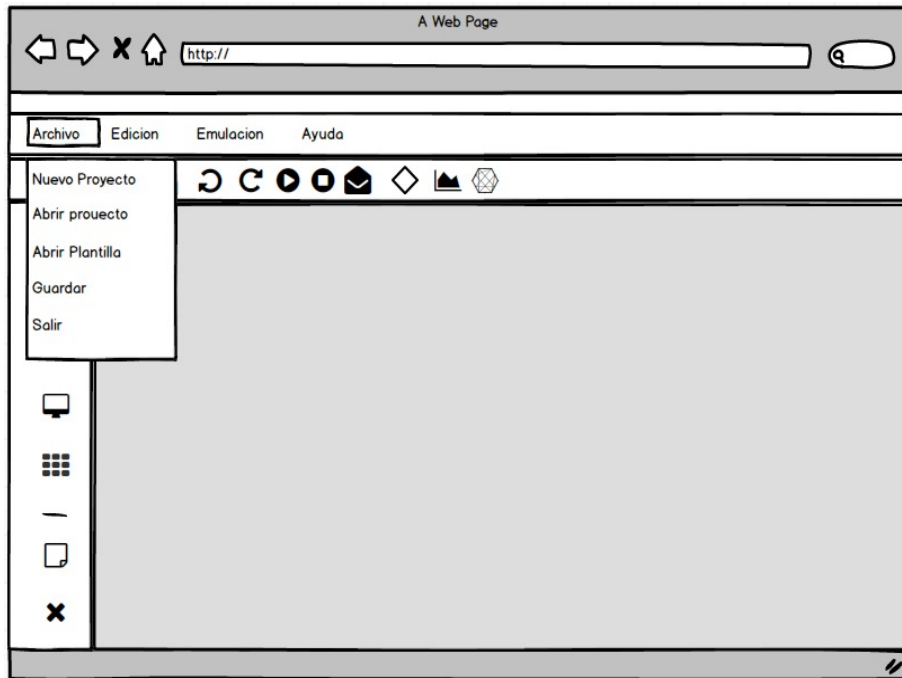


Figura B.1: Interfaz de menú de archivo

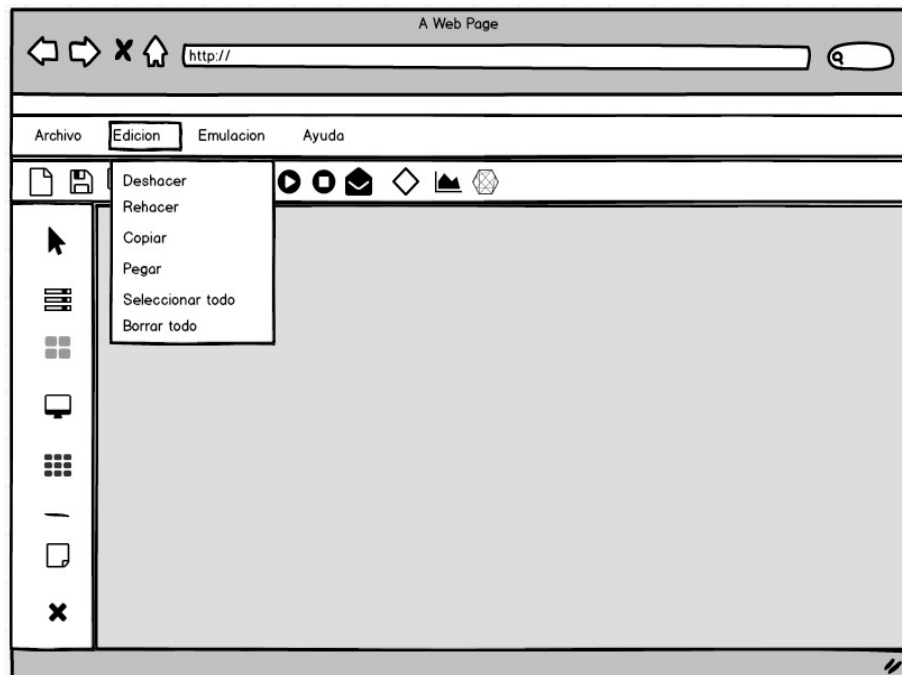


Figura B.2: Interfaz de menú de edición



Figura B.3: Interfaz de menú de emulación

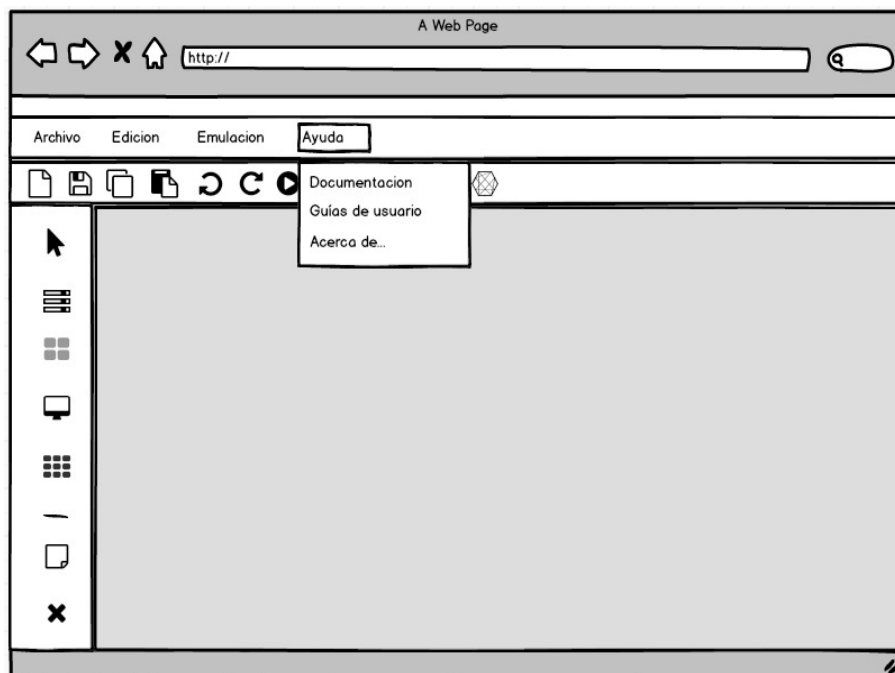


Figura B.4: Interfaz de menú de ayuda

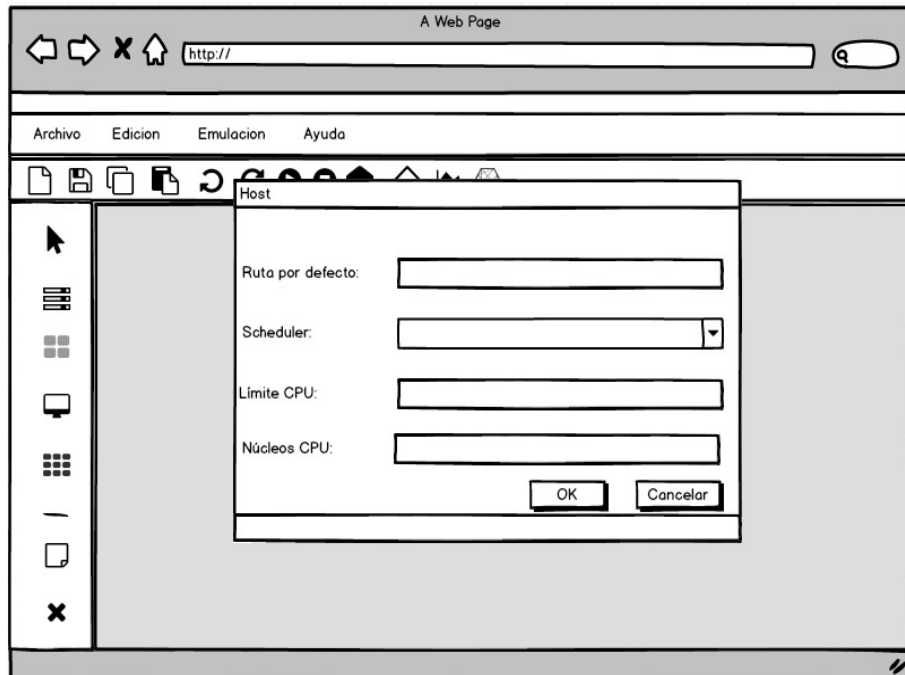
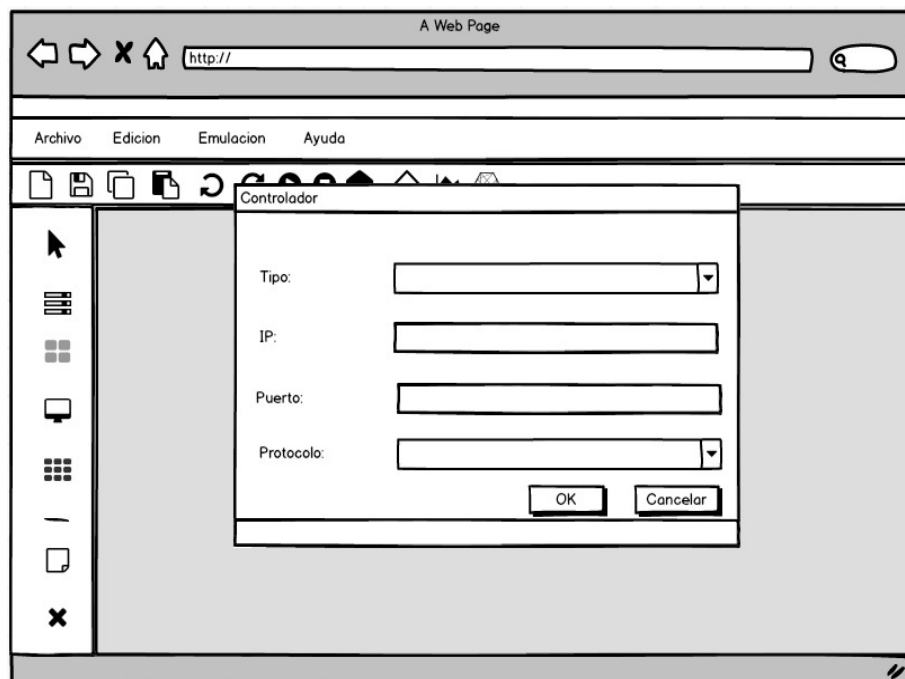
Figura B.5: Interfaz de configuración elemento *host*

Figura B.6: Interfaz de configuración elemento controlador

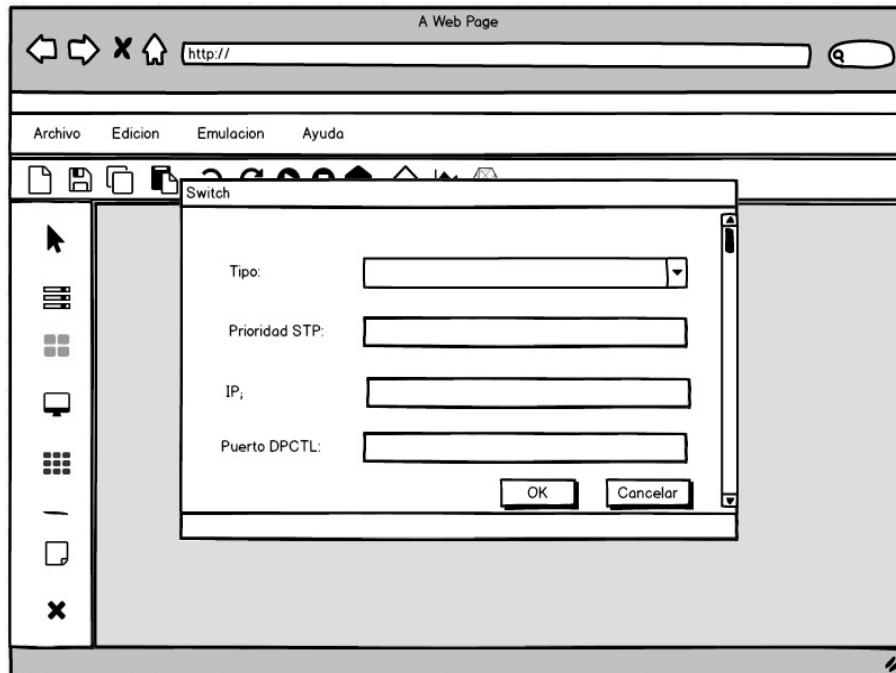


Figura B.7: Interfaz de configuración elemento conmutador *SDN*

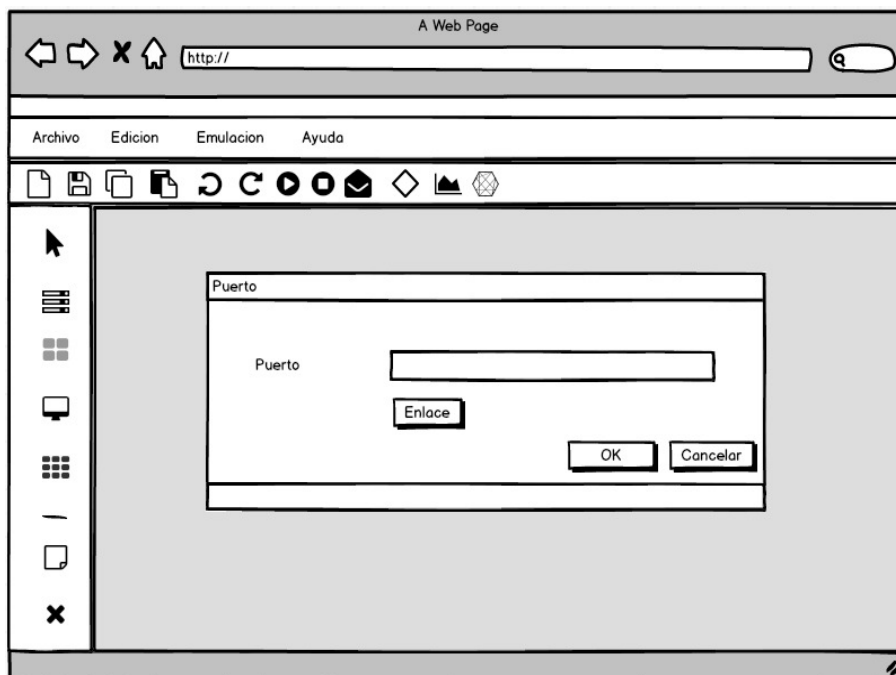


Figura B.8: Interfaz de configuración elemento puerto de red

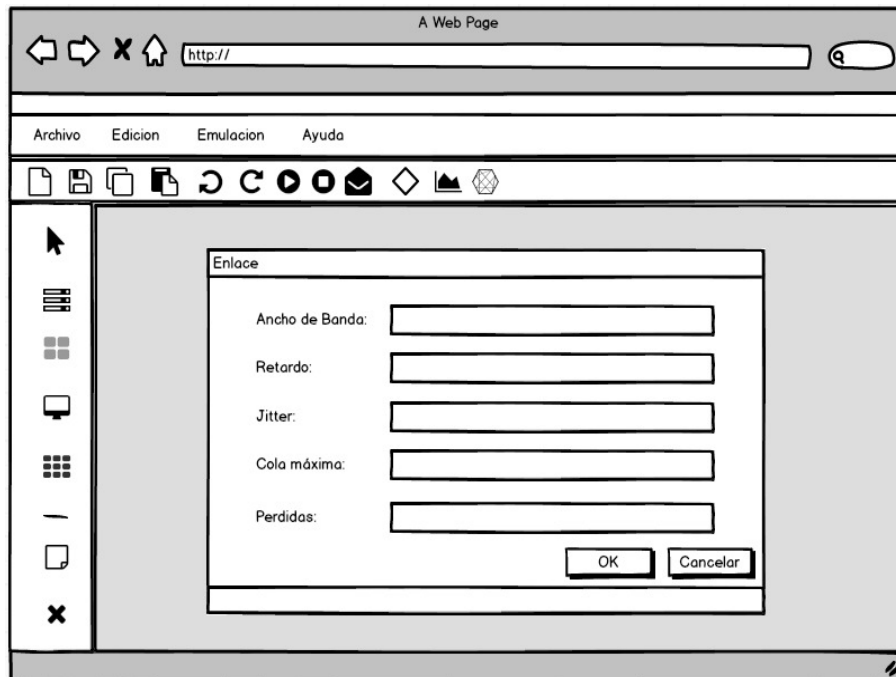


Figura B.9: Interfaz de configuración elemento enlace

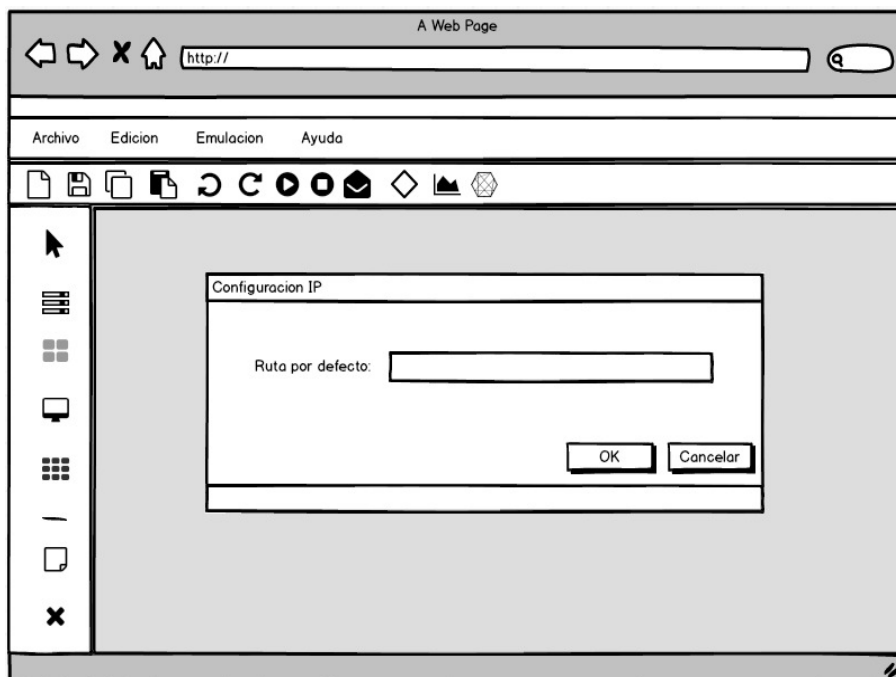


Figura B.10: Interfaz de inicio de emulación

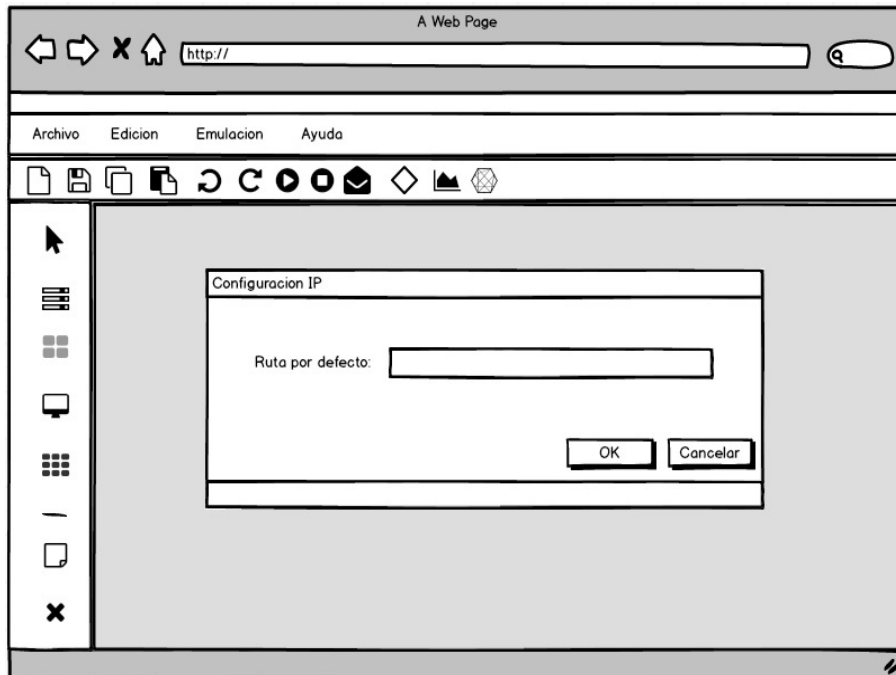


Figura B.11: Interfaz de inicio de emulación

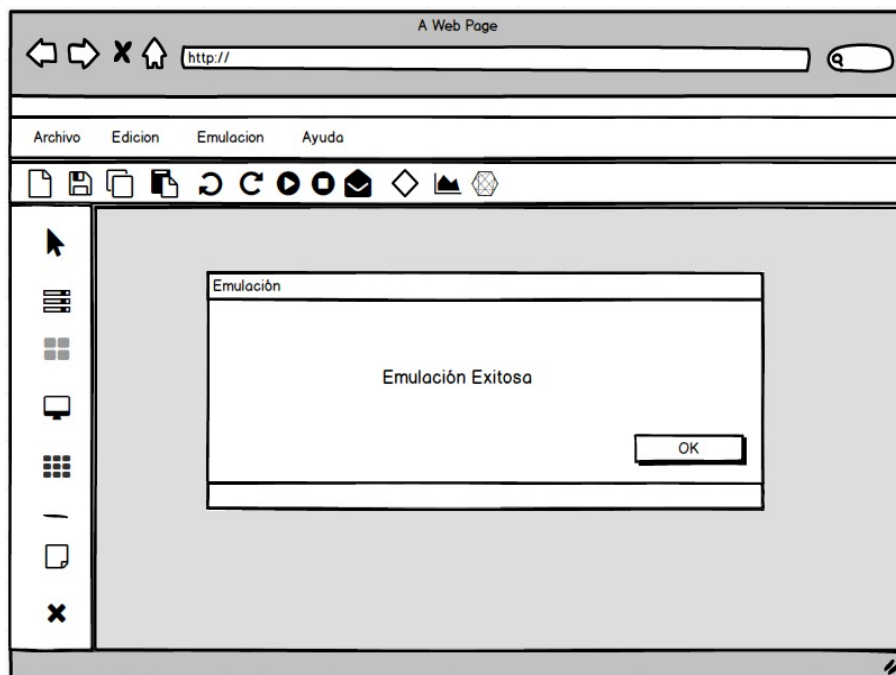


Figura B.12: Interfaz de notificaci3n de estado de la emulaci3n

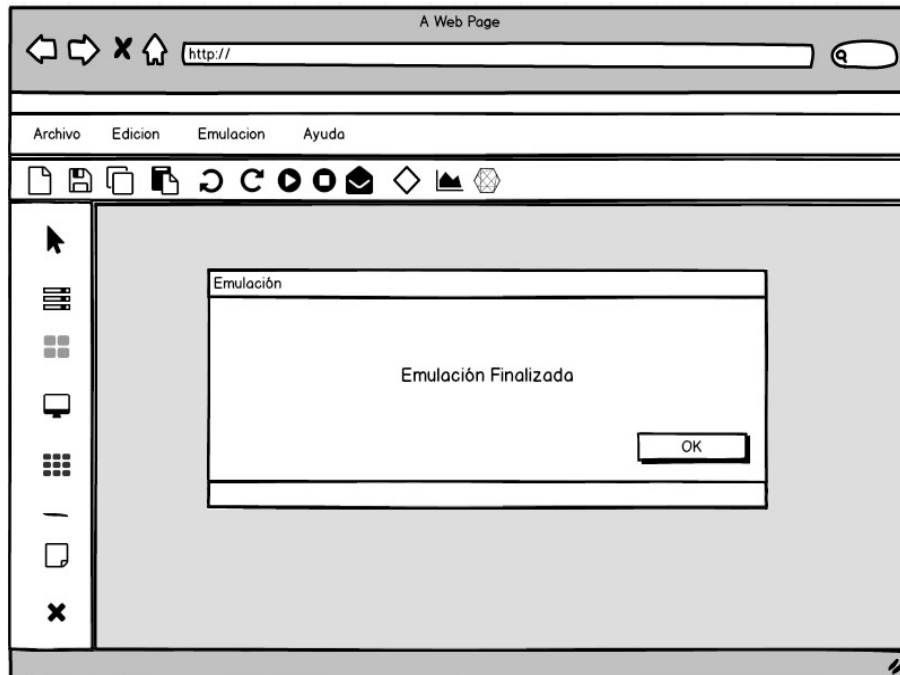


Figura B.13: Interfaz de notificación de estado de la emulación



Figura B.14: Interfaz de analizador de protocolos

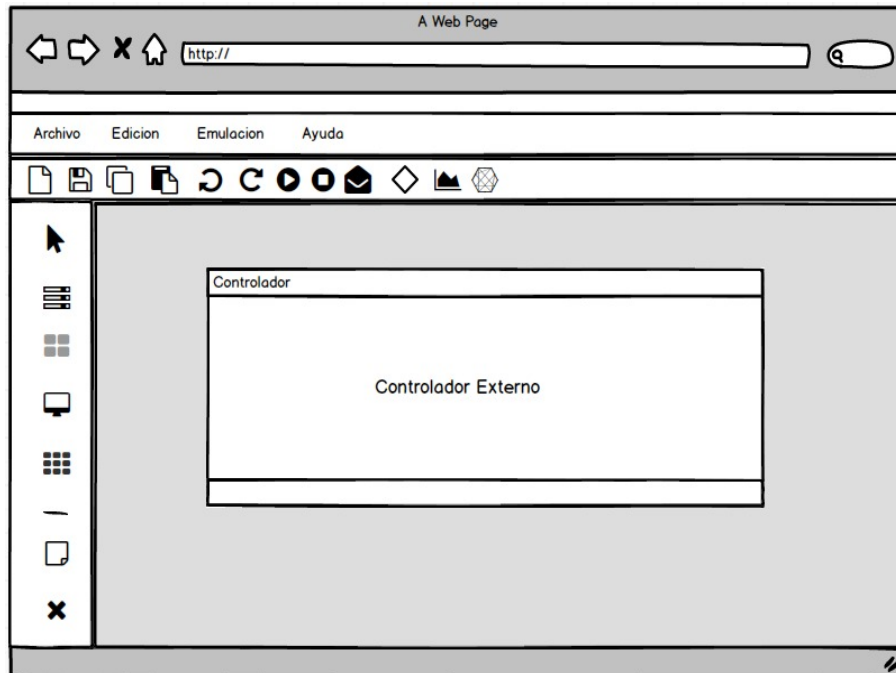


Figura B.15: Interfaz del controlador *SDN* externo

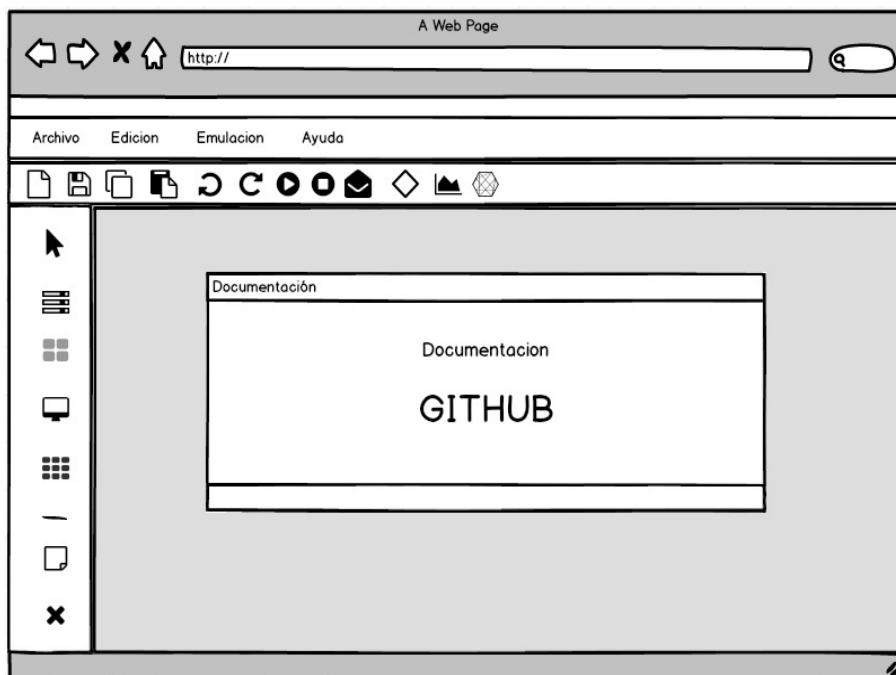


Figura B.16: Interfaz de documentación - repositorio del ambiente

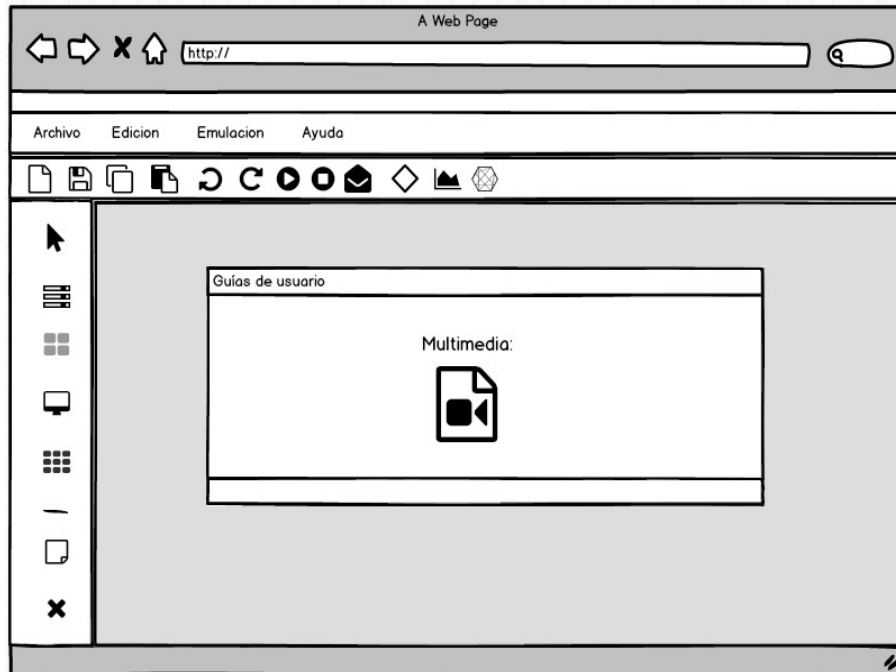


Figura B.17: Interfaz de documentación - guía de usuario del ambiente



Figura B.18: Interfaz de documentación - descripción del ambiente

Anexos C

Interfaces Gráficas de Usuario

Esta sección define las vistas de interacción aceptadas por el cliente conforme al correcto despliegue y función del ambiente de pruebas *vSDN*. Las *GUI* propuestas son visualizadas a continuación.

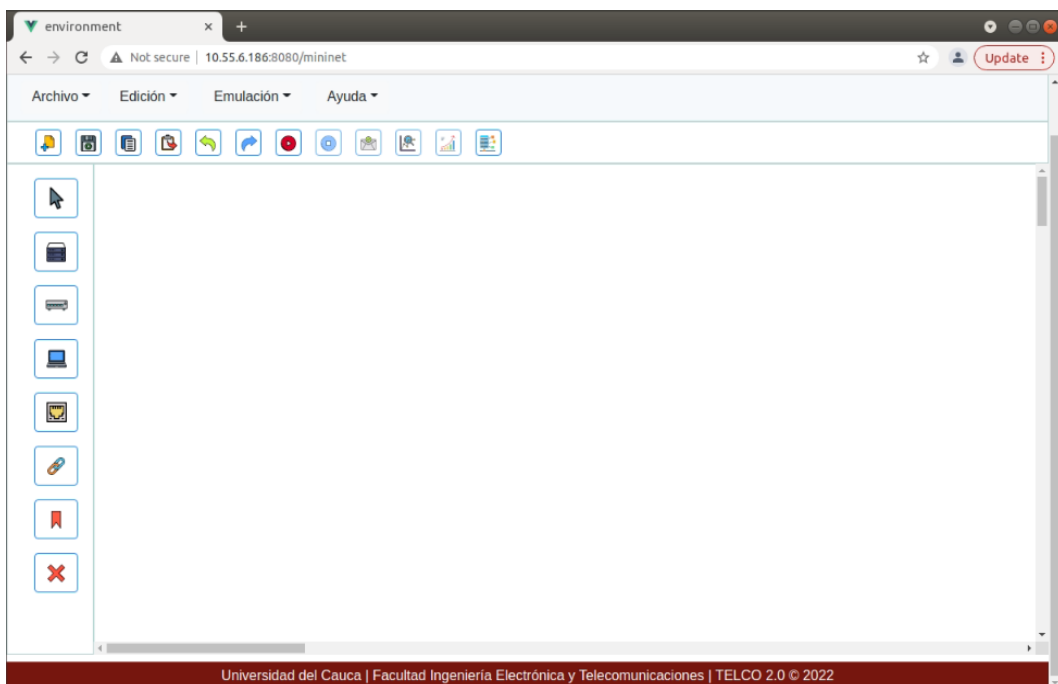


Figura C.1: Interfaz de inicio del ambiente de pruebas *vSDN*

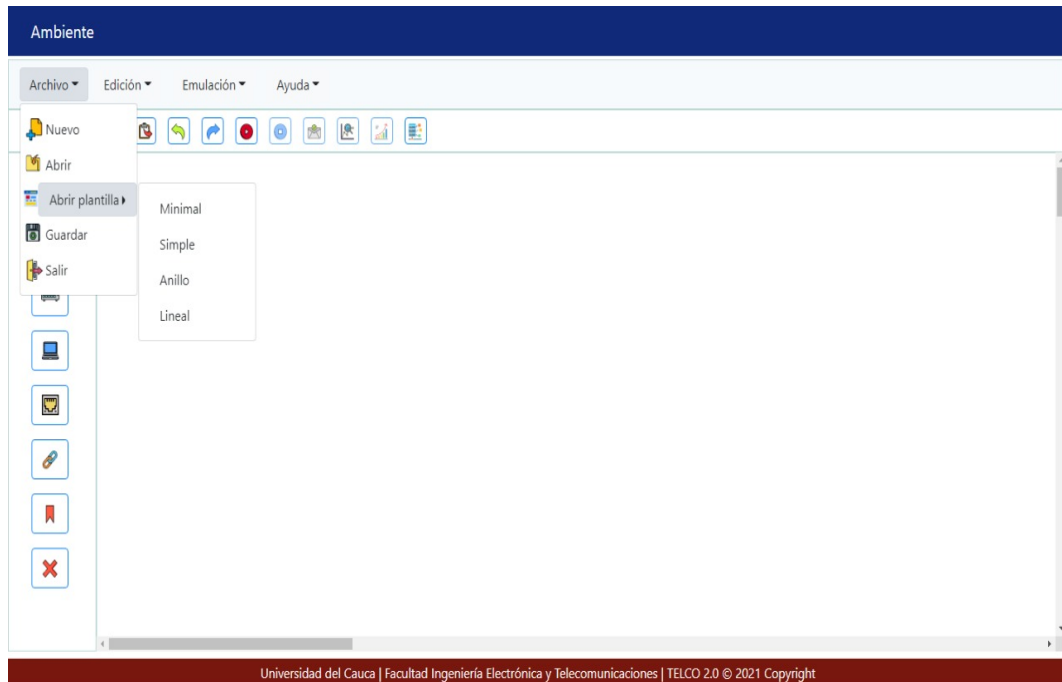


Figura C.2: Interfaz de selección de topologías *vSDN* prediseñadas.

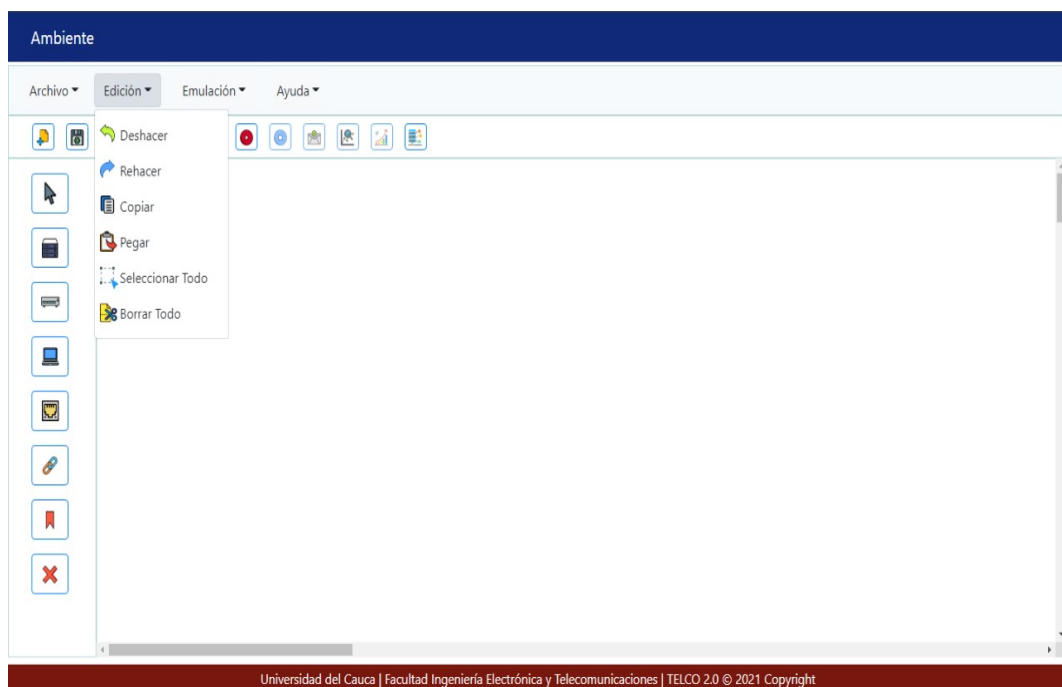


Figura C.3: Interfaz de menú de edición.

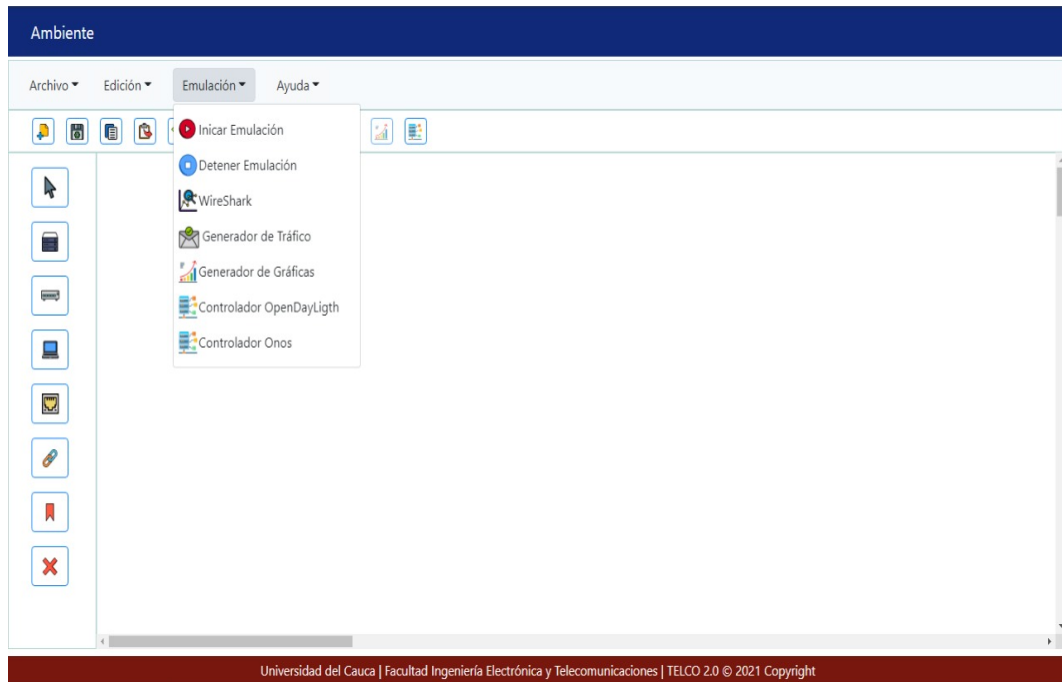


Figura C.4: Interfaz de menú de emulación.

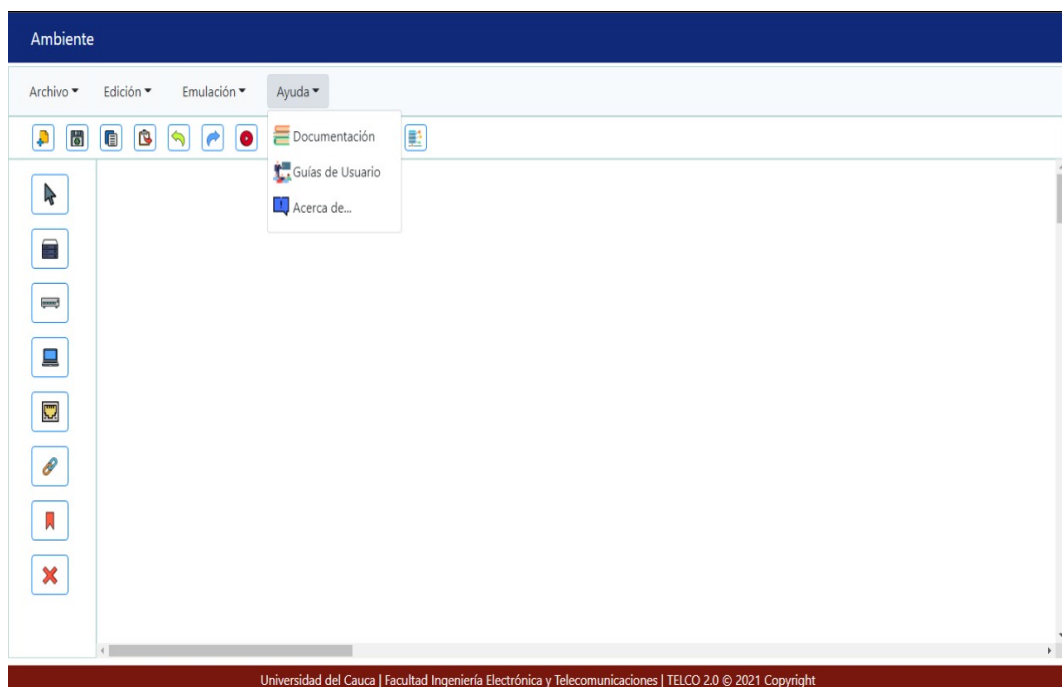


Figura C.5: Interfaz de menú de ayuda.

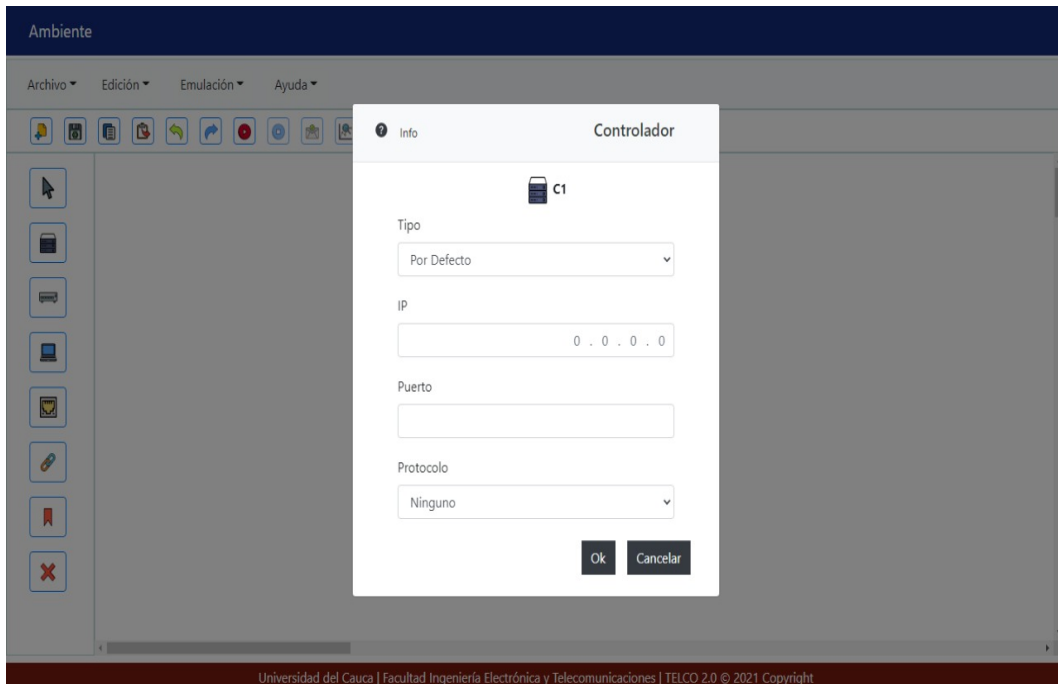


Figura C.6: Interfaz menú de configuración del controlador.

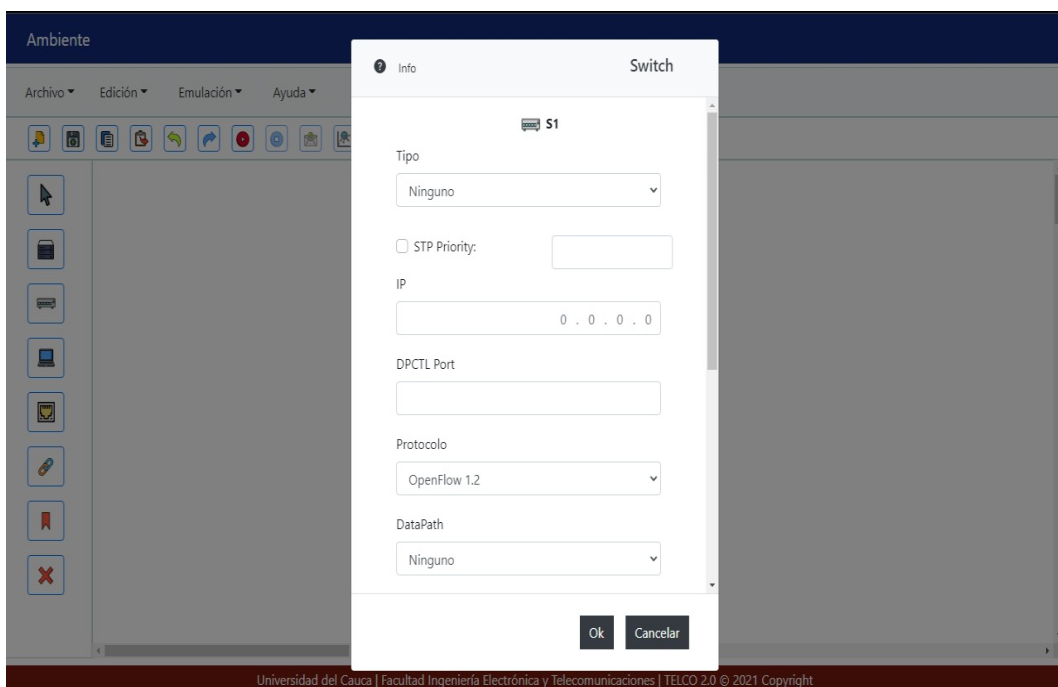


Figura C.7: Interfaz menú de configuración del *switch*

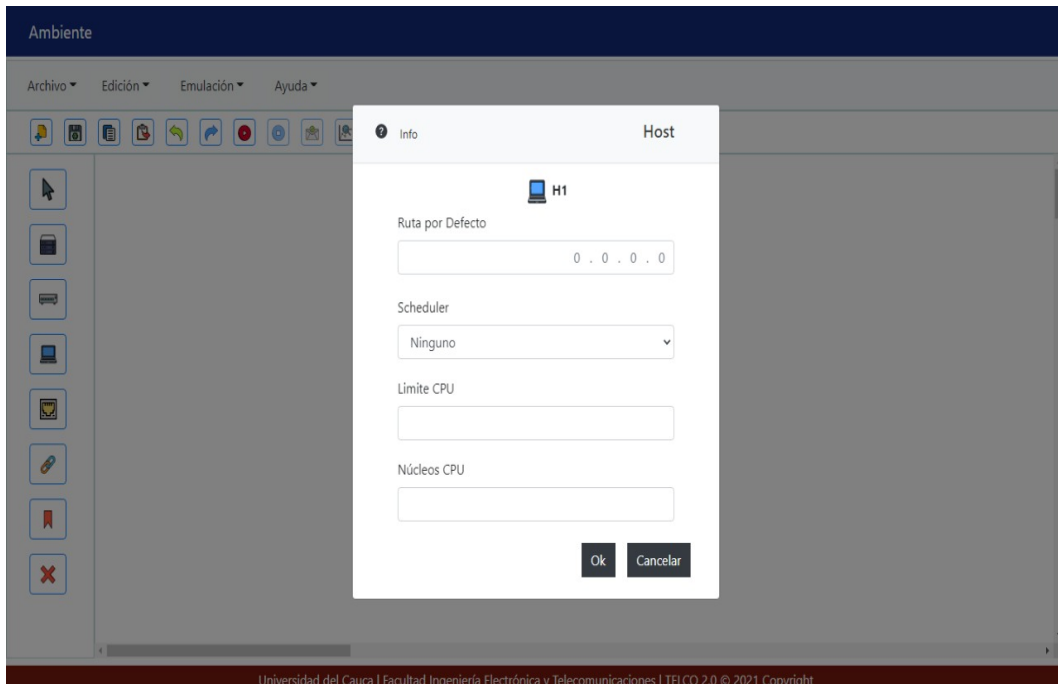


Figura C.8: Interfaz menú de configuración del *host*

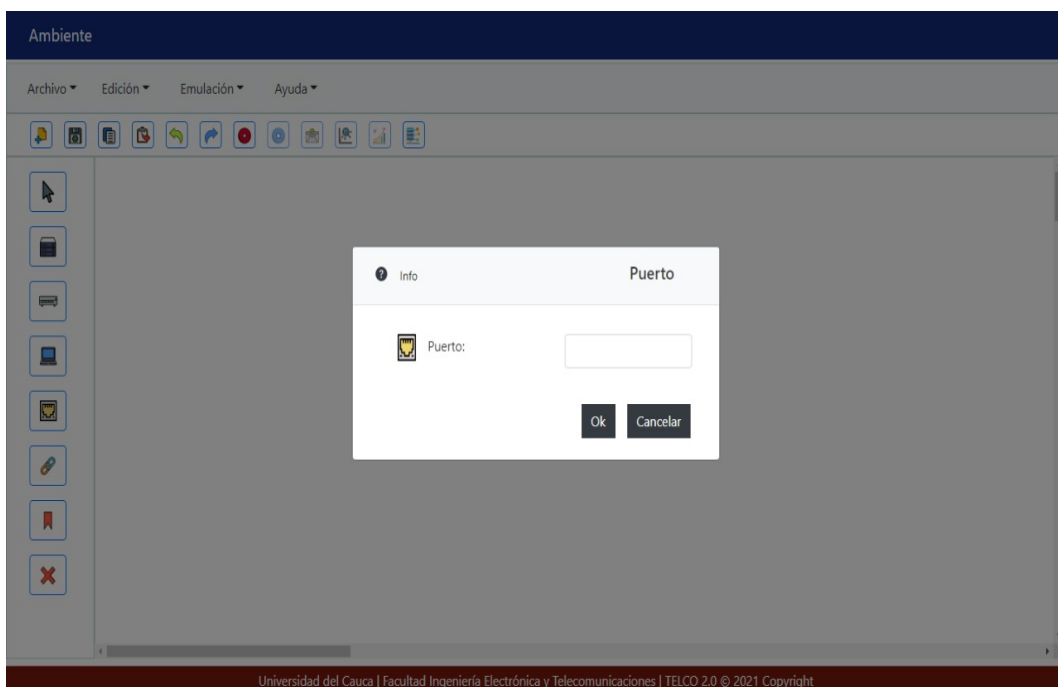


Figura C.9: Interfaz menú de configuración del puerto

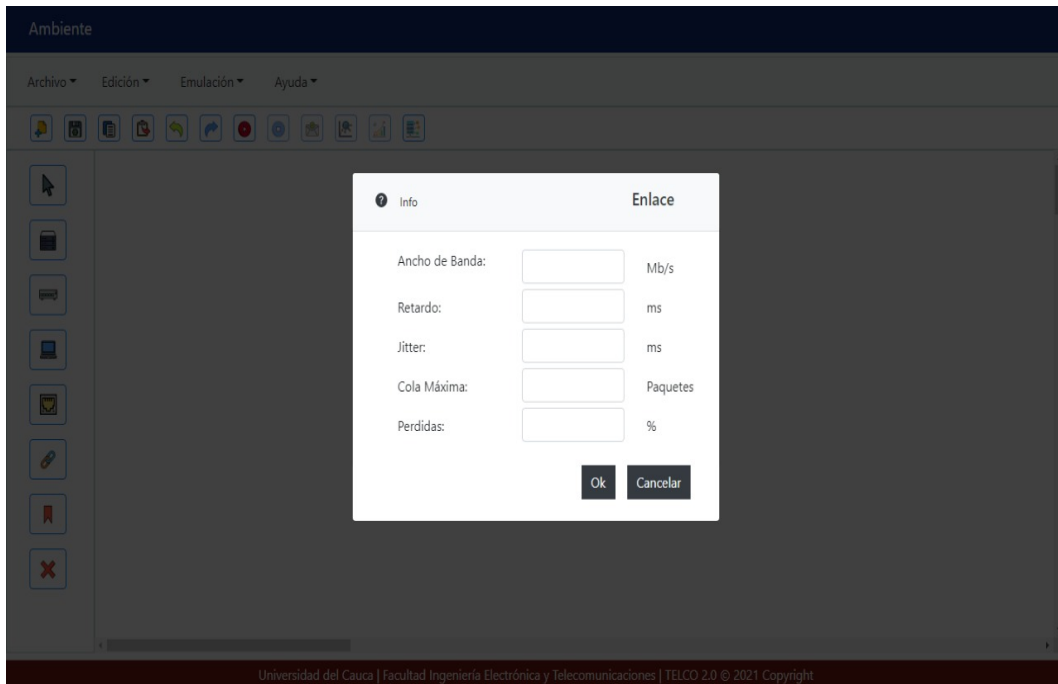


Figura C.10: Interfaz menú de configuración del enlace

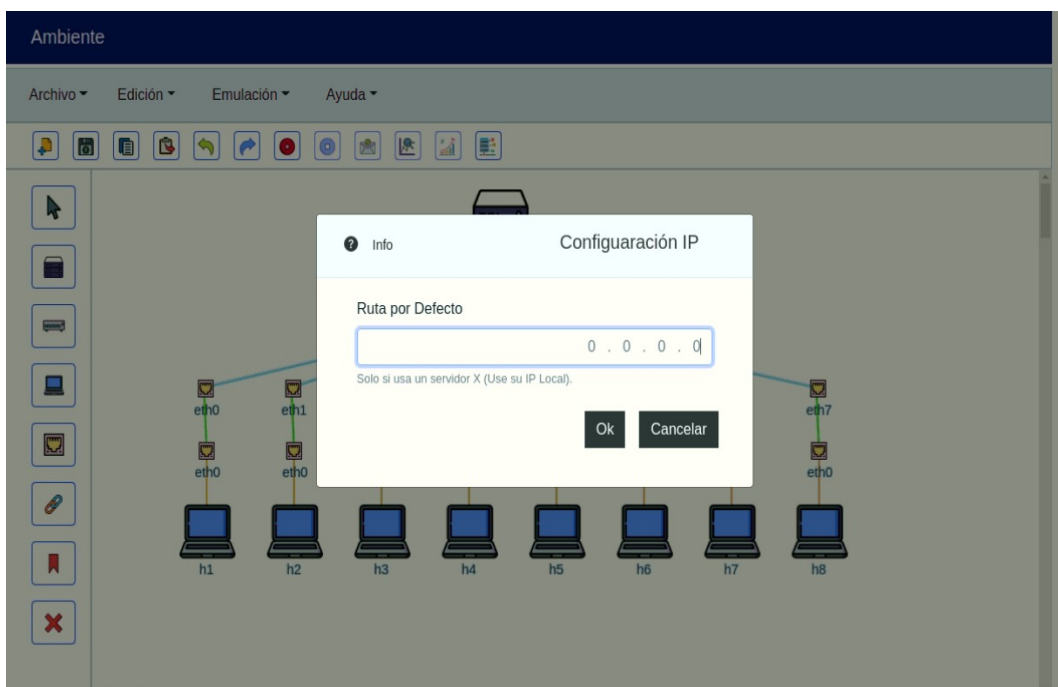


Figura C.11: Interfaz menú de configuración IP servidor gráfico del cliente

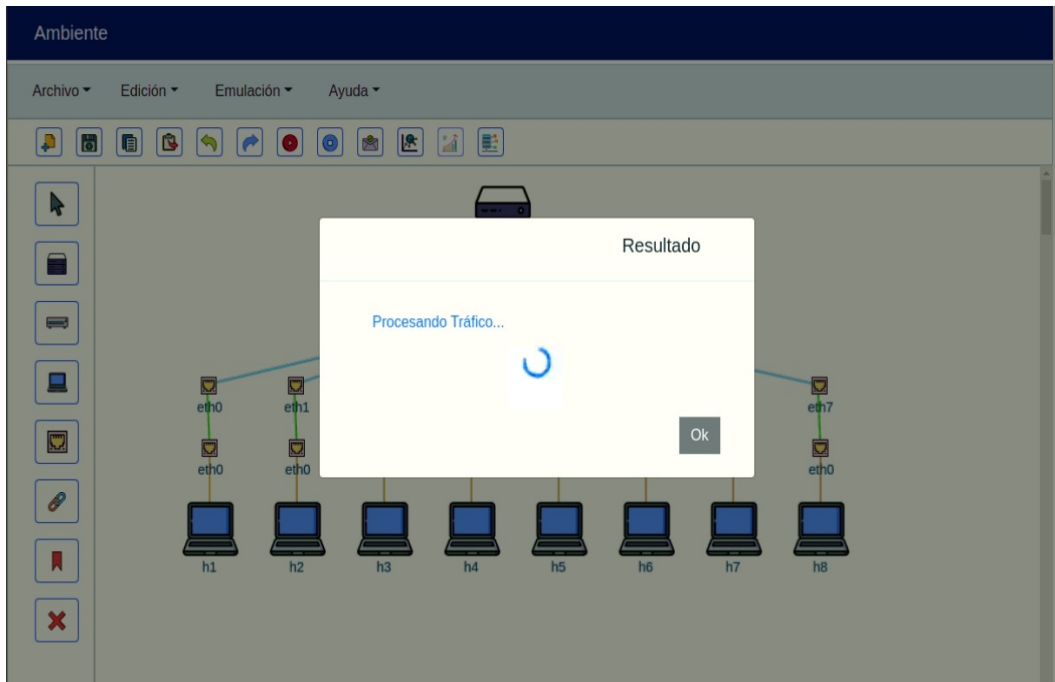


Figura C.12: Interfaz de notificación del estado de la emulación

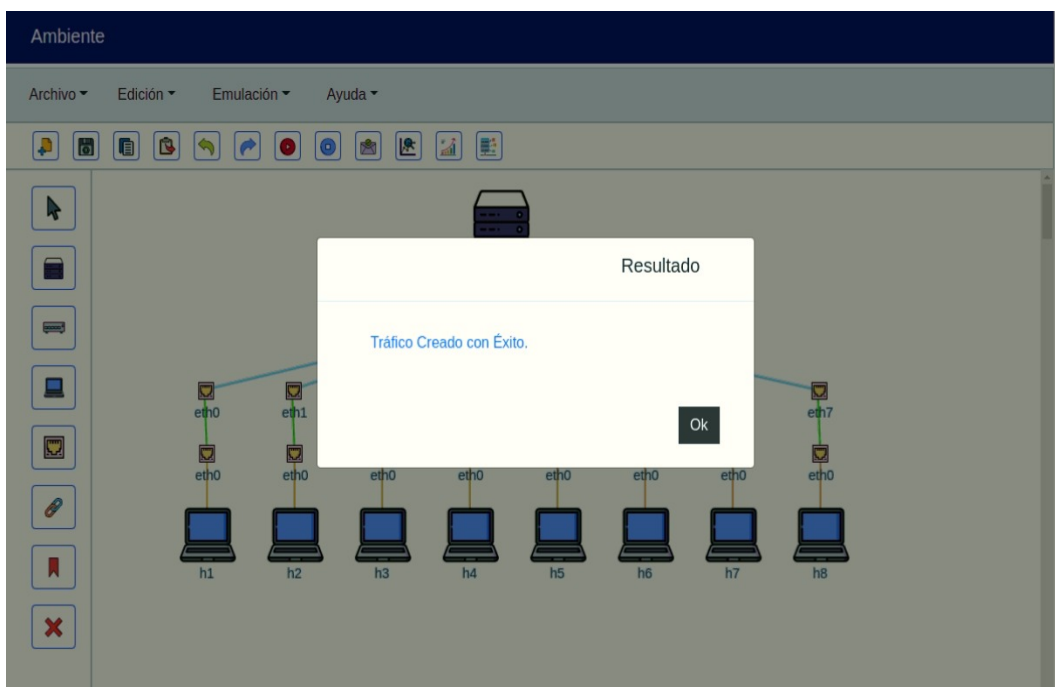


Figura C.13: Interfaz de notificación del estado de la emulación

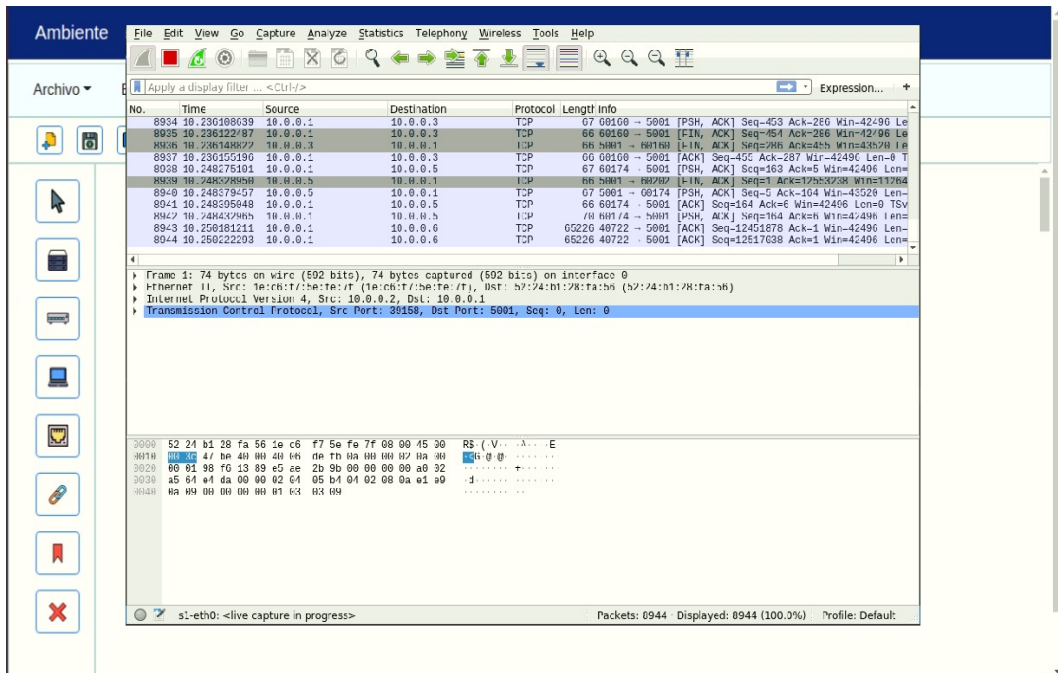


Figura C.14: Interfaz del analizador de protocolos

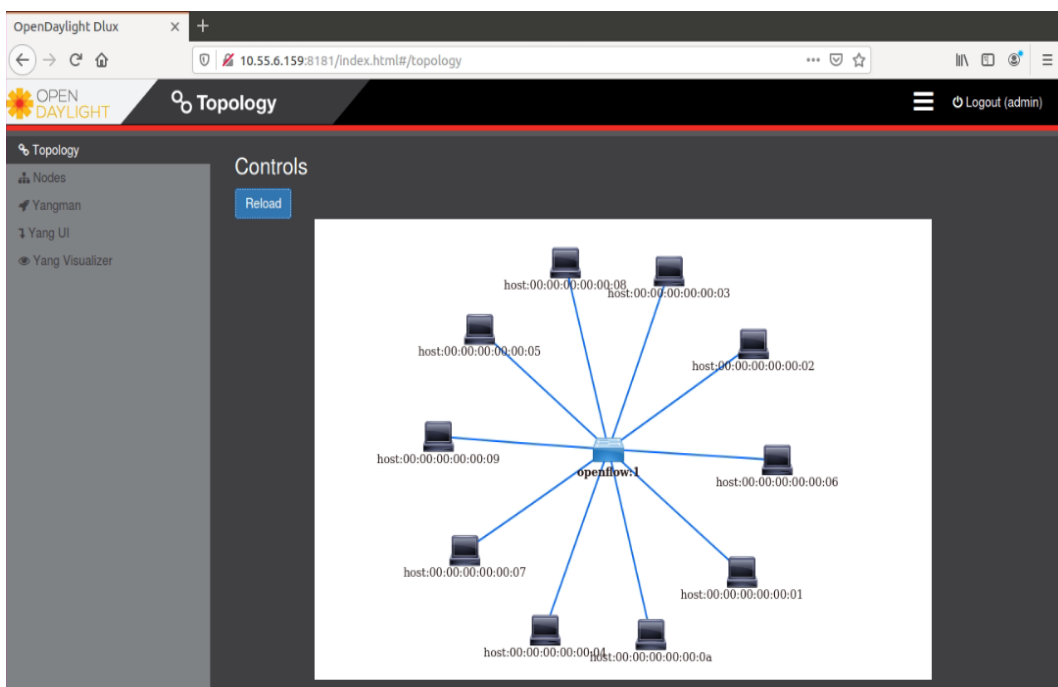


Figura C.15: Interfaz gráfica del controlador externo ODL

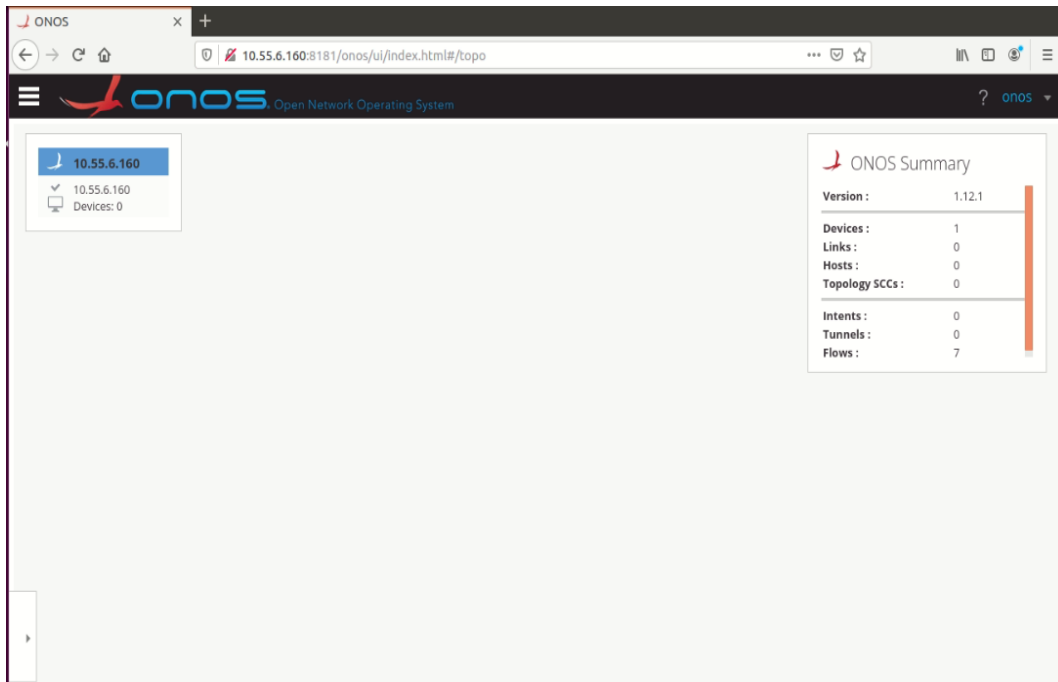


Figura C.16: Interfaz gráfica del controlador externo *ONOS*

Anexos D

Repositorio Telco IMS

Este anexo contiene el enlace donde está almacenado el código fuente de la *GUI* del ambiente de pruebas *vSDN* sobre Telco 2.0. El repositorio ha sido almacenado en GitHub y es accesible mediante la URL:

`https://github.com/cristianvelascofm/emulation_environment`

Este anexo contiene el enlace donde está almacenado el código fuente del servidor *Web* del ambiente de pruebas *vSDN* sobre Telco 2.0. El repositorio está alojado almacenado en GitHub y es accesible mediante la URL:

`https://github.com/cristianvelascofm/serverMininetFlask`

Anexos E

Guías de usuario

Las guías de usuario realizadas para ayudar al experimentador con el uso y configuración del ambiente de pruebas *vSDN* han sido desarrolladas en diferentes archivos multimedia de video que facilitan la descripción y funcionamiento del entorno. En la figura E.1 es presentado el título y la descripción de los archivos multimedia que describen las guías de usuario del ambiente.

Título	Descripción
Introducción al Ambiente	Describe en general el ambiente de pruebas <i>vSDN</i> y las herramientas que permite desplegar
Herramientas y menús elementos de red	Describe el menú de configuración de cada elemento de red desplegado en el lienzo de la aplicación
Diseño Libre	Describe el proceso de despliegue de los diferentes elementos de red para el diseño de una topología en el modo que quiera diseñarla el usuario
Topologías predeterminadas	Describe la selección y desplie de los diferentes diseños de topologías generadas por el ambiente para su posterior emulación
Generador de tráfico y resultados	Describe la ventana de configuración de los parámetros de gestión del tráfico que se generará en una red emulada, además, ejemplifica los modos y vistas de los resultados en la interfaz de métricas de desempeño del emulador
Controlador OpenDayLight	Describe el acceso a la configuración predeterminada del controlador externo OpenDayLight
Controlador Onos	Describe el acceso a la configuración predeterminada del controlador externo Onos

Figura E.1: Guías de usuario multimedia del ambiente de pruebas *vSDN*