UNIVERSIDAD ECOTEC

TÍTULO:

# DISEÑO DE UN MODELO DE RED DEFINIDA POR SOFTWARE PARA LA VIRTUALIZACIÓN A TRAVÉS DEL CONTROLADOR FLOODLIGHT EN LA EMPRESA ELÉCTRICO HAZ S.A.

LÍNEA DE INVESTIGACIÓN:

TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

MODALIDAD:

PROPUESTA TECNOLÓGICA

CARRERA:

INGENIERÍA EN SISTEMAS

TÍTULO POR OBTENER:

INGENIERA EN SISTEMAS CON ÉNFASIS EN ADMINISTRACIÓN DE REDES

AUTORA:

LINDA KARINA CHALÉN VÉLEZ

TUTOR:

MSC. MANUEL RAMÍREZ

GUAYAQUIL 2021

## DEDICATORIA

Dedico esta tesis con todo mi corazón a mi madre Aracely que es mi soporte más grande, es quien lo ha dado todo por mí, jamás ha dudado de mis capacidades y se ha esforzado tanto dándome su apoyo incondicionalmente para poder lograr este objetivo.

A mis abuelitos Cledia y Rubén porque con su amor y su manera de cuidarme he aprendido lo más hermoso de la vida.

A Cristina que ha seguido cada paso conmigo en esta etapa de mi vida.

A Juliana que es mi más grande motivación para todo.

A Layita que desde que llegó a mi vida soy más feliz.

## AGRADECIMIENTO

Agradezco a mi madre Aracely porque nunca dejo que me rindiera, me apoyó con todas sus fuerzas y es por ella principalmente que puedo cumplir esta meta.

A mis abuelitos Cledia y Rubén por darme tanto amor incondicional y siempre estar pendientes de mí en la realización de este proyecto.

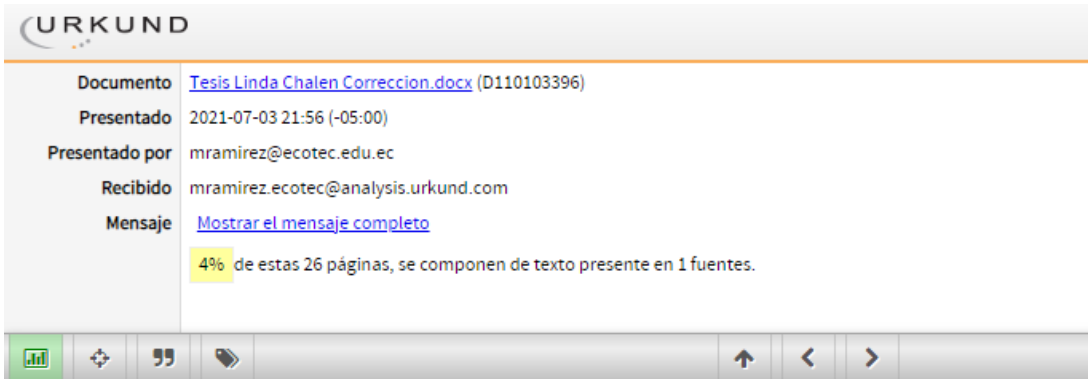
A Juliana por creer en mí, no dejarme caer y acompañarme incondicionalmente en todo este proceso.

## CERTIFICADO DEL PORCENTAJE DE COINCIDENCIAS

Habiendo sido nombrado MANUEL RAMÍREZ, tutor del trabajo de titulación DISEÑO DE UN MODELO DE RED DEFINIDA POR SOFTWARE PARA LA VIRTUALIZACIÓN A TRAVÉS DEL CONTROLADOR FLOODLIGHT EN LA EMPRESA ELÉCTRICO HAZ S.A. elaborado por LINDA KARINA CHALÉN VÉLEZ**,** con mi respectiva supervisión como requerimiento parcial para la obtención del título de INGENIERA EN SISTEMAS CON ÉNFASIS EN ADMINISTRACIÓN DE REDES.

Se informa que el mismo ha resultado tener un porcentaje de coincidencias (4%) mismo que se puede verificar en el siguiente link: https://secure.urkund.com/view/104957713-775619-833182. Adicional se adjunta print de pantalla de dicho resultado.

**Nombres y Apellidos del Tutor: Msc. Manuel Ramírez.**




**MSC. MANUEL RAMÍREZ**

# CERTIFICACION DE REVISION FINAL

QUE EL PRESENTE PROYECTO DE PROPUESTA TECNOLÓGICA TITULADO:

**DISEÑO DE UN MODELO DE RED DEFINIDA POR SOFTWARE PARA LA VIRTUALIZACIÓN A TRAVÉS DEL CONTROLADOR FLOODLIGHT EN LA EMPRESA ELÉCTRICO HAZ S.A.**

ACOGIÓ E INCORPORÓ TODAS LAS OBSERVACIONES REALIZADAS POR LOS MIEMBROS DEL TRIBUNAL ASIGNADO Y CUMPLE CON LA CALIDAD EXIGIDA PARA UN TRABAJO DE TITULACIÓN, POR LO QUE SE AUTORIZA A: **(LINDA KARINA CHALÉN VÉLEZ),** QUE PROCEDA A SU PRESENTACION.

**Samborondón, 03-07-2021**

**Nombres y Apellidos del Tutor: Msc. Manuel Ramírez.**

## RESUMEN

El siguiente proyecto se basó en una propuesta tecnológica para la empresa Eléctrico Haz S.A. Esta propuesta está compuesta por cuatro capítulos. El primer capítulo contiene la descripción de las redes definidas por software o SDN, con sus características, beneficios, diferencias de una SDN con una red de arquitectura tradicional y sus protocolos con todas las especificaciones necesarias, así mismo los controladores y softwares de virtualización existentes compatibles para su uso. En el capítulo dos, se encuentra el desarrollo del proyecto, en donde se explica paso a paso la instalación del software de virtualización y su respectivo controlador. En el capítulo tres, se encuentra el diseño de la red que fue escogido de acuerdo a las necesidades e intereses de la compañía antes mencionada, se muestran los pasos que se siguieron para la creación de la red con las funciones de cada comando utilizado. El capítulo cuatro indica los resultados obtenidos de las pruebas hechas a manera de simulación. De acuerdo a estos análisis, los resultados fueron satisfactorios y cumplieron las expectativas esperadas.



**Palabras clave:** Controlador, Floodlight, MiniNet, SDN, redes tradicionales, simulación, virtualización, banda ancha, seguridad de red, centralización, software, openflow, protocolos, control de acceso.

## ABSTRACT

The following project was based on a technological proposal for the company Eléctrico Haz S.A. This proposal is made up of four chapters. The first chapter contains the description of software-defined networks or SDN, with its characteristics, benefits, differences of an SDN with a traditional architecture network and its protocols with all the necessary specifications, as well as the existing compatible virtualization drivers and software. for your use. In chapter two, you will find the development of the project, where the installation of the virtualization software and its respective driver is explained step by step. In chapter three, the design of the network that was chosen according to the needs and interests of the aforementioned company is found, the steps that were followed for the creation of the network are shown with the functions of each command used. Chapter four indicates the results obtained from the tests carried out in a simulation manner. According to these analyzes, the results were satisfactory and fulfilled the expected expectations.



**Keywords:** Controller, Floodlight, MiniNet, SDN, traditional networks, simulation, virtualization, broadband, network security, centralization, software, OpenFlow, protocols, access control.

# ÍNDICE

## ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

**TEMA**

DISEÑO DE UN MODELO DE RED DEFINIDA POR SOFTWARE PARA LA VIRTUALIZACIÓN A TRAVÉS DEL CONTROLADOR FLOODLIGHT EN LA EMPRESA ELÉCTRICO HAZ S.A

# 1. INTRODUCCIÓN

El reciente proyecto se basa en una propuesta tecnológica para la empresa Eléctrico HAZ S.A. La misma busca garantizar sus niveles óptimos en sus equipos y mantenerlos actualizados, ya que actualmente solo utiliza una red de datos con arquitectura tradicional la cual presenta retrasos, fallas de seguridad y altos costos debido a que han renovado sus equipos informáticos varias veces, pero sin obtener el resultado esperado.

Es por este motivo se requiere analizar y realizar un estudio detallado para brindar una propuesta que permitirá establecer un nivel de última tecnología aumentando la calidad y eficiencia en el cumplimiento de la misión asignada.

La empresa Eléctrico Haz S.A. empezó su funcionamiento en el año 2001 se dedica a la venta de suministros y materiales eléctricos para el sector del hogar y construcción, está ubicada en la parte central de la ciudad de Guayaquil y cuenta con una matriz conformada por cinco áreas, entre las cuales están el área contable, ventas, gerencia, sistemas y bodega.

## 1.1. ANTECEDENTES

En los últimos años el incremento de las redes de datos se ha desarrollado de una forma exponencial, de la misma manera han migrado a nuevas formas en capacidad y acumulación de datos que tienen lugar en la red, así mismo como las nuevas tecnologías que se han incorporado. Es a partir de ello que surge el concepto de las redes definidas por software o SDN. (Chafloque Mejía, 2018).

Las SDN son una perspectiva de edificación de una red que permite a la misma ser supervisada o monitoreada de forma ingeniosa y centralizada empleando un software. Esto contribuye a que se administre la red de forma certera y totalmente separada de alguna red que ya exista. Como beneficios o ventajas que otorgan las SDN al usuario, se pueden mencionar que son más flexibles, escalables, brindan un ahorro de costos a largo plazo, emplea la automatización,

incrementa la seguridad y cuentan con una administración total y sencilla. (Bernal & Mejía, 2016).

Por un lado, los ambientes en los que se pueden aplicar las SDN son para los propósitos de incremento de las funciones de la red, enrutamiento de paquetes controlados por aplicaciones, administración de dispositivos, calidad de servicio, definición y distribución total de políticas de seguridad. Además, entre algunos de los tipos de análisis que se pueden realizar están, el internet de las cosas o IOT, virtualización y base de datos. (López, 2019).

En conclusión, los resultados de los diferentes análisis de las categorías mencionadas y los aspectos de los trabajos investigados se propone hacer un modelo de red de datos definidas por software con un controlador para virtualización.

## 1.2. PLANTEAMIENTO DEL PROBLEMA

Las redes que no son definidas por software no pueden responder a patrones de tráfico impredecibles y tampoco a los picos de demanda. El aumento de tráfico en las redes recientes está impulsando a optimizar el funcionamiento de las redes actuales. La cantidad de datos han estado aumentando en un gran volumen y esto es a causa del tráfico multimedia, usualmente se tiende a hacer un uso intensivo del mismo, el cual demanda un ancho de banda más amplio para que pueda trabajar de una manera acertada. (Osaba, 2016).

A lo largo de las recientes décadas, pese a las gigantescas inversiones en búsqueda y costos en la adquisición de equipos, forman en la actualidad un impedimento al ingreso de nuevas maneras de tecnología para optimizar recursos, lo que resulta una barrera en la agilidad de desarrollo, la misma que hoy en día no es suficiente para llenar la demanda y necesidad de los usuarios y transformación del mercado, obteniéndose así dependencia de los involucrados. (Marín Muro, 2016).

La empresa Eléctrico Haz S.A. ha presentado continuamente problemas en su red, al momento utilizan una red local con arquitectura tradicional, esto ha ocasionado retrasos y que no se den abasto con el ancho de banda. Así mismo en su momento por decisión del área de gerencia hicieron una actualización de todos sus equipos lo cual les representó un costo muy alto y el inconveniente no fue solucionado del todo, adicional las autoridades de esta compañía en vista de que no dio resultado la decisión tomada, empezaron la búsqueda de un sistema que le permita tener su información con extrema seguridad sin que le represente un costo alto.

Estos son algunos de los problemas relacionados con el diseño y la arquitectura de su red, los mismos que se utilizarán de base para demostrar mediante un análisis las mejoras que tiene un modelo novedoso de red como lo son las SDN. Por medio del controlador OpenFlow Floodlight que funciona con switches, routers, switches virtuales y puntos de acceso, se brinda módulos que son más sencillos y con muchas ventajas simplificando el trabajo al usuario final. Este controlador es capaz de ejecutar varios procesos a la vez, es de alto rendimiento y puede operar con redes mixtas OpenFlow y no OpenFlow.

En definitiva, los resultados que se obtendrán con el uso de las SDN son ahorros de costos, optimización de recursos, seguridad de la información y una administración más sencilla que servirán como propuesta de implementación en la empresa Eléctrico Haz S.A.

## 1.3. OBJETIVOS

### 1.3.1. OBJETIVO GENERAL

Diseñar un modelo de red definida por software para la virtualización a través del controlador Floodlight en la empresa Eléctrico Haz S.A.

### 1.3.2. OBJETIVOS ESPECÍFICOS

- Determinar los aspectos relacionados al diseño de redes definidas por software para la virtualización.
- Analizar los tipos de software de virtualización existentes para el desarrollo de las SDN.
- Investigar los tipos de controladores utilizados para virtualizar una red definida por software.
- Proponer una red definida mediante el controlador Floodlight.

## 1.4. JUSTIFICACIÓN

Es necesario el desarrollo de una SDN debido que, a diferencia de una red común, las SDN brindan un crecimiento en optimización de recursos, no representa costos altos, ofrece flexibilidad, son escalables, permiten la utilización de la automatización, incrementa la seguridad y cuentan con una administración sencilla.

La virtualización de redes posibilita adaptar los requerimientos que la misma necesite a esta, basándose en la ejecución de un software. Las funciones de red pueden ejecutarse de manera rápida cuando el usuario lo desee y lo requiera, evitando hacer tediosas configuraciones.

Este novedoso modelo de red resume el trabajo del usuario, no consume mucho tiempo y tampoco cantidades excesivas de dinero, facilita el manejo de la información y por lo consiguiente este modelo resulta ideal para la empresa Eléctrico Haz S.A. que uno basado en una red no definida por software.

## 1.5 ALCANCE

El alcance de este trabajo es proponer a la empresa Eléctrico Haz S.A. un diseño de red definida por software que se acople a sus necesidades, es por esta razón que se sugiere utilizar el método de virtualización usando el controlador Floodlight ya que, este sería ideal para el mejorar el funcionamiento de la empresa.

Los métodos que se usarán en este trabajo son los siguientes:

**Descriptivo:** Dado que la información que será incorporada a este trabajo será mediante conceptos y definiciones ya previamente investigadas, de esta manera se obtendrá una base confiable de lo que se expone y plantea dentro de este trabajo.

**Exploratorio:** Puesto que se necesita ahondar un poco más en las problemáticas que se presentan y por lo consecuente llevar a cabo un análisis más detallado sugiriendo una propuesta de solución.

# MARCO TEÓRICO

# CAPÍTULO I

# 2. CAPÍTULO I. MARCO TEÓRICO

En el presente capítulo se abordarán los conceptos teóricos relacionados a las redes definidas por software o SDN; así como, un análisis de los principales proyectos de investigación propuestos por la literatura científica con el fin de establecer una propuesta que permita responder a los fines de este trabajo.

## 2.1. REDES CON ARQUITECTURA TRADICIONAL

Las redes con arquitectura tradicional, son aquellas que se basan en una dirección IP. Estas redes son las más comunes hoy en día y por esa razón se puede decir que son las más implementadas a nivel corporativo.

El mayor inconveniente que tiene esta red con arquitectura tradicional es su administración, ya que resulta más complejo para el administrador su implementación debido a sus configuraciones manuales y su integración de forma vertical. Otros inconvenientes que presenta esta arquitectura con red tradicional son los costos de implementación y mantenimiento, la vulnerabilidad de la información y el tráfico en procesos de transferencia de datos.

Las redes con arquitectura tradicional tienen como característica principal que cada uno de los dispositivos se controlan a sí mismos, es decir que cada uno de los equipos que se encuentren que se encuentren conectados toman sus propias decisiones y cuentan con su propio firmware que se encuentra instalado en el espacio que este tiene disponible en la memoria. Por lo consiguiente, para poder hacer cualquier modificación que represente un cambio como por ejemplo en la topología, las reglas o los protocolos, el administrador de la red tiene el deber de configurar de forma manual a cada uno de los dispositivos disponibles.

Esto aumenta en gran medida la complejidad de su función y da más posibilidad a que se produzcan errores inesperados en su ejecución. Como resultado de todas estas causas que se encuentran en las redes con arquitectura tradicional, se puede decir que estas arquitecturas de red están evolucionando a pasos

agigantados hacia ciertas topologías que son más dinámicas y a su vez más programables como lo son las SDN. (La Salle, 2020).

## 2.2. REDES DEFINIDAS POR SOFTWARE O SDN

Las redes aparecieron a raíz de la llegada de los ordenadores, las primeras redes permitían que los usuarios que estén en cualquier parte del mundo tuvieran la facilidad de poder comunicarse intercambiando datos, como lo son archivos, textos, fotos, etc. También les permitió el acceso a todos los recursos que estarían compartidos en una red. En la actualidad las redes han evolucionado tanto que ya no solo se tiene el acceso para enviar archivos y compartir recursos ahora existe un modelo de red centralizada que se encargaría de toda la administración de una manera más sencilla como lo son las redes definidas por software o SDN.

Las SDN en pocas palabras son inteligentes, centralizadas y programadas, es por esa razón que las organizaciones cada vez más buscan monitorizar y controlar toda la topología de una red desde un mismo punto, fusionando aplicaciones desde la programación. De esta manera nace el paradigma introducido por las redes definidas por software o SDN, que supone una irrupción en el mundo de las telecomunicaciones, pues pretende evitar o suprimir la necesidad de configurar cada equipo por separado.

Estas redes funcionan con un switch, en este existen dos partes diferenciadas, el plano de datos y el plano de control, el plano de datos corresponde a la parte del hardware, las tramas llegan al switch por cierto puerto y éste lo reenvía o distribuye por uno o varios puertos, este nivel de circuitos lo conforma el plano de datos. El plano de control es el cerebro del switch, es el que va a tomar esas decisiones para que esas tramas salgan por el puerto asignado, es decir, donde se decidirán y ejecutarán las decisiones de los protocolos que tengan implementados. Las SDN lo que propone es separar ambos planos, lo que hace que en el switch no se implemente el plano de control.

En definitiva, ya no se definirían los protocolos, sino que todas las reglas ya vendrían implementadas en un controlador y ese mismo controlador será el que mande las instrucciones al switch para que este pueda tomar las decisiones necesarias según el tráfico de datos que se presente, por lo tanto, no tiene que ser el switch quien implemente todos esos protocolos y en su lugar lo que se necesita es que este pueda entender el lenguaje por el que el controlador le va a hablar y le va a enviar las instrucciones que en este caso son las pertinentes a ejecutar. (Rivoir & Morales, 2019).

## 2.3. DIFERENCIAS ENTRE LAS SDN Y NO SDN

La característica principal que diferencia a las SDN con las redes que no son SDN, es la separación del software y el hardware. Hace varios años esta separación era inimaginable, pero a partir de esa idea se comenzó a desarrollar en gran cantidad y variedad algunos dispositivos que son capaces de hacerlo.

Las redes no SDN o con arquitectura tradicional era hasta esa época lo más novedoso y cumplía su función de una manera aceptable, pero desde la llegada de las SDN, la tecnología empezó a tomar un camino diferente. Un poco más de una década ha pasado desde que este tipo de tecnología centralizada comenzó a desarrollarse, esto es muy favorable ya que poco a poco las empresas y organizaciones están empezando a notar más interés por este tipo de tecnología para aplicarla.

Existen grandes empresas que ya apostaron por un cambio en su diseño y arquitectura de redes, y al momento ya cuentan con este tipo de tecnología en sus organizaciones, entre estas empresas están Google, IBM, Microsoft, Cisco, Tech Mahindra. (Marín Muro, 2016).

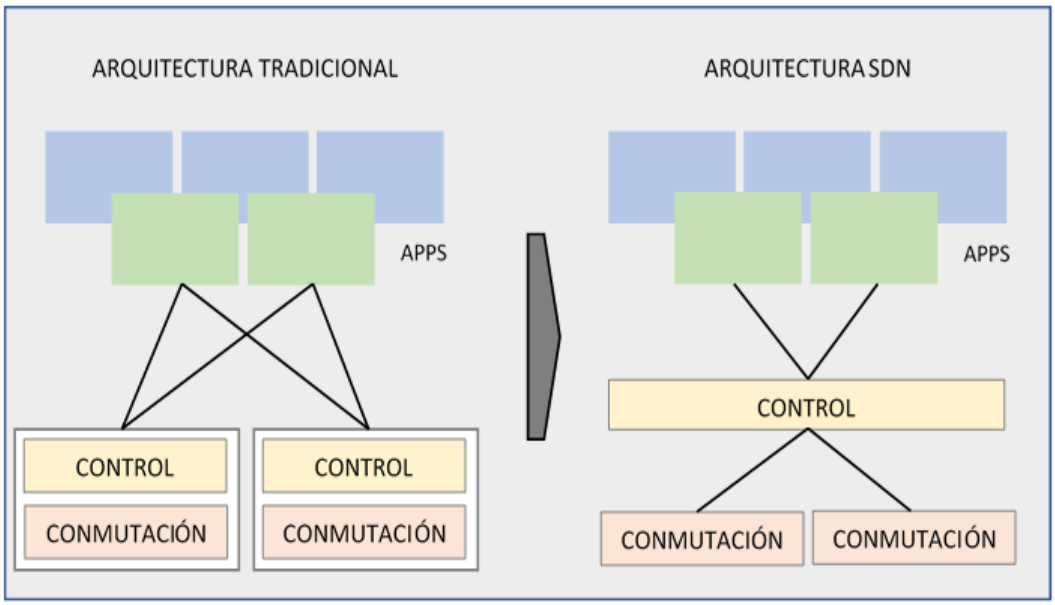
Como resultado de todos los cambios que ha tenido la tecnología en las últimas décadas se puede decir que las SDN han tenido un mayor impacto en organizaciones más grandes por su atractivo en configuraciones centralizadas, lo cual les representa menos costo de implementación, y mantenimiento.

Adicional a esto, las SDN también han sido de gran interés como objeto de estudio y de desarrollo por otros sectores de la industria.

La arquitectura de estos dos tipos de redes es totalmente distinta, a pesar de cumplir una misma función, se puede notar que la estructura funciona de forma diferente entre los dos tipos de redes.

En la siguiente figura se puede visualizar su manejo y distribución.

**FIGURA 2.1.** Arquitectura de las redes tradicionales y SDN.




**Nota:** La red con arquitectura tradicional está conectada dividiendo el plano de control, por lo tanto, las configuraciones tendrán que ser individuales, mientras que la arquitectura SDN está conectada de una manera más centralizada. Elaboración propia.

**TABLA 2.1.** Redes SDN y no SDN.

| Redes SDN | Redes no SDN |
|---|---|
| Administración optimizada, rápida y dinámica. | Su administración demanda mayor tiempo y es estática. |
| Es escalable. | No es escalable. |
| Sus tablas contienen apertura. | No tienen apertura las tablas de flujo. |
| Operatividad centralizada. | Operatividad manual |
| Compatibilidad con dispositivos y software. | Limitaciones de compatibilidad en dispositivos y software. |
| Usa automatización. | No usa automatización. |
| Bajos costos de operación. | Altos costos de operación. |
| Accesibilidad por medio del hardware al plano de datos. | Accesibilidad por medio del software al plano de datos. |

**Nota:** En esta tabla se muestran las diferencias existentes entre las redes que no son definidas por software y las que sí son definidas por software. Elaboración propia.

## 2.4. CARACTERÍSTICAS DE LAS SDN

Las SDN tienen una **infraestructura programable**, es decir la posibilidad que ofrecen hoy en día los dispositivos y los distintos sistemas de programar y aplicar código programable a la infraestructura completa. Sin las SDN la infraestructura de red común se caracteriza por ser bastante rígida ya que los equipos vienen con un sistema operativo que por lo general son propietarios. Con SDN se ofrece

la posibilidad de programar mediante lenguajes de programación la infraestructura completa de red.

Son **administrados centralizadamente**, con las SDN se puede tener un sistema informático, una aplicación o un software que gestione de manera centralizada cada parámetro de la infraestructura de red, lo cual permite obtener una infraestructura optimizada, esto es sumamente importante porque en el enfoque común se complica la administración de todos los dispositivos a medida que la red va creciendo de tamaño ya que se debe ingresar manualmente cada uno de los dispositivos y aplicar configuración de VLAN o cambiar protocolos de enrutamiento, en cambio con una infraestructura centralizada se puede lograr obtener una **optimización dinámica y rápida.**

**La automatización** en redes quiere decir que después de programar de manera correcta y la cual funcione sola cuando hay un evento, puede responder de manera inteligente a las tareas pre programadas que se han habilitado, lo que con una red tradicional podría demorar varias horas de trabajo. Al automatizar con SDN existe ahorro de tiempo para la administración porque se ejecuta en segundo plano.

Las SDN están basadas en código abierto, **open source** como comúnmente se lo conoce, esto es una ventaja favorable ya que permite la neutralidad del fabricante, es decir que no se atan al mismo, sino que pueden interactuar con distintas marcas de dispositivos para que sean compatibles. (Sulca, 2018).

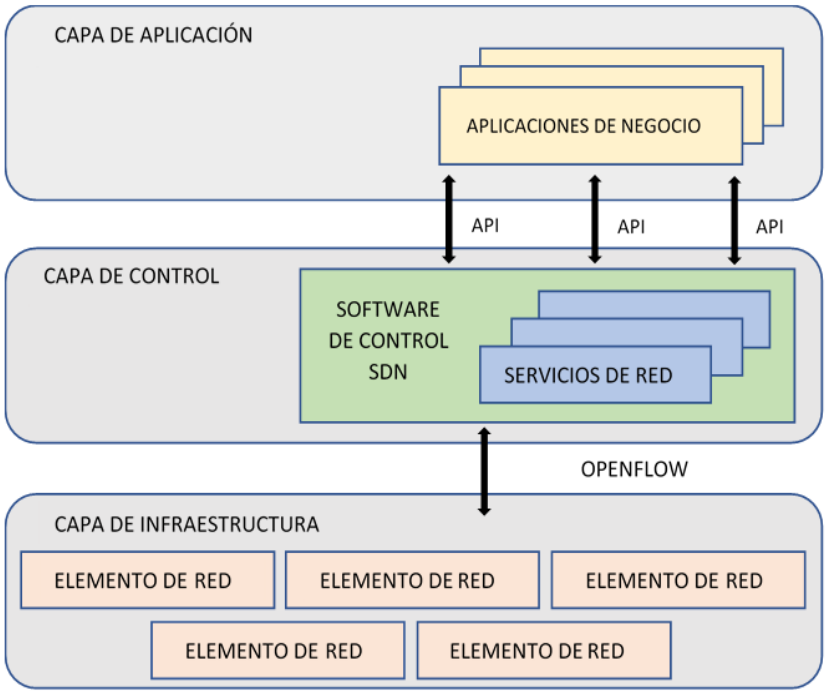
## 2.5. COMPONENTES DE UNA SDN

Según Marín (2016) las redes definidas por software o SDN se componen de la siguiente manera:

- Capa de infraestructura
- Capa de control
- Capa de aplicación

En la siguiente figura se muestra la distribución de estos elementos con sus capas.

**FIGURA 2.2.** Componentes de una red SDN.




**Nota:** En esta figura se puede visualizar la composición de las SDN mediante sus tres capas con sus elementos y cómo están conectadas mediante un controlador Openflow. Elaboración propia.

Para comprender mejor cómo actúan estas tres capas Marín (2016) menciona los siguientes conceptos de cada una de ellas:

**Capa de infraestructura:** De forma semejante a una red con arquitectura tradicional, la capa de infraestructura se conforma por un grupo de dispositivos, en los cuales se puede mencionar conmutadores y enrutadores que se encuentran conectados entre ellos, lo que da por resultado una red robusta. La distinción primordial radica en que los dispositivos habituales son componentes que se especializan en el reenvío de paquetes y no dispone de un control o software inmerso para la toma de decisiones de manera independiente. De esta manera el funcionamiento de la red ya no se encuentra en los dispositivos de

datos, se traslada a uno centralizado al que se denomina sistema operativo de red.

**Capa de control:** Es un organismo natural que tiene el cometido de transformar las peticiones de las diversas aplicaciones SDN en órdenes para los dispositivos que se encuentran en las capas debajo de esta, por lo que provee a las aplicaciones SDN una percepción abstracta de la red. Esta capa es la encargada de la toma de decisiones en cuanto a cómo los paquetes tendrán que ser distribuidos por uno o más componentes de red, así mismo de la programación de los nodos de la red para la implementación de las decisiones ya tomadas. Posee la función de conservar de manera actualizada las tablas de distribución de los dispositivos del plano de datos o infraestructura apoyado en la topología de la red o peticiones de servicios externos.

**Capa de aplicación:** Permite establecer aplicaciones que se encargan de automatizar las funciones de la configuración, provisión y despliegue de los nuevos servicios de red que se presenten. Las aplicaciones que se encuentran en esta capa podrían ser denominadas “el cerebro de la red” ya que por medio de estas se implementa la lógica de control que se convierten en órdenes que a su vez son remitidas al plano de datos para establecer el comportamiento y manejo de los nodos especializados en la distribución de los paquetes de datos. Las aplicaciones pueden determinar la dirección por la que los paquetes van a trasladarse desde un punto a otro dentro de la red. (Marín Muro, 2016).

## 2.6. SDN SIMÉTRICA Y ASIMÉTRICA

A pesar de que los principios en los que se basan la mayoría de los softwares son para brindar la mejor administración posible de la inteligencia de los dispositivos de red, existen de igual manera algunos procedimientos para llevar a cabo una SDN donde las funciones que son manejadas por el plano de control se distribuyen en diferentes unidades de control.

En un diseño asimétrico de estas características, los diferentes sistemas usualmente disponen de una pequeña cantidad de información para que pueda

funcionar correctamente, por este motivo si llega a fallar la unidad de control los mismos logran seguir con sus operaciones. En este diseño es más probable que las redundancias de datos innecesarios aparezcan en el proceso, a diferencia del diseño simétrico, ya que este está basado en una sola unidad de control. (Digital Guide Ionos, 2019).

## 2.7. BENEFICIOS DE LAS REDES SDN

Según Punt Informatic (2018) las redes SDN cuentan con una gran cantidad de beneficios interesantes para cualquier usuario independiente u organización entre los cuales están los siguientes:

**Flexibilidad:** Son más flexibles ya que en el momento en que una compañía utiliza la virtualización para la infraestructura de su red se pueden acondicionar esta red a las distintas necesidades de la misma sin tener que adquirir equipos o dispositivos nuevos, lo único que se debe hacer es una reprogramación del software para que la red actualice los cambios realizados. Las SDN tienen una gestión centralizada y más simple, así mismo llegan a ser más eficientes que las redes con arquitectura tradicional porque su administración o manejo es de un mismo sitio, lo cual hace que todo sea de forma más sencilla.

**Ahorro de costos:** En comparación con las redes de arquitectura tradicional, las SDN debido a su manejo centralizado no requiere de altos costos, y este es uno de los beneficios por los cuales más personas y empresas están optando por el cambio de una red tradicional a una SDN.

**Automatización:** Gracias a la abstracción de los planos de control y datos, las SDN permiten balancear la carga y distribución del tráfico de forma eficiente para prevenir los puntos de bloqueo, esto conlleva a un mejor rendimiento y uso, adicional a esto, la instalación y configuración de estas redes son automáticas, de esta forma se aminoran los costos operacionales y de mantenimiento, al mismo tiempo que se simplifican funciones que han sido previamente instaladas.

**Seguridad:** Las SDN se vuelven más robustas con la automatización, también mejoran la seguridad ya que por el despliegue que tiene esta red en infraestructura se puede resolver de una manera más fácil cualquier tipo de inconveniente que esta tenga a partir de un plano de control centralizado en menor cantidad de tiempo a diferencia de las redes tradicionales. Uno de los atractivos de este tipo de redes es que brindan una seguridad con gran precisión en aplicaciones, puntos finales y dispositivos BYOD (Bring your own device) esto claramente una red con arquitectura tradicional no puede proporcionar.

**Utiliza cloud computing:** Una de las mayores revoluciones en la tecnología es el almacenamiento en la nube, de manera que, las compañías comienzan a interesarse en actualizar sus sistemas con este servicio, para esto la estructura debe ser virtualizada y administrada de forma centralizada para que así se puedan establecer nuevos servicios en la parte superior de la red.

Este tipo de infraestructura por su flexibilidad y escalabilidad les permite a las compañías desarrollar, desplegar aplicaciones y servicios en pocos días en vez de semanas o meses, que es lo que comúnmente se demoraría si se aplica una red con arquitectura tradicional y como se sabe en la actualidad la velocidad es lo que más se cotiza en la industria, la mayoría de las empresas buscan que sus procedimientos sean llevados a cabo en un tiempo mínimo, para ellos la optimización del mismo es fundamental e indispensable, lo que hace que sea muy pedido por las organizaciones. (Punt Informatic Becomit Company, 2018).

## 2.8. POSIBLES ESCENARIOS DE APLICACIÓN SDN

Debido a las amplias ventajas que existen en las redes SDN en comparación con las redes de arquitectura tradicional, Digital Guide (2019) menciona los posibles escenarios en las que estas redes pueden presentarse:

**Calidad de servicio (QoS):** Es el control central de todos los nodos de red, brinda facilidad al administrador de red para saber cuánto ancho de banda se utiliza una sola conexión, de manera que pueda dar una respuesta con rapidez.

Así mismo regula el tráfico de distribución de datos para otorgar a todos los participantes el ancho de banda necesario en todo momento.

**Gestión de dispositivos:** La utilización de un protocolo uniforme como OpenFlow convierte a las SDN en una solución con resultados extraordinarios si se gestiona con terminales de diferentes fabricantes en una red.

**Amplificación de las funciones de la red:** La independencia que brinda la tecnología SDN también representa una solución excelente en los escenarios donde se amplían las funciones de las redes en cualquier ocasión y sin problemas, adicional a esto, llega a ser una ventaja determinante para el usuario la libertad respecto a los fabricantes de hardware.

**Enrutamiento de paquetes controlado por aplicación:** Las SDN proporcionan la comodidad necesaria para que las aplicaciones externas logren participar en el enrutamiento de paquetes, es decir, modificar y ajustar routers en la red. La circunstancia idónea para ello es que la unidad de control disponga de la interfaz correspondiente.

**Políticas de seguridad:** Básicamente se establecen por medio de la unidad central de control, las mismas que permiten enviar directrices de seguridad a los conmutadores de la red fácilmente. (Digital Guide Ionos, 2019).

Las SDN funcionan con un protocolo OpenFlow, existen algunos que son compatibles con ellas y otros que son plataformas suman aplicaciones en el entorno de la nube.

## 2.9. PROTOCOLO OPENFLOW

Hoy en día solo se utiliza OpenFlow como protocolo open source para la comunicación de las redes SDN, se usa NETCONF como protocolo de gestión de red internet y adicional se pueden adherir plataformas como OpenStack.

Openflow es una tecnología de switching que surgió a raíz del proyecto de Investigación: “OpenFlow: Enabling Innovation in Campus Networks” que surgió

en el año 2008 en la Universidad de Stanford. Se define como un protocolo emergente y abierto de comunicaciones que permite a un servidor de software determinar el camino de reenvío o distribución de paquetes que debería seguir en una red de switches. Con el protocolo OpenFlow, una red puede ser gestionada como un todo y no como un número de dispositivos que se gestionan de manera individual, de esta manera es el propio servidor el que dice a los switches dónde deben enviar o distribuir los paquetes. Con esta tecnología, las decisiones que impliquen el movimiento o traslado de paquetes de datos están centralizadas, por lo que la red puede ser programada independientemente de los switches.

En un switch convencional, el reenvío o distribución de paquetes denominado plano de datos y el encaminamiento de alto nivel se denomina plano de control, los dos se realizan en el mismo dispositivo, sin embargo, en los switches OpenFlow ambos se separan. Con OpenFlow, una parte del plano de datos reside en el mismo switch, pero es un controlador el que realiza las decisiones de encaminamiento de alto nivel. Ambos elementos se comunican por medio del protocolo OpenFlow.

Esta metodología conocida como SDN permite una mayor efectividad en el uso de los recursos de la red que en una red de arquitectura tradicional. OpenFlow está pensado para afrontar la movilidad de máquinas virtuales, redes con misiones críticas o redes NGN móviles. (Ortin Calabrese, 2016).

### 2.9.1. APLICACIONES DEL PROTOCOLO OPENFLOW

Existen algunas aplicaciones y plataformas que integran el protocolo OpenFlow, cada una de ellas cuenta con una función específica que contribuye a que sea más dinámica la funcionalidad de este protocolo, dando como resultado una experiencia de administración más sencilla al usuario.

Vélez (2018) menciona algunas de las aplicaciones disponibles y más utilizadas en el desarrollo para la configuración de una SDN con OpenFlow.

**Visor de flujo (FlowVisor):** Es la aplicación que permite ver o visualizar el flujo de datos que existe sobre una topología de red, filtrando la información con características específicas como el tipo de datos, destino y remitente. Esta aplicación utiliza por defecto los puertos 8080 y 6633 para OpenFlow versión 1.0.

**Aster'X:** Esta aplicación se utiliza más en una topología de red dedicada principalmente a voz IP, se efectúa de manera dinámica con un balance de cargas, la cual mejora la calidad de servicio y baja el porcentaje de utilización de cada elemento de red.

**Usando toda la red inalámbrica que me rodea:** Esta aplicación se establece o implementa en un proceso de traslado sobre una red SDN. La implementación se realiza bajo una aplicación que se basa en streaming utilizando una red Wifi y una red WiMAX.

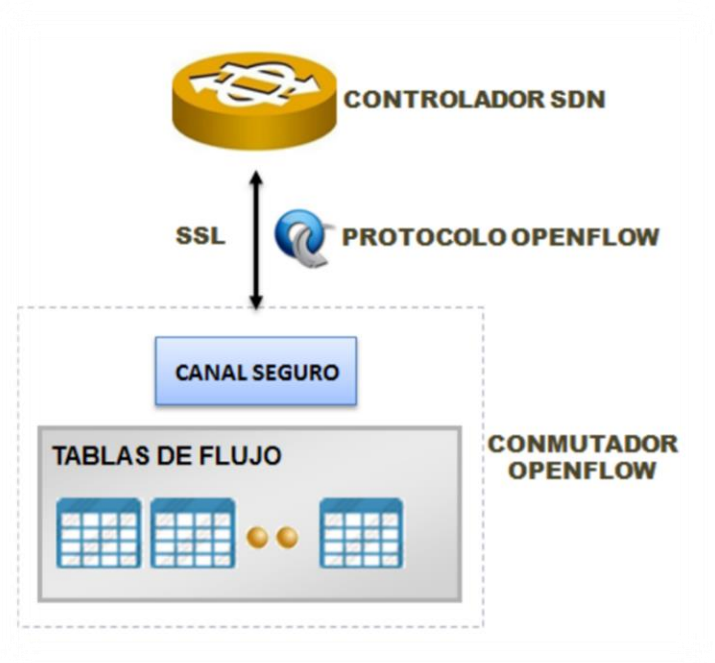
**ElasticTree:** Esta aplicación es la que conforma un centro de datos y permite evaluar el consumo total de energía con y sin la arquitectura de red definida por software o SDN.

**Canales abiertos (Open Pipes):** Esta es una plataforma que se utiliza para la construcción de sistemas de hardware, los cuales son distribuidos en módulos y al mismo tiempo son conectados en diferentes puertos físicos que existen en una red OpenFlow. (Velez Mejia, 2018).

### 2.9.2. COMPONENTES DEL CONMUTADOR OPENFLOW

Los conmutadores ethernet usualmente abarcan tablas de flujo y en el modelo del protocolo OpenFlow da la opción de poder editar estas tablas de flujo de una manera más dinámica con la posibilidad de incorporar cortafuegos, NAT, QoS y recolectar estadísticas.

**FIGURA 2.3.** Componentes del conmutador OpenFlow




**Nota:** La figura representa como los conmutadores OpenFlow tienen la capacidad de sostener distintas funcionalidades dentro de su arquitectura, a pesar de que todos ellos poseen características muy comunes. Tomado de *Plataforma de pruebas para evaluar el desempeño de las redes definidas por software basadas en el protocolo openflow* (p.23), por Y.A. Marín, 2016, Universidad Central “Marta Abreu” De Las Villas.

### 2.9.3. ARQUITECTURA DE LOS CONMUTADORES OPENFLOW

La arquitectura de los conmutadores OpenFlow es básica y sencilla, esto se debe a que se busca una administración simple para el usuario, se componen de tres partes las cuales son asociadas entre sí, formando una construcción correcta y funcional para la configuración y el traslado de los datos.

Según Marín (2016) las tres partes que conforman la arquitectura son las siguientes:

- Tabla de flujos
- Canal seguro
- Controlador

**Tabla de flujos:** Esta tabla funciona mediante una acción que está asociada con cada una de las entradas de la tabla, la misma que comunica al switch ya establecido la manera en la que se debe procesar el flujo.

**Canal seguro:** Por medio del canal seguro se establece la conexión del switch a un proceso de control remoto, en este caso lo es el controlador, el mismo que abre el paso a la distribución de los comandos y los paquetes de datos que existen entre el conmutador y controlador.

**Controlador:** El controlador en este caso tiene la función de adicionar o borrar las entradas que puedan ingresarse en las tablas de flujo.

Los conmutadores convencionales utilizan protocolos que son específicos para disponer de la distribución de los de datos, con el protocolo OpenFlow esta decisión de distribución de los paquetes de datos se transfiere o traslada desde los conmutadores a los controladores, los cuales se ejecutan principalmente en un equipo que es utilizado como servidor o en uno que también puede ser utilizado como una simple estación de trabajo.

La aplicación de esta gestión dará comienzo su ejecución en la interfaz del controlador, la misma que compila o acopla en la red a todos los conmutadores disponibles, para de esta manera dar más facilidad a la configuración de los trayectos que son utilizados como distribución de datos que ocupan el espacio restante y que está disponible de la banda ancha. (Marín Muro, 2016).

### 2.9.4. TIPOS O VERSIONES DE OPENFLOW

Según Carrillo (2020) existen las siguientes versiones de OpenFlow:

**OpenFlow 1.0**

- Surge en el mes de diciembre del año 2009.
- Contiene una sola tabla de flujo con tres componentes: campos de cabecera, contadores y acciones.
- Solo cuentan con doce campos de coincidencia en los campos de cabecera.
- Ofrece poca flexibilidad de administración sobre la red.
- Contiene una sola tabla de flujo, lo que da como resultado algunos problemas de explosión de entradas.

**OpenFlow 1.1**

- Surge en el mes de febrero del año 2011.
- Incorpora múltiples tablas de flujo y grupos de tablas, lo que atribuye un mejor manejo de los paquetes en el flujo de la red.
- Los campos de cabecera y las acciones son renombradas a campos de coincidencia e instrucciones respectivamente.

**OpenFlow 1.2**

- Surge en el mes de diciembre del año 2011.
- Incorpora la estructura TLV (Type Lenght Value) que atribuye adicionar más campos de comparación de una manera más modular. Esta funcionalidad se denomina OpenFlow Extensible Match (OXM).
- Incluye un soporte básico para IPv6 a través de OXM.
- Se incorpora el mecanismo de cambio de rol de controlador, lo que atribuye a utilizar más de un controlador en una red y reduce el riesgo del único punto de falla.

**OpenFlow 1.3**

- Surge en el mes de abril del año 2012.
- Se perfecciona el soporte para QoS con la introducción de la tabla de medición.
- Cada entrada en la tabla de medición abarca una lista de bandas de medidor, las mismas que detallan el comportamiento de la red ante cierto tipo de tráfico.
- Se amplía la tabla de flujo con una tabla de entradas perdidas, esto atribuye a la obtención de un mejor comportamiento de la red al momento en que un paquete no se ajuste con las entradas de las tablas de flujo y que fueron desechados.

**OpenFlow 1.4**

- Surge en el mes de agosto del año 2013.
- Incorpora la sincronización de tablas las mismas que pueden ser de forma bidireccional o unidireccional.
- Se adiciona una nueva característica denominada empaquetamiento, que atribuye la preparación y validación de varios mensajes OpenFlow los cuales serán aplicados por múltiples switches.

**OpenFlow 1.5**

- Surge en el mes de enero del año 2015.
- Incorpora la calendarización de paquetes por medio de la inclusión de tiempo de ejecución como propiedad del paquete. Un conmutador que recibió un paquete programado aplicará el mensaje lo más cerca posible del tiempo de ejecución. Esto fortalece aún más la sincronización entre múltiples switches.

## 2.10. PROTOCOLO NETCONF

El Protocolo de configuración de red o Network Configuration Protocol NETCONF brinda herramientas para instalar, mantener y eliminar configuraciones de dispositivos de red. Este protocolo puede utilizarse para adquirir diferentes configuraciones y el estado de los dispositivos de red. También autoriza que un dispositivo de red suministre interfaces de programación de aplicaciones estándar, que mediante las aplicaciones permiten enviar y obtener datos de configuración del dispositivo de red.

A medida que aumentan la escala y la complejidad de la red, el protocolo simple de administración de redes SNMP de arquitectura tradicional se ha quedado atrás de las demandas para una administración más sencilla, especialmente la administración de la configuración de redes complejas. El protocolo NETCONF se basa en Extensible Markup Language XML y se ha creado para brindar la gestión de configuración necesaria.

Adicional este protocolo puede implementarse aplicando las funciones existentes de un dispositivo. Esto aminora los costos de implementación y posibilita un acceso fácil a las nuevas funciones. De la misma manera en que permite descubrir funciones extendidas admitidas por un servidor y ajustar su comportamiento para utilizar las funciones proporcionadas por el dispositivo. (Huawei Enterprise, 2021).

### 2.10.1. CARACTERÍSTICAS DEL PROTOCOLO NETCONF

Según Huawei Enterprise (2021) menciona las siguientes características:

- Brinda un método de bloqueo para proteger y evitar conflictos de configuración.
- Permite consultar directamente cualquier dato de configuración en el sistema y filtrar los datos de configuración para consultar.
- Ofrece una buena escalabilidad.

- Utiliza un modelo multicapa, en el que cada capa es independiente de otras capas. La extensión de una capa tiene poco efecto sobre otras capas.
- Usa el formato de codificación XML para ampliar la capacidad de gestión del protocolo y la compatibilidad del sistema.
- Emplea protocolos de seguridad existentes para garantizar la seguridad y no está vinculado a ningún protocolo de seguridad específico.
- Es más flexible que el SNMP para garantizar la seguridad.
- Prefiere Secure Shell (SSH) como protocolo de capa de transporte para transmitir mensajes XML.

## 2.11. OPENSTACK

OpenStack es una plataforma que utiliza open source y recursos virtuales en conjunto para diseñar y gestionar nubes privadas y públicas. Las opciones de herramientas que conforman la plataforma OpenStack tienen por nombre “proyectos” y se encargan de los servicios principales de cloud computing, como lo son la computación, las redes, el almacenamiento y la identidad e imagen. Además, se pueden juntar más de una docena de proyectos opcionales para desarrollar nubes únicas que se pueden implementar.

En la virtualización, los recursos, como el almacenamiento, la CPU y la RAM, se extraen de distintos programas específicos de los proveedores y se dividen con un hipervisor antes de ser distribuidos. OpenStack utiliza un conjunto uniforme de interfaces para la programación de aplicaciones y así extraer todavía más recursos virtuales, los cuales se distribuyen en conjuntos distintos que se utilizan para potenciar las herramientas del cloud computing y estándares que emplean los administradores y los usuarios. (Redhat, Sf).

### 2.11.1. FUNCIONAMIENTO DE OPENSTACK

OpenStack es un protocolo que funciona con una serie de comandos conocidos como scripts. Esos scripts están agrupados en paquetes denominados “proyectos”, que transmiten las tareas que generan los entornos en la nube. Para que se puedan crear esos entornos, OpenStack depende de otros dos tipos de software entre estos se mencionan los siguientes:

**La virtualización**: Es la base de la informática que tiene como función establecer los recursos virtuales extraídos del hardware.

**Un sistema operativo**: Es un conjunto de órdenes y programas que ejecutan los distintos comandos que provienen de los scripts de OpenStack.

La plataforma OpenStack por sí sola no es capaz de virtualizar los recursos, sino que los utiliza para esquematizar nubes. De la misma manera no puede ejecutar los comandos, sino que los traslada al sistema operativo base. Las tres tecnologías como lo son OpenStack, la virtualización y el sistema operativo base, obligatoriamente tienen que trabajar en conjunto. Estas tres tecnologías reflejan dependencia y también por qué la gran parte de las nubes de OpenStack se implementan con el sistema operativo Linux. (Redhat, Sf).

### 2.11.2. COMPONENTES DE OPENSTACK

La plataforma OpenStack se conforma por una enorme cantidad de proyectos con código abierto, los mismos que son utilizados en esta arquitectura para establecer el undercloud y el overcloud del protocolo OpenStack, que usa los administradores de sistemas y los usuarios de la nube, respectivamente. Los underclouds están constituidos por algunos componentes clave que necesitan a los administradores de sistemas para configurar y gestionar todo el ambiente OpenStack de los usuarios finales, los cuales son conocidos como overclouds.

Pueden encontrarse seis servicios básicos permanentes que administran la informática, las conexiones en red, el almacenamiento, la identidad y las

imágenes, y más de doce servicios que son opcionales y cambian de acuerdo a la solidificación del desarrollo. Los seis servicios principales conforman la infraestructura que autoriza a los proyectos restantes gestionar los paneles, la coordinación, el aprovisionamiento de equipos sin sistema operativo, la mensajería, los contenedores y la gobernabilidad. (Redhat, Sf).

Según Redhat (Sf) la arquitectura OpenStack cuenta con los siguientes componentes:

**Nova:** Es una herramienta que se basa en controlar la planificación, creación y la eliminación de todos los datos de gestión y acceso.

**Neutron:** Se encarga de la conexión de las redes disponibles con los demás servicios que existen y que ofrece el protocolo OpenStack.

**Swift:** Es un servicio que sirve para almacenar objetos de datos, este cuenta con una máxima tolerancia a fallos, adicional a esto guarda y restaura los objetos de datos que no están estructurados.

**Cinder:** Proporciona una cantidad de almacenamiento persistente de los bloques, y también se puede acceder a él a través de una aplicación de autoservicio.

**Keystone:** Autentica y autoriza todos los servicios disponibles de OpenStack. Adicional también es el catálogo de todos los servicios.

**Glance:** Almacena y recupera las imágenes del disco de la máquina virtual desde varias ubicaciones.

Para poner en práctica todos estos protocolos y plataformas se necesita un software de virtualización como base, así se podrá llevar a cabo la implementación de los controladores compatibles con SDN y sus protocolos.

## 2.12. SOFTWARES PARA LA VIRTUALIZACIÓN

Existen muchos softwares para la virtualización, en este caso se mencionan tres, que son los más utilizados para este tipo de estudio:

VMware, Oracle VM VirtualBox y Microsoft Hyper-V.

### 2.12.1. VMWARE

VMware es una plataforma que brinda soluciones eficaces y contiene una gran lista de paquetes de software completamente disponibles para la virtualización.

Esta plataforma cuenta con servicios, soluciones y aplicativos en el ámbito de escritorio que funcionan con licencias pagadas. Algunas versiones ofrecen la oportunidad de poder aplicar ciertas herramientas de manera gratuita para las organizaciones. (Profesional Review, 2018).

### 2.12.2. ORACLE VM VIRTUALBOX

VirtualBox es una plataforma que permite al usuario crear ambientes de virtualización y la instalación de máquinas virtuales con los sistemas operativos Linux, Windows y Mac con opción de conectarse en red de manera física.

Utiliza herramientas denominadas "**Guest additions**" las mismas que sirven y permiten al usuario poder interactuar con ellas de manera más avanzada autorizando opciones, para copiar y pegar documentos de forma directa. También dispone de una versión portable. VirtualBox soporta ambas tecnologías de virtualización de **Intel** y **AMD**. (Profesional Review, 2018).

### 2.12.3. MICROSOFT HYPER-V

Esta plataforma se **encuentra disponible de forma nativa** en los sistemas operativos versión Pro y Server, es decir quien tenga instalado Windows 10 Pro tendrá acceso para utilizar Hyper-V de manera totalmente gratis. Esta herramienta permite virtualizar sistemas operativos con todo incluido, incluso el hardware como si fueran máquinas reales, tal y como hacen VirtualBox y VMware. Además, es compatible con las tecnologías de AMD-V y Intel VT-x.
Si se cuenta con el sistema operativo de Windows Server, éste ya vendrá equipado con algunas funciones extras como:

- Función de red SR-IOV.
- Traslado de máquinas virtuales desde un servidor a otro.
- VHDX compartido.

Uno de los problemas que puedan producirse con Hyper-v es que de error en la ejecución del mismo si ya cuenta con otros programas de virtualización instalados. (Profesional Review, 2018).

## 2.13. CONTROLADORES SDN

Las SDN vienen equipadas con un componente muy importante, este componente se denomina controlador. Un controlador tiene la función de transmitir los requerimientos que van de la capa de aplicación a los elementos de la red, por lo consiguiente se podría llegar a la conclusión que un controlador vendría a ser la parte cerebral de toda la arquitectura ya que es quien maneja los flujos y da las órdenes pertinentes para que los demás componentes las cumplan. (Carrillo Rodas, 2020).

Hoy en día existen muchos controladores OpenFlow y no OpenFlow, en este caso se nombra específicamente los más utilizados con el protocolo OpenFlow.

### 2.13.1. OPEN DAYLIGHT

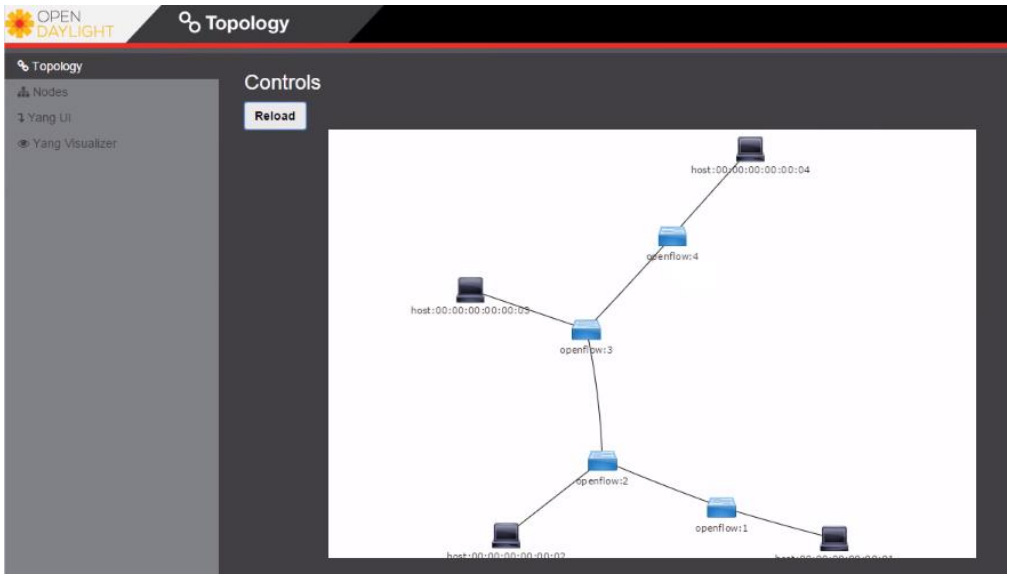
OPEN DAYLIGHT es una plataforma modular de libre acceso que sirve para personalizar y automatizar las redes de cualquier tipo de tamaño y escala. Proporciona el control programático con características de centralización y monitoreo de los dispositivos que conforman la red. Esta plataforma está basada en el lenguaje de programación java y utiliza como herramienta open source de creación a Apache Maven. (Open Daylight, 2021).

Apache Maven es la base más utilizada por los compiladores actuales, se utiliza como herramienta de estandarización, es decir que maneja la configuración de compilado, empaquetado y la instalación de las librerías que luego podrán ser manejadas por los desarrolladores. (Yagüe, 2019)

Open Daylight hoy en día es una opción que destaca entre las demás plataformas, según Open Daylight (2021) entre las características más relevantes se pueden mencionar las siguientes:

- Permite conectividad con OpenStack.
- Brinda servicio de plataforma de autenticación, autorización y contabilidad.
- Los esquemas de datos están basados en la estructura en forma de árbol.
- Genera automáticamente códigos de interfaz.
- Proporciona conectividad a través de los protocolos openflow mediante openflowplugin.
- Soporta una gran cantidad de protocolos de red OpenFlow y no OpenFlow.
- Utiliza Apache Maven.
- Administración sencilla.

**FIGURA 2.4.** Controlador OPEN DAYLIGHT.




**Nota:** Esta figura muestra la interfaz del software OPEN DAYLIGHT en el cual se puede visualizar la arquitectura de una red en forma de árbol con cuatro switches openflow y cuatro hosts. Tomado de OPEN DAYLIGHT (ODL) and MiniNet demo - SDN & OpenFlow on GNS3 [Fotografía], por David Bombal, 2016, LaptrinhX. CC BY 2.0.

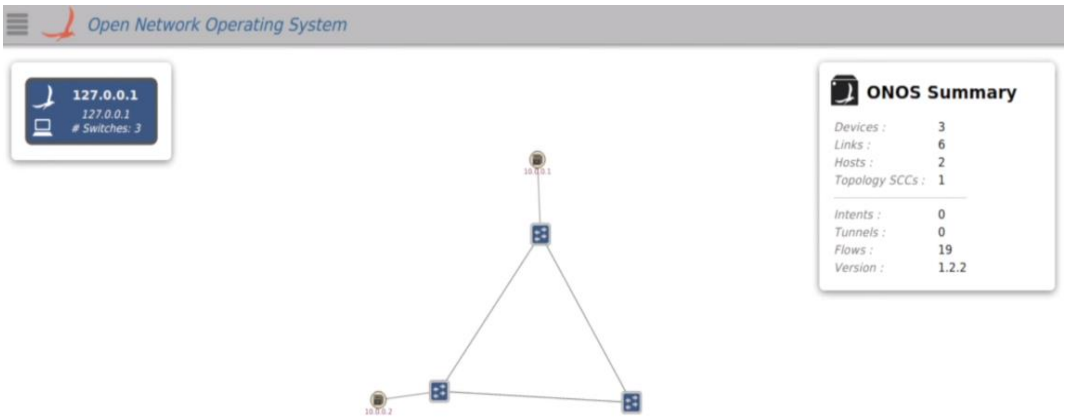
### 2.13.2. ONOS

ONOS es un controlador que trabaja en la nube y es una de las plataformas open source más usadas para la creación de las SDN. Admite configuraciones de tiempo real de la red, esto suprime la necesidad de ejecutar protocolos de conmutación y enrutamiento dentro de la estructura de la red. También permite al usuario crear de manera sencilla aplicaciones de red sin tener la necesidad de modificar los sistemas del plano de datos. (ONOS, 2021)

Según ONOS (2021) las características que predominan a este controlador son:

- Funciona con un gran número de aplicaciones que actual como un controlador SDN distribuido, modular y extensible.
- Su gestión, configuración e implementación son sencillas incluyendo software, hardware y servicios.
- Su arquitectura es escalable de forma horizontal, de esta manera ofrece flexibilidad y escalabilidad.

**FIGURA 2.5.** Controlador ONOS.




**Nota:** En esta figura se puede apreciar la interfaz gráfica de ONOS en la cual se forma una topología en forma de anillo con un controlador, tres switches openflow y dos hosts. Tomado de ONOS: Getting Started [Fotografía], por Furrukh Fahim, 2015, Youtube. CC BY 2.0.

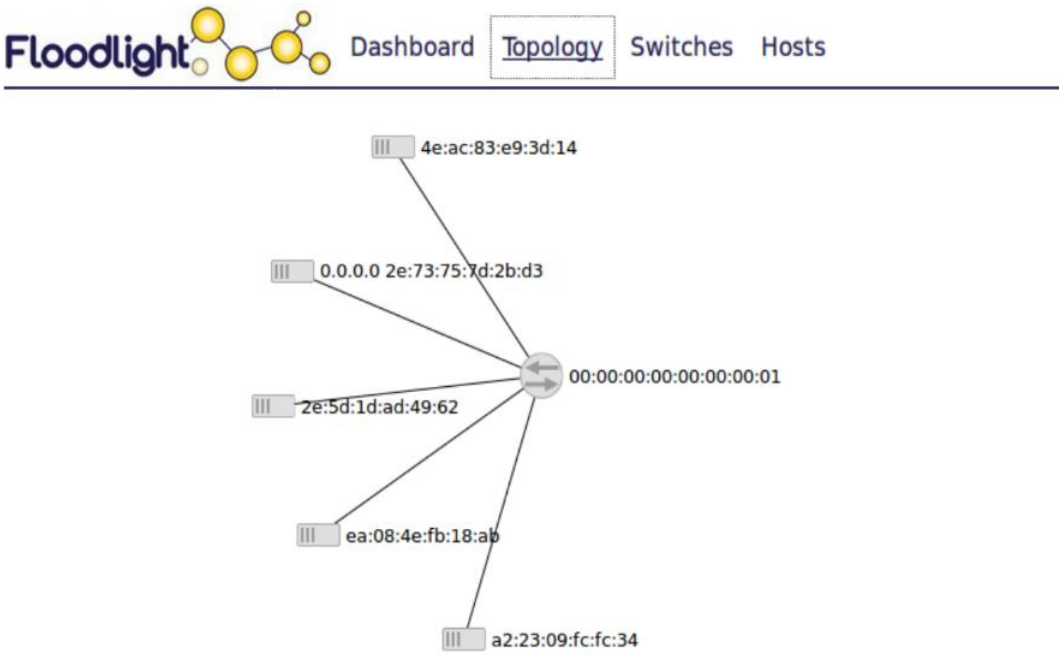
### 2.13.3. FLOODLIGHT (JAVA)

Floodlight es un controlador OpenFlow que está basado en el lenguaje de programación Java, es uno de los más utilizados ya que contiene una máquina virtual que viene ya con configuraciones previas y con el software MiniNet integrado, Open vSwitch y Floodlight en su primera versión. También brinda a los usuarios maneras de ver la información sobre el estado del controlador y monitorización de los equipos remotos. Además, permite conectar varios switches, hosts en la red, tablas de flujo y todo lo que contiene una topología de red. (Floodlight, 2016).

Entre sus principales características Floodlight (2016) nombra las siguientes:

- Ofrece módulos de sistema de carga que son escalables y de fácil administración.
- Configuración sencilla.
- Es compatible con una gran cantidad de switches OpenFlow, virtuales y físicos.
- Está creado para brindar un alto rendimiento, opciones de seguridad eficaces, y es tolerante a fallos.
- Permite el uso de OpenStack.

**FIGURA 2.6.** Controlador Floodlight.




**Nota:** Así trabaja la interfaz gráfica del controlador Floodlight, en este ejemplo se puede visualizar una topología de forma lineal, la cual cuenta con un switch OpenFlow y cinco hosts. Elaboración Propia.

## 2.14. MININET

MiniNet es un software que crea redes virtuales, utiliza switches, controladores y conexiones. Es compatible con Linux y permiten usar el protocolo OpenFlow. Este software es muy utilizado para desarrollar compartir y experimentar con otros sistemas de SDN. Adicional es una buena herramienta para desarrollar, enseñar, investigar, crear y simular redes que serán de mucha ayuda en un entorno educativo o a nivel corporativo. (Mininet, 2021)

### 2.14.1. CARACTERÍSTICAS DE MININET

MiniNet puede combinar grandes características que poseen los demás emuladores, bancos de pruebas de hardware y simuladores.

Según MiniNet (2021) se pueden mencionar las siguientes características:

En virtualización de sistemas completos

- Enciende de forma más rápida.
- Su escalabilidad es enorme.
- Incrementa el ancho de banda.
- Es de instalación sencilla.

En bancos de pruebas de hardware

- Es económico y de alta disponibilidad.
- Su reconfiguración es veloz y reiniciable.

En otros simuladores

- Trabaja con código real sin hacer modificaciones.
- Es de fácil conexión a las redes reales.
- Su rendimiento es interactivo.

Características generales de MiniNet:

- Ofrece al usuario bancos de pruebas de red sencillo y económico.
- Se puede trabajar independiente, pero en la misma topología por varios desarrolladores al mismo tiempo.
- Autoriza pruebas de regresión a nivel de sistema.
- Sus herramientas permiten pruebas de topologías complejas.
- Mediante CLI reconoce topologías y OpenFlow.
- Permite la utilización de topologías personalizadas y parametrizadas.

- No es necesario usar programación, pero cuenta también con una aplicación del lenguaje Python.
- Administra la estructuración y el comportamiento del sistema de una manera correcta.
- Utiliza kernel real.
- Tiene integrada la función de admitir otros sistemas operativos con virtualización basada en procesos.

# DESARROLLO DE LA PROPUESTA TECNOLÓGICA

# CAPÍTULO II

# 3. CAPÍTULO II. DESARROLLO DE LA PROPUESTA TECNOLÓGICA

En este capítulo se expondrá los métodos, variables, la información de los componentes de hardware y software que se utilizaran para el desarrollo y posterior diseño de la red SDN que se va a proponer.

## 3.1 METODOLOGIA DE INVESTIGACION

Los tipos de métodos que se utilizarán en este trabajo son los siguientes:

**Descriptivo:** Dado que la información que será incorporada a este trabajo será mediante conceptos y definiciones ya previamente investigadas, de esta manera se obtendrá una base confiable de lo que se expone y plantea dentro de este trabajo.

**Exploratorio:** Puesto que se necesita ahondar un poco más en las problemáticas que se presentan y por lo consecuente llevar a cabo un análisis más detallado sugiriendo una propuesta de solución.

## 3.2 VARIABLES

Las variables tienen como función medir como se va a formar una hipótesis a partir de la información recaudada. Existen dos tipos de variables, independientes y dependientes.

Las variables independientes son la causa de la problemática, mientras que las variables dependientes son los resultados que se dieron a consecuencia de la previa investigación.

**TABLA 3.1.** Variables

| Variable | Conceptualización | Indicadores | Instrumentos y/o métodos |
|---|---|---|---|
| Red obsoleta en la empresa Eléctrico HAZ S.A. | La red es una ethernet simple de característica domestica | • Switch.<br>• Servidor.<br>• Router.<br>• Conmutador.<br>• Estaciones de trabajo. | entrevista |
| Propuesta de implementación SDN mediante el controlador Floodlight | Una SDN es una red administrable centralizada que simplifica procesos, es escalable, de alto rendimiento y reduce recursos. | • Software de virtualización.<br>• Controlador Floodlight.<br>• Software MiniNet.<br>• Plan de pruebas. | • Metodología de desarrollo.<br>• Instalación.<br>• Configuración de métricas y políticas mediante comandos.<br>• Simulación. |

**Nota:** Definición y comportamiento de las variables en este estudio. Elaboración propia.

## Métodos empleados e instrumentos de la investigación.

El método utilizado es el empírico ya que por medio de él se permite la obtención y elaboración de los datos. El método empírico para la recolección de datos utilizado en esta propuesta tecnológica es la entrevista.

### Componentes de hardware

El dispositivo portátil donde se va a realizar la instalación de la máquina virtual Floodlight que contiene MiniNet se compone de las siguientes características:

- Marca y Modelo: Toshiba Satellite P55W-C5204
- Procesador: Intel™ Core™ i7-5500U 2.40Ghz
- Memoria RAM: 8,00 GB
- Sistema Operativo: Windows 10 64 bits
- Disco Duro: 1TB

### Componentes de software

Los componentes de software que serán utilizados para esta propuesta son los siguientes:

- Virtual Box
- Floodlight
- MiniNet

Posterior a realizar el análisis de los softwares de virtualización y controladores en el capítulo anterior, se ha optado por utilizar el software Oracle VM VirtualBox y el controlador Floodlight ya que este último está basado en el lenguaje de programación Java, el cual maneja una interfaz web para la visualización de la topología, adicional a esto permite soportar unas grandes cantidades de switches que son de fácil configuración.

Previo a la instalación del controlador Floodlight y el software de virtualización Oracle VM VirtualBox se deben descargar los archivos de ejecución desde las siguientes direcciones url:

- Oracle VM VirtualBox
  https://www.virtualbox.org/wiki/Downloads
- Controlador Floodlight
  https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight%2BVM

### 3.3. INSTALACIÓN DE ORACLE VM VIRTUALBOX

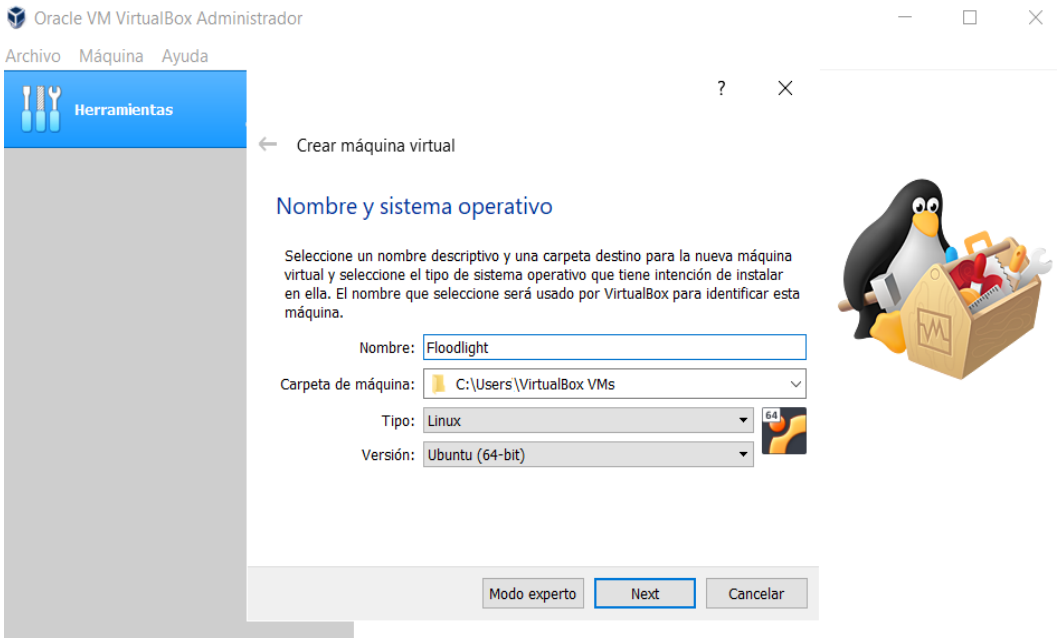
1) Primero se procede a ejecutar el archivo descargado siguiendo los pasos de instalación.

**FIGURA 3.1.** Instalación de Oracle VM VirtualBox.


**Nota:** Se muestra el primer paso para la ejecución del software de virtualización Oracle VM VirtualBox. Elaboración propia.

2) Luego de tener instalado Oracle VM VirtualBox en el dispositivo portátil que se utilizara para la realización de este proyecto, se selecciona la opción “Crear” y se escoge el archivo del controlador Floodlight previamente descargado, se le asigna nombre, en este caso es “Floodlight” y se procede a escoger el tipo de sistema operativo y su versión, el que se utilizará es Linux con su versión de Ubuntu 64 bits, así como se muestra en la Figura 3.2.

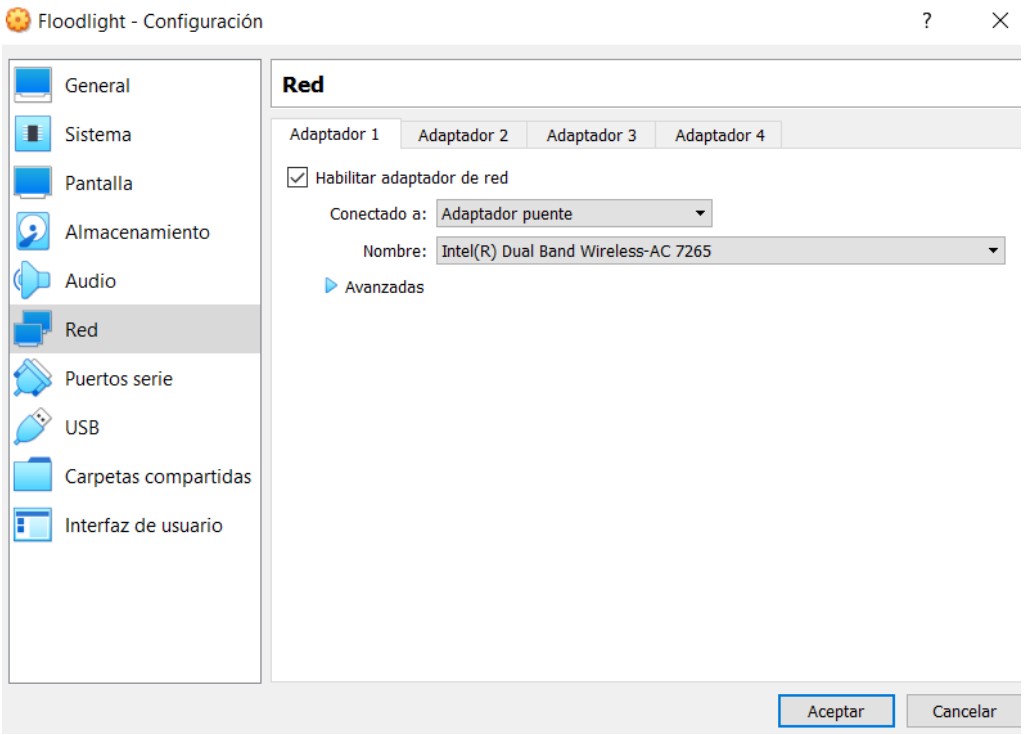
**FIGURA 3.2.** Importación de Floodlight a Oracle VM VirtualBox.




**Nota:** Se observa como Oracle VM VirtualBox permite configurar la cantidad de memoria RAM que será asignada a la máquina virtual que se requiere instalar, así mismo permite decidir la cantidad de gigabytes de disco duro disponible para su uso. Elaboración propia.

3) Este paso es muy importante para que pueda funcionar nuestro controlador de manera correcta, ya que muestra la configuración que va a llevar la máquina virtual Floodlight que está en proceso de instalación, para esto es necesario habilitar la tarjeta de red con la opción de “adaptador puente” y el nombre de la tarjeta de red que por defecto lo escoge automáticamente el software de virtualización.

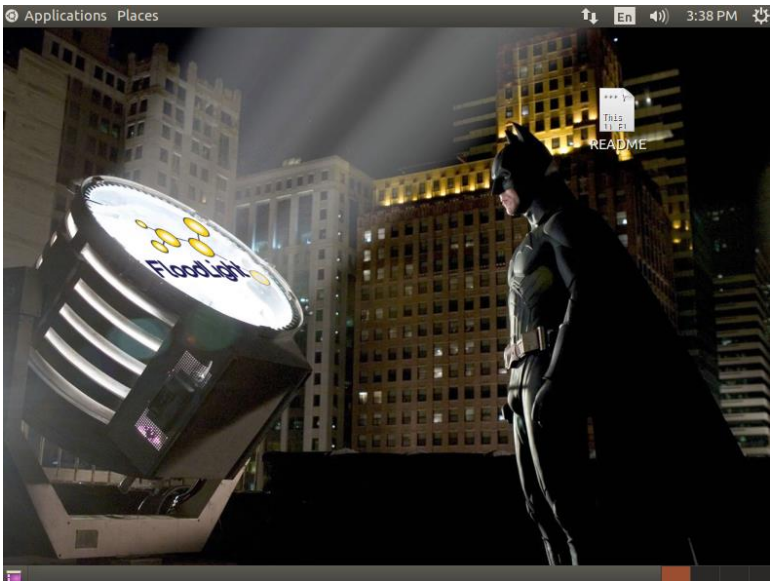
**FIGURA 3.3.** Configuración de tarjeta de red y adaptador puente.




**Nota:** Esta configuración permite que las máquinas virtuales creadas con sus diferentes componentes automáticamente se conecten a internet usando el adaptador puente que conecta al dispositivo base que se utiliza para realizar este proyecto. Elaboración propia.

4) Como último paso se procede a encender la máquina virtual la cual tiene como contraseña default: Floodlight

**FIGURA 3.4.** Ejecución del controlador Floodlight.



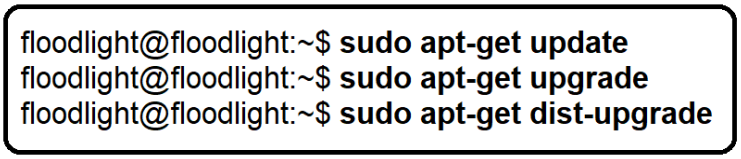

**Nota:** Pantalla de inicio del controlador Floodlight. Elaboración propia.

## 3.4. INSTALACIÓN DE MININET

Para continuar la instalación de MiniNet en la máquina virtual de Floodlight que se encuentra previamente descargada se deben realizar los siguientes pasos:

1) Es necesario realizar una actualización de todo el sistema operativo UBUNTU que es el que se utilizara para este proyecto, para proceder con dicha actualización se utilizarán en la terminal los siguientes comandos:

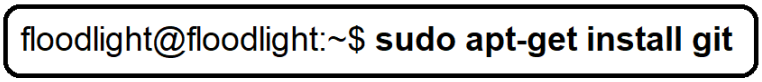
**FIGURA 3.5.** Comandos para la actualización de Ubuntu.




**Nota:** Estos comandos permiten actualizar a su última versión el controlador Floodlight. Elaboración propia.

2) Después de que se hayan realizado las debidas actualizaciones se procede a instalar el software Git mediante el siguiente comando:

**FIGURA 3.6.** Comando para la instalación del software Git.



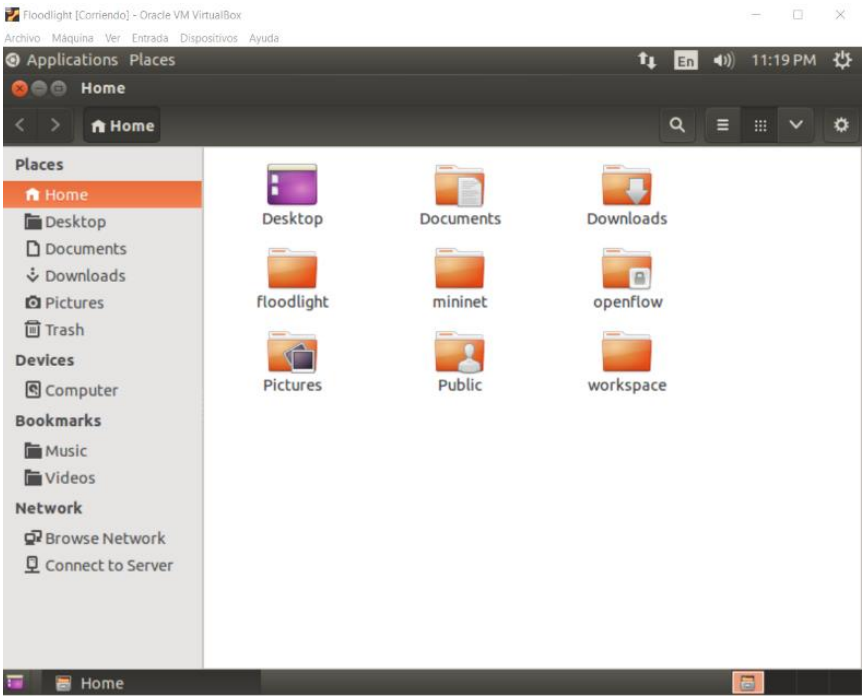

**Nota:** El comando Git permite la descarga de archivos de ejecución en código fuente almacenados en un repositorio, en este caso se procederá la descarga desde el repositorio de GitHub. Elaboración propia.

3) El siguiente paso es la descarga del software de instalación de MiniNet por medio del comando siguiente:

**FIGURA 3.7.** Descarga del software MiniNet.


**Nota:** En el comando anterior se puede visualizar que el software MiniNet es descargado del repositorio de GitHub. Elaboración propia.

**FIGURA 3.8.** Instalación del software MiniNet.




**Nota:** Este comando directamente crea una carpeta en Ubuntu en el directorio "Home" con el nombre de "MiniNet" la misma que contiene otras carpetas y archivos que son necesarios para que pueda funcionar de manera correcta al momento de proceder a ejecutarlo. Elaboración propia.

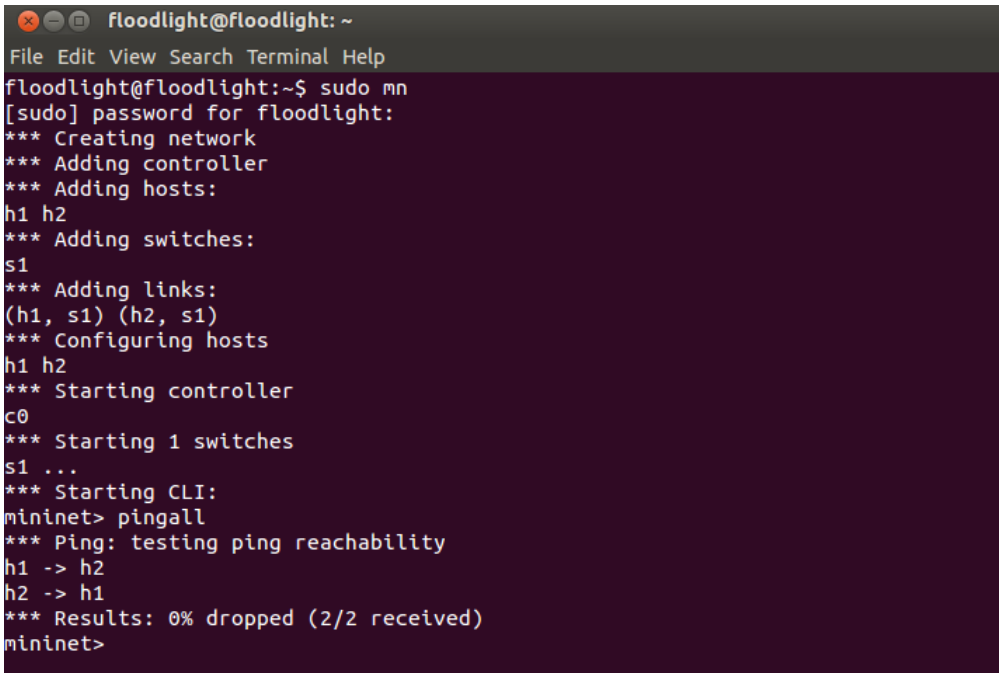
4) Para asegurarse que el software fue debidamente instalado se procede a ejecutar el siguiente comando:

**FIGURA 3.9.** Comando de Ingreso a MiniNet y comprobación mediante ping.

```
floodlight@floodlight:~$ sudo mn
mininet> pingall
```

**Nota:** Este comando agrega el controlador y también los hosts, al mismo tiempo que indicará si existe conexión entre los mismos. Elaboración propia.

**FIGURA 3.10.** Verificación de instalación de MiniNet.




**Nota:** Esta figura demuestra que MiniNet se instaló correctamente. Elaboración propia.

Luego de haber comprobado que el software de MiniNet fue debidamente descargado e instalado con conexión entre los dispositivos disponibles se empieza la creación del diseño de red.

## 3.5. COMANDOS MININET

Si se utiliza Linux o Ubuntu como sistema operativo se tiene que tener en cuenta que para ejecutar un comando en MiniNet en la terminal de estos sistemas operativos siempre se comienza con “sudo mn” esto creará topologías básicas adicionando switches y estableciendo una conexión con los hosts.

A continuación, se muestra una tabla con los comandos más comunes del software MiniNet, la mayoría utilizados para esta propuesta tecnológica.

**TABLA 3.2.** Comandos MiniNet.

| Comando | Definición |
|---|---|
| ping | Comprueba la comunicación con otro equipo remoto. |
| pingall | Verifica la comunicación que existe entre todos los equipos remotos. |
| ifconfig | Obtiene información de la interfaz de red. |
| xterm | Abre una terminal adicional para equipos especificados. |
| dump | Indica información detallada, condición de los puertos, tipos de dispositivos usados y datos IP. |
| help | Es un comando de ayuda y muestra información sobre otros comandos. |
| nodes | Visualiza la información de los nodos emulados. |
| net | Visualiza en pantalla las conexiones y los puertos enlazados entre los equipos remotos. |
| exit | Finaliza y cierra el programa. |

**Nota:** Los comandos mostrados son los más comunes al momento de utilizar el software MiniNet. Elaboración propia.

# DISEÑO DE LA RED

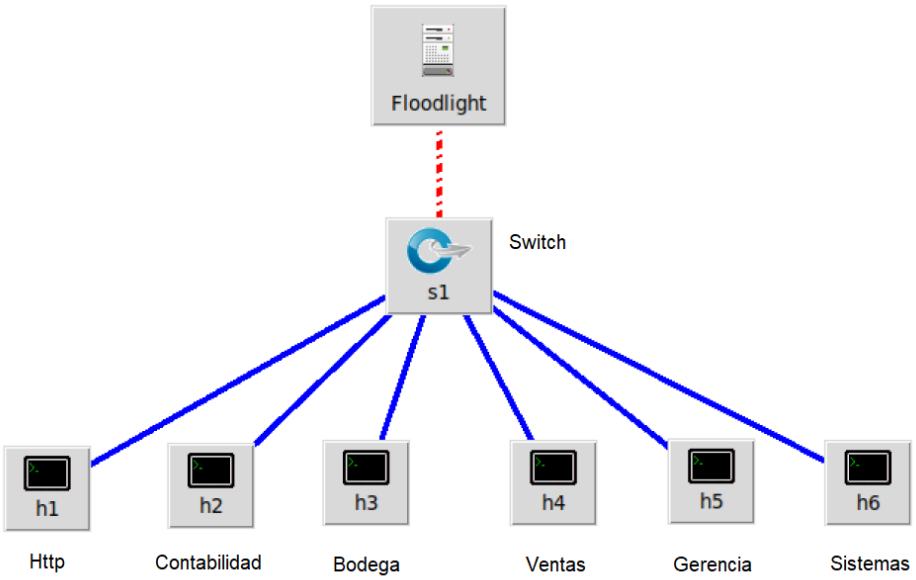
# CAPÍTULO 3

# 4. CAPÍTULO III. DISEÑO DE LA RED

En este capítulo se creará el diseño de red de acuerdo a las posibilidades de la empresa a la cual se le pretende presentar la propuesta de actualización de red.

## 4.1. TOPOLOGÍA DE SIMULACIÓN

**FIGURA 4.1.** Topología de la red SDN.




**Nota:** Existen algunas topologías de red que pueden ser creadas en MiniNet, entre ellas están la topología de estrella, malla, lineal, árbol, bus y anillo. Algunas ya vienen con su creación por defecto en el software, otras se pueden crear de forma sencilla. Elaboración propia.

La topología que se va a crear es lineal ya que la empresa solo cuenta con un servidor y por el momento no está en sus planes incorporar más servidores a su red, pero tampoco lo descartan debido que, con la actualización de su red local a SDN le permite ser escalable sin ninguna complicación.

**TABLA 4.1.** Nombre y direccionamiento IP y MAC.

| Áreas | Nombre | IP | MAC |
|---|---|---|---|
| HTTP | H1 | 10.0.0.1 | 46:7c:af:0a:3f:5b |
| CONTABILIDAD | H2 | 10.0.0.2 | 26:17:47:76:1d:9e |
| BODEGA | H3 | 10.0.0.3 | 42:e3:87:66:39:dd |
| VENTAS | H4 | 10.0.0.4 | 5ª:91:ab:42:08:0e |
| GERENCIA | H5 | 10.0.0.5 | ea:36:d2:db:9f:d1 |
| SISTEMAS | H6 | 10.0.0.6 | 42:cd:eb:7d:d2:97 |

**Nota:** Esta tabla muestra las áreas de la topología con su IP y MAC. Elaboración propia.

## 4.2. CONFIGURACIÓN DE LA TOPOLOGÍA LINEAL

La topología que se propone en este caso, se realizará en MiniNet, este software permite realizar diseños de topologías personalizadas en el lenguaje de programación Python. Se utilizará la información previamente mostrada en los apartados anteriores.

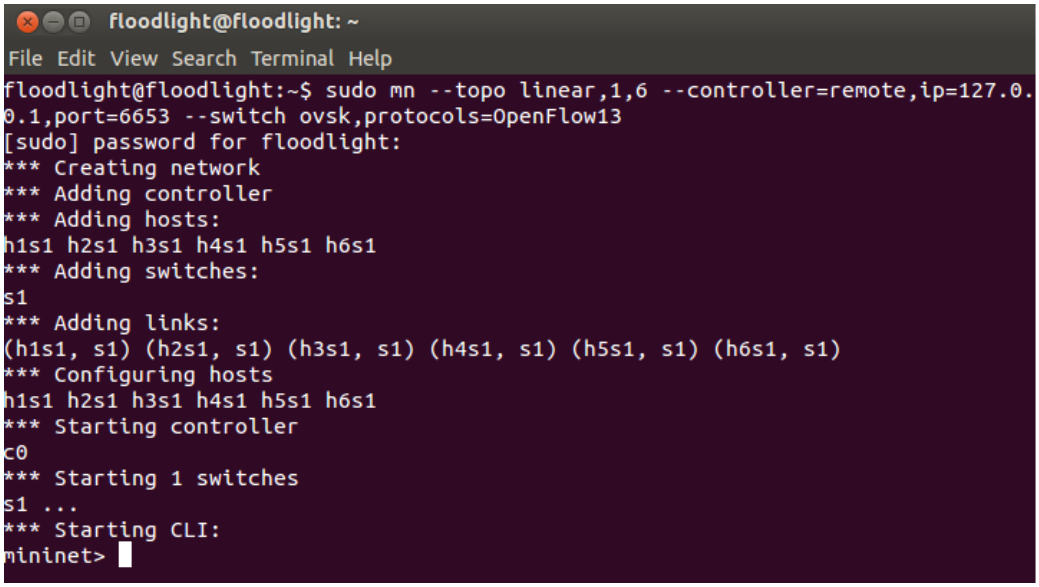
Para empezar con la construcción de la topología lineal se utilizará el comando siguiente:

**FIGURA 4.2.** Creación de la topología lineal.

```
floodlight@floodlight:~$ sudo mn --topo linear,1,6 --controller=remote,ip=
127.0.0.1,port=6653 --switch ovsk,protocols=OpenFlow13
```

**Nota:** Este comando permite adherir los hosts y switches agregando conexiones entre sí para de esta forma crear una arquitectura lineal por medio del controlador Floodlight. Elaboración propia.

**FIGURA 4.3.** Adición de conexión y configuración de los hosts.




**Nota:** En esta figura se visualiza la creación de la topología lineal switches, hosts y sus enlaces. Elaboración propia.

## 4.3. REGLAS DE CONTROL DE ACCESO

Las reglas de control de acceso son las que van a permitir que los equipos remotos puedan tener conexión y visualización unos con otros o entre sí. Estas reglas se configurarán en base a la información de las direcciones MAC (Media

Access Control) mostradas anteriormente. Es fundamental dar permiso o denegar el acceso ya que, al momento de crear una topología, por defecto todos los equipos remotos tienen conexión entre ellos.

Estos son los puntos que no tendrán acceso en la topología ya creada:

- H1 no tendrá acceso a H3
- H3 no tendrá acceso a H1.
- H3 no tendrá acceso a H5.
- H5 no tendrá acceso a H3

Para proceder con las reglas de control de acceso antes mencionadas utilizamos el siguiente comando:

**FIGURA 4.4.** Comando para implementar las reglas de control de acceso.

```
floodlight@floodlight:~$ curl -X POST -d '{"src-ip":"10.0.0.3/32","dst-ip":"10.0
.0.1/32","action":"deny"}' http://localhost:8080/wm/acl/rules/json
```

**Nota:** En la figura anterior se puede apreciar cómo se está denegando el acceso a la IP 10.0.0.3 para que no pueda tener visualización con la IP 10.0.0.1. Elaboración propia.

A continuación, se utiliza el mismo comando para la creación de las reglas en la topología escogida:

**FIGURA 4.5.** Creación de las reglas de control de acceso.

```
floodlight@floodlight:~$ curl -X POST -d '{"src-ip":"10.0.0.3/32","dst-ip":"10.0
.0.1/32","action":"deny"}' http://localhost:8080/wm/acl/rules/json
floodlight@floodlight:~$ curl -X POST -d '{"src-ip":"10.0.0.3/32","dst-ip":"10.0
.0.5/32","action":"deny"}' http://localhost:8080/wm/acl/rules/json
```

**Nota:** Estas reglas dan o niegan permiso de enlace entre los dispositivos, en este caso está denegando el acceso en dos equipos. Elaboración propia.

**FIGURA 4.6.** Comando para visualizar las reglas de control de acceso.

```
curl http://localhost:8080/wm/acl/rules/json | python -mjson.tool
```

**Nota:** Este comando permite visualizar las reglas ya creadas. Elaboración propia.

# ANÁLISIS Y RESULTADOS

# CAPÍTULO IV

# 5. CAPÍTULO IV. ANÁLISIS Y RESULTADOS.

En este capítulo se realizarán todos análisis posibles incluyendo el de la entrevista que se llevó a cabo al momento de hacer este estudio, la simulación y las pruebas necesarias con todos los parámetros anteriores que dará como resultado el cumplimiento de la misión asignada.

## 5.1 ANALISIS E INTERPRETACION DE LA ENTREVISTA

Se entrevistó al Sr. José Catagua quien es técnico de sistemas y asesor DTI de la empresa “Eléctrico HAZ S.A.”, quien respondió algunas preguntas que se le hizo en base a la problemática que existe en la organización. Esta entrevista se hizo por correo electrónico ya que el Sr. José Catagua se encuentra al momento infectado con el virus Covid-19.

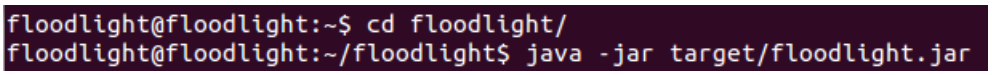
Cómo resultado de la entrevista realizada se obtuvieron los siguientes puntos importantes:

- La empresa esta gastando de más en equipos nuevos y mantenimiento de los mismos sin solucionar el problema que tiene con el trafico de datos y seguridad de su red.
- La red que maneja no es la adecuada para una empresa porque se enfoca más en una red doméstica.
- La dispersión que tienen en su arquitectura y con los dispositivos en su organización hace que existan grietas por donde se puede filtrar información valiosa de la empresa.
- Es necesario un cambio a SDN ya que solo con un switch OpenFlow manejado por un controlador les permitiría tener una seguridad completa y también simplificar operaciones y gastos recurrentes.

## 5.2. SIMULACIÓN DE SDN

Para comenzar con la simulación es necesario tener el sistema operativo en uso completamente actualizado, luego de eso se debe abrir una terminal en el sistema operativo que en este caso es Ubuntu y ejecutar el controlador que en este caso es Floodlight, mediante el siguiente comando:

**FIGURA 5.1.** Comando de ejecución del controlador Floodlight.




**Nota:** Este comando permite la ejecución de la interfaz gráfica por el puerto 8080. Elaboración propia.

En este análisis se va a demostrar como las SDN funcionan con control de acceso y sin control de acceso. Primero la SDN se configura sin establecer el control de acceso y luego se realizará otro análisis con el control de acceso ya establecido, de esta manera se va a poder diferenciar el comportamiento de la misma con cada una de las configuraciones.

En la primera configuración sin control de acceso, el resultado esperado es que los equipos remotos tengan conexión y visualización entre todos.

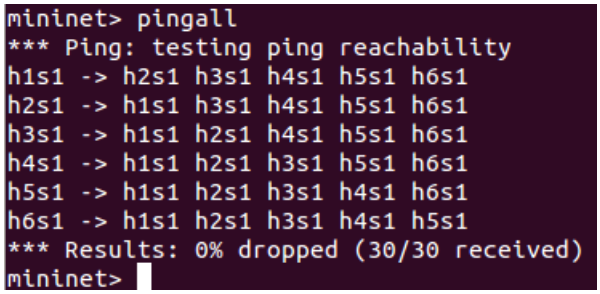
En la segunda configuración con control de acceso, el resultado esperado es que se bloquee la conectividad entre algunos de los equipos de acuerdo a los parámetros ya planteados en las reglas de control de acceso.

## 5.3. ANÁLISIS Y RESULTADOS SIN CONTROL DE ACCESO

Para demostrar la vulnerabilidad que existe al no configurar las reglas de control de acceso se empezará con una prueba en donde simplemente se pedirá el detalle de conexión de los equipos.

Esta prueba se efectuará sin habilitar las reglas de control de acceso, se realizará mediante el comando “pingall” ejecutado dentro del software MiniNet.

**FIGURA 5.2.** Prueba sin control de acceso.




**Nota:** En la figura se puede apreciar cómo todos los equipos tienen conectividad y visualización entre ellos. Elaboración propia.

Para que la topología de red creada funcione de manera más segura se tienen que establecer reglas de acceso, pues como se demostró en la figura anterior todos los equipos pueden visibilizarse, por lo tanto, también tener conexión entre ellos. Las compañías en general necesitan tener reglas de acceso, ya que existen áreas o departamentos en su organización donde se requieren restricciones.

No se debería dejar la configuración por defecto de estas reglas bajo ninguna circunstancia, porque se podría estar dejando una puerta abierta a que intrusos aparezcan y vulneren la información.

## 5.4. ANÁLISIS Y RESULTADOS CON CONTROL DE ACCESO

En este análisis se va a demostrar que, con las reglas de acceso previamente configuradas y habilitadas, a través del mismo comando utilizado en la prueba anterior, el cual es pingall, se puede notar como algunos equipos no tienen conexión y visibilidad con otros.

**FIGURA 5.3.** Comando que deshabilita el control de acceso.

```
floodlight@floodlight:~$ curl -X POST -d '{"src-ip":"10.0.0.3/32","dst-ip":"10.0
.0.1/32","action":"deny"}' http://localhost:8080/wm/acl/rules/json
floodlight@floodlight:~$ curl -X POST -d '{"src-ip":"10.0.0.3/32","dst-ip":"10.0
.0.5/32","action":"deny"}' http://localhost:8080/wm/acl/rules/json
```

**Nota:** Acceso deshabilitado por medio de comandos con reglas json. Elaboración Propia.

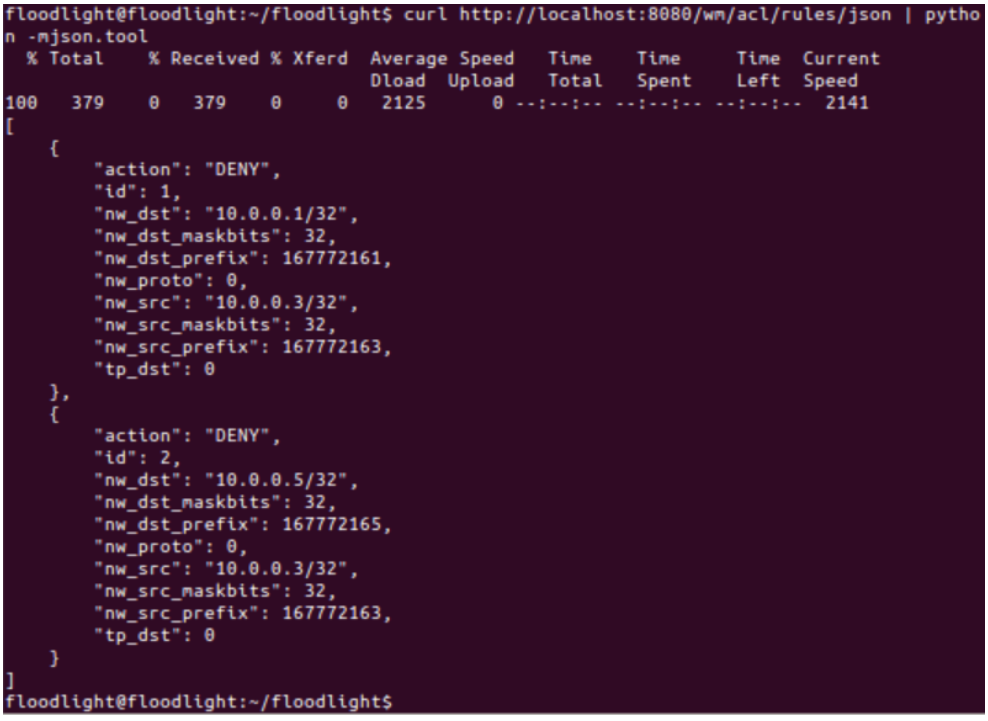
**FIGURA 5.4.** Prueba con control de acceso.

```
mininet> pingall
*** Ping: testing ping reachability
h1s1 -> h2s1 X h4s1 h5s1 h6s1
h2s1 -> h1s1 h3s1 h4s1 h5s1 h6s1
h3s1 -> X h2s1 h4s1 X h6s1
h4s1 -> h1s1 h2s1 h3s1 h5s1 h6s1
h5s1 -> h1s1 h2s1 X h4s1 h6s1
h6s1 -> h1s1 h2s1 h3s1 h4s1 h5s1
*** Results: 13% dropped (26/30 received)
mininet>
```

**Nota:** En esta figura se puede apreciar cómo H3 no tiene permisos con H1 y tampoco con H5. Elaboración propia.

Esta configuración de reglas evita que intrusos ingresen a los equipos, vulneren la información y tengan acceso para cambiar o modificar configuraciones en los equipos emulados. Los resultados obtenidos con estas pruebas son extremadamente favorables y eso es lo que se pretendía desde el comienzo de este estudio. Las pruebas de control de acceso demuestran lo necesarias que son para una organización. Se pudo apreciar con éxito y de una manera efectiva las configuraciones donde se les otorga o deniega permiso a los equipos emulados.

**FIGURA 5.5.** Comprobación de reglas de control de acceso.

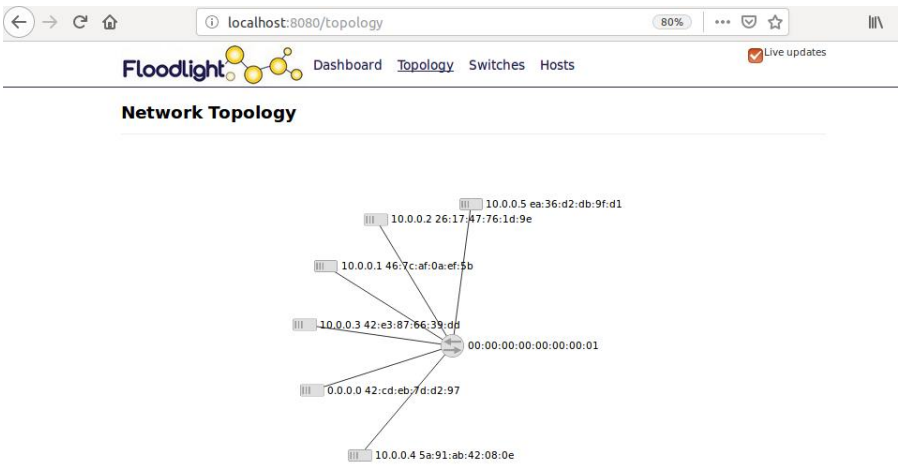



**Nota:** Comprobación de las reglas denegando el permiso a los hosts previamente mencionados. Elaboración propia.

## 5.5. TOPOLOGÍA GRÁFICA SDN EN FLOODLIGHT

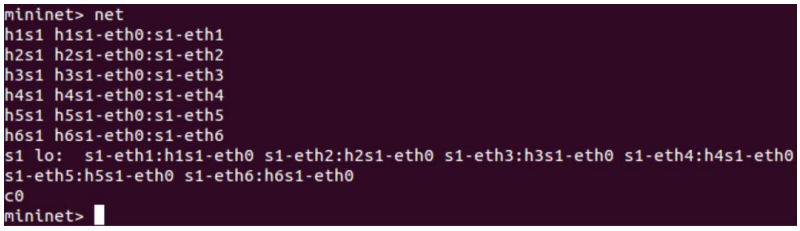
El controlador Floodlight a más de ser uno de los controladores más completos, ofrece la opción de poder visualizar la topología ya creada mediante su interfaz web. Para esto se tiene que ejecutar el explorador “Mozilla” que viene ya por defecto en el sistema operativo Ubuntu. En el apartado donde va la URL se introduce la siguiente dirección: localhost:8080/topology

**FIGURA 5.6.** Topología gráfica de la SDN.



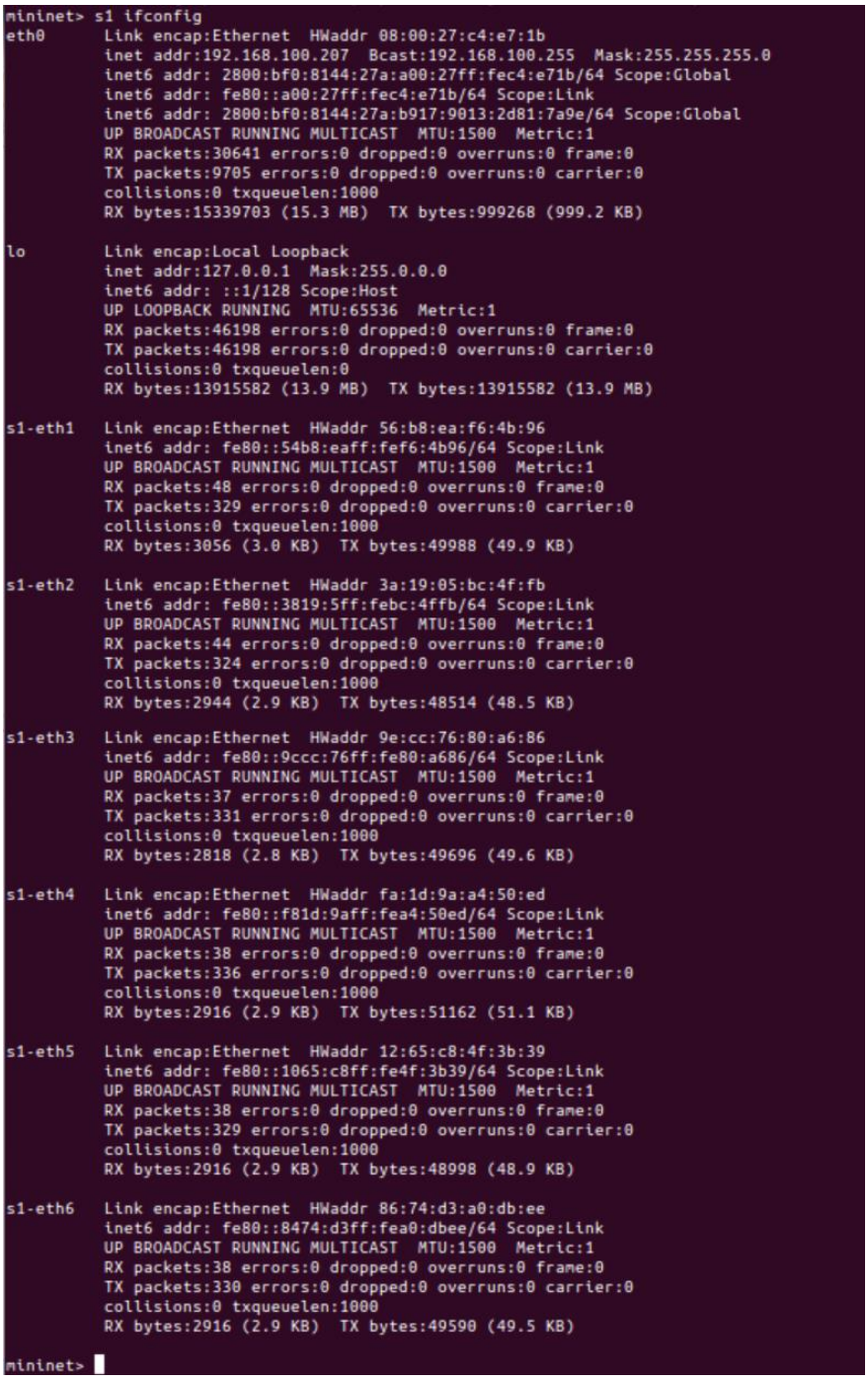

**Nota:** En esta figura se muestra la interfaz gráfica del controlador Floodlight con la topología de red ya creada. Elaboración propia.

**FIGURA 5.7.** Comprobación de los nodos enlazados.




**Nota:** El comando net muestra cuales son los nodos, que nodos están conectados y cuáles tienen conexión. Elaboración propia

**FIGURA 5.8.** Información total del Switch.

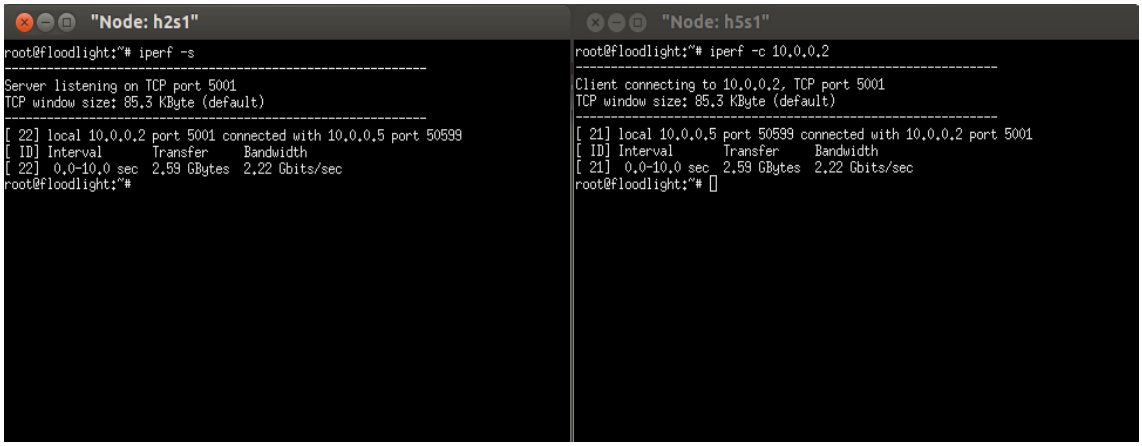



**Nota:** Esta figura muestra toda la información detallada del switch y sus enlaces. Elaboración propia.

## 5.6. ANÁLISIS Y RESULTADOS DEL ANCHO DE BANDA

Este análisis se va a enfocar en la transferencia de ancho de banda, para la realización de esta prueba primero se configuran los hosts dentro de la terminal de MiniNet. Se van a abrir dos terminales en Ubuntu, en este caso se va a utilizar el nodo H2 y el nodo H5 para demostrar el tiempo transcurrido y la cantidad de datos transferidos de un nodo a otro. Se utilizará el comando “iperf” el cual nos indica un diagnóstico de velocidad de la red y qué equipo está teniendo inconvenientes para lograr su rendimiento máximo.

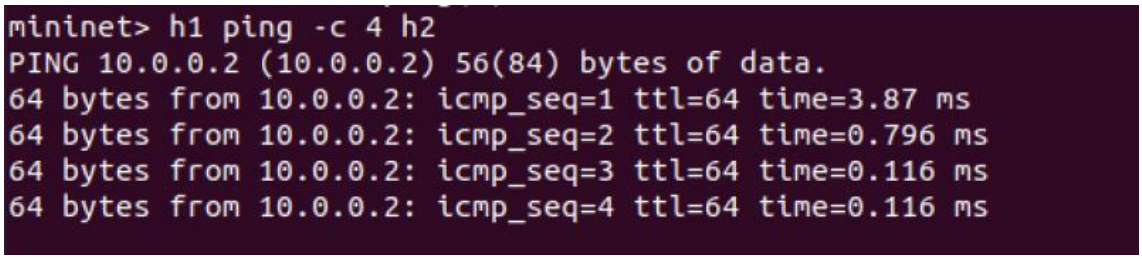
**FIGURA 5.9.** Prueba de ancho de banda.




**Nota:** En la primera ventana se ejecutará el comando “iperf -s” y en la segunda ventana para acceder al H5 se ejecutara “iperf -c 10.0.0.2” de esta manera se demuestra la transferencia que hay desde H2 a H5, los segundos transcurridos y la medición de ancho de banda. Elaboración propia.

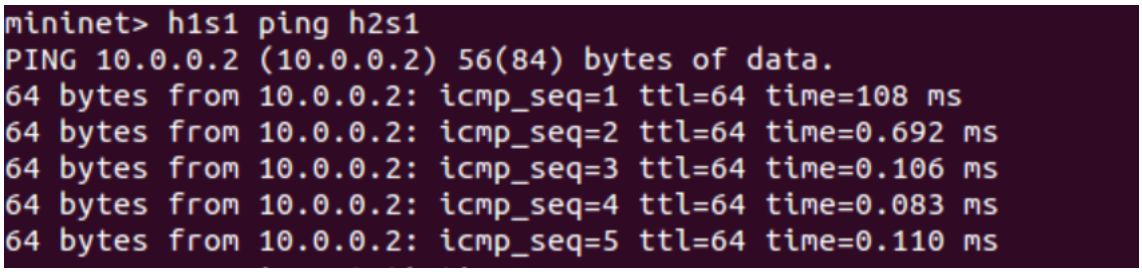
Para un análisis más detallado en la siguiente figura se muestra los segundos transcurridos en la transferencia de paquetes sin tener el controlador floodlight instalado.

**FIGURA 5.10.** Prueba de transferencia de paquetes sin Floodlight




**Nota:** Se muestra la velocidad de bytes en milisegundos desde el host1 al host2 sin tener activado el controlador Floodlight, mediante el comando ping ejecutado en MiniNet. Elaboración propia.

**FIGURA 5.11.** Prueba de transferencia de paquetes con Floodlight




**Nota:** Se muestra la velocidad de bytes en milisegundos desde el host1 al host2 ya con el controlador Floodlight activado y los segundos de transferencia que son menores en comparación a la imagen anterior. Elaboración propia.

El análisis realizado para medir el ancho de banda, velocidad y transferencia de paquetes de datos fue exitoso ya que, se visualiza cómo la distribución es más lenta sin tener la administración del controlador floodlight activa, por el contrario, al tener de base el controlador Floodlight y mediante el software MiniNet, este estableció por defecto la distribución de los paquetes de datos, dando como resultado más velocidad, rendimiento y optimización de recursos. De esta manera se define una nueva banda ancha y se potencia para así aumentar en gran medida la velocidad de transferencia.

# CONCLUSIONES

Para este estudio de propuesta tecnológica se llegó a la conclusión de que las SDN brindan posibilidades extraordinarias al momento de implementarlas en una organización.

La plataforma de virtualización Oracle VM VirtualBox tuvo el desempeño esperado al momento de la instalación de la máquina virtual con el controlador Floodlight y que a su vez integraba al software MiniNet.

Se pudo visualizar la reducción del tráfico de paquetes de datos en la red mediante el controlador que en este caso es Floodlight, el mismo que con la ayuda de MiniNet tomó la decisión de cómo distribuir el tráfico de datos de manera que esta no se sature.

De la misma manera Floodlight permite de forma sencilla la creación de reglas para la configuración del acceso de control, estas reglas fueron fundamentales en este estudio ya que las redes con arquitectura tradicional son muy vulnerables a intrusos.

Adicional a esto MiniNet es el software más recomendado para usuarios con poca experiencia ya que su administración es muy sencilla, les permite crear simulaciones de redes conmutadas con interfaces gráficas, esto le da al usuario una forma de entendimiento más simple sobre el manejo las funciones que tienen sus controladores y las tablas de flujo.

De acuerdo a todos los puntos anteriores se puede demostrar que mediante la implementación de las SDN se logra superar la saturación de la red local por el excesivo tráfico de paquetes, logrando optimizar el uso y amplitud del ancho de banda dentro de la organización ya que al aplicar las reglas de control de acceso que son utilizadas en el criterio de las aplicaciones logra mejorar la comunicación entre los dispositivos conectados.

# RECOMENDACIONES

- Utilizar como la plataforma de virtualización a Oracle VM Virtual Box, ya que su licencia es gratis y cuenta con herramientas que se utilizan en plataforma con licencias de pago, adicional soporta muy bien la máquina virtual basada en Linux como lo es Ubuntu.
- Escoger el controlador Floodlight para la realización de los análisis y pruebas necesarias porque brinda diversos módulos que funcionan muy bien con las SDN, permite configurar la seguridad máxima de los datos mediante el control de acceso de una manera sencilla, y tiene una interfaz gráfica muy útil.
- Usar MiniNet, en este caso ya viene integrado en el controlador Floodlight, este brinda facilidad de instalación y comandos básicos que pueden ser aplicados sin necesidad de tener experiencia previa con el software, es el más apto para poder hacer pruebas sencillas de banda ancha ya que MiniNet establece los parámetros de velocidad de la misma de acuerdo a la necesidad del usuario, de esta manera se potencia la velocidad en los procesos de datos sin importar la cantidad de banda ancha que tengan anteriormente.
- Continuar con más estudios sobre las redes definidas por software o SDN, ya que su implementación en organizaciones grandes o pequeñas son favorables, en ambas ahorra costos de instalación y servicio, adicional a esto simplifica procesos y aumenta el rendimiento con su administración centralizada.

## REFERENCIAS Y BIBLIOGRAFÍA


Bernal, I., & Mejía, D. (2016). *Las Redes Definidas por Software y los Desarrollos Sobre Esta Temática en la Escuela Politécnica Nacional.* Quito. Obtenido de https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/610/pdf

Carrillo Rodas, C. A. (2020). *Simulación de una red definida por software (SDN) para el control de acceso de los elementos de la red a nivel de capa 2.* Guayaquil. Obtenido de http://repositorio.ucsg.edu.ec/handle/3317/14837

Chafloque Mejía, J. D. (2018). *Propuesta de diseño de una red de datos de área local bajo la arquitectura de redes definidas por software para la Red Telemática de la Universidad Nacional Mayor de San Marcos.* Lima, Perú. Obtenido de https://hdl.handle.net/20.500.12672/10017

Digital Guide Ionos. (12 de junio de 2019). *Digital Guide Ionos*. Obtenido de https://www.ionos.es/digitalguide/servidores/know-how/software-defined-network/

Floodlight, P. (7 de Febrero de 2016). *Project Floodlight*. Obtenido de https://floodlight.atlassian.net/wiki/spaces/HOME/overview?mode=global

Huawei Enterprise. (13 de Enero de 2021). *Huawei Enterprise*. Obtenido de https://forum.huawei.com/enterprise/es/introducci%C3%B3n-al-protocolo-de-comunicaci%C3%B3n-netconf-utilizado-en-switches-cloudengine/thread/687241-100237

La Salle. (2020). *REDES TRADICIONALES VS SDN DEFINIDAS POR SOFTWARE.* Barcelona, España. Obtenido de https://blogs.salleurl.edu/es/redes-tradicionales-vs-sdn-definidas-por-software

López, S. C. (2019). *Estudio de redes SDN mediante Mininet y MiniEdit.* Valencia. Obtenido de http://hdl.handle.net/10251/127877

Marín Muro, Y. A. (2016). *Plataforma de pruebas para evaluar el desempeño de las redes definidas por software basadas en el protocolo Openflow.* Cuba. Obtenido de https://dspace.uclv.edu.cu/handle/123456789/7255

Mininet. (2021). *Mininet*. Obtenido de http://mininet.org/

ONOS. (2021). *ONF*. Obtenido de https://opennetworking.org/onos/

Open Daylight. (2021). *The linux foundation projects*. Obtenido de https://www.opendaylight.org/#more

Ortin Calabrese, E. (2016). *Medición y obtención del rendimiento y capacidades máximas del Switch HP 2920-24G OpenFlow mediante test y pruebas de estrés.* Argentina. Obtenido de https://rdu.iua.edu.ar/bitstream/123456789/1138/1/TESIS%20EMILIANO%20ORTIN%20CALABRESE.pdf

Osaba, M. N. (2016). *Virtualización en redes definidas por software.* Buenos Aires. Obtenido de http://ri.itba.edu.ar/handle/123456789/785

Profesional Review. (9 de Noviembre de 2018). *Profesional Review*. Obtenido de https://www.profesionalreview.com/2018/11/09/aplicaciones-virtualizacion/

Punt Informatic Becomit Company. (3 de Mayo de 2018). *Punt Informatic Becomit Company*. Obtenido de https://puntinformatic.com/beneficios-de-la-red-definida-por-software-sdn/

Redhat. (Sf). *Redhat*. Obtenido de https://www.redhat.com/es/topics/openstack

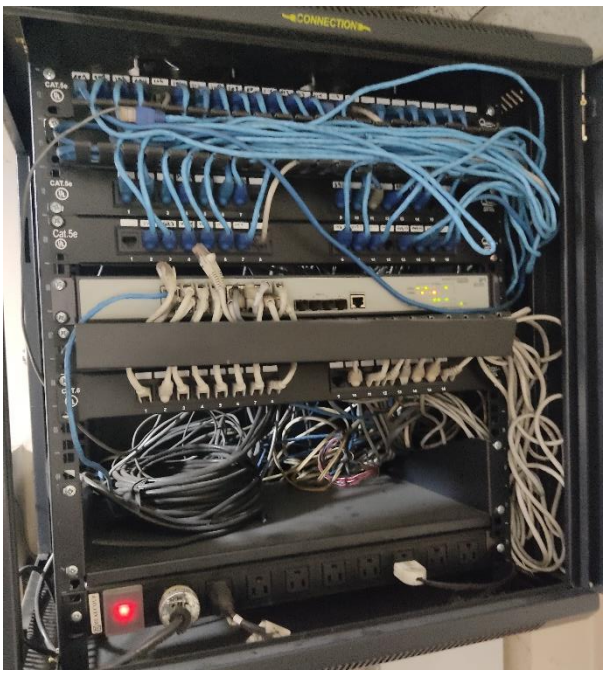
Rivoir, A. L., & Morales, M. J. (2019). *Tecnologías digitales Miradas críticas de la apropiación en América Latina.* Buenos Aires, Argentina. Obtenido de http://biblioteca.clacso.edu.ar/clacso/se/20191128031455/Tecnologias-digitales.pdf

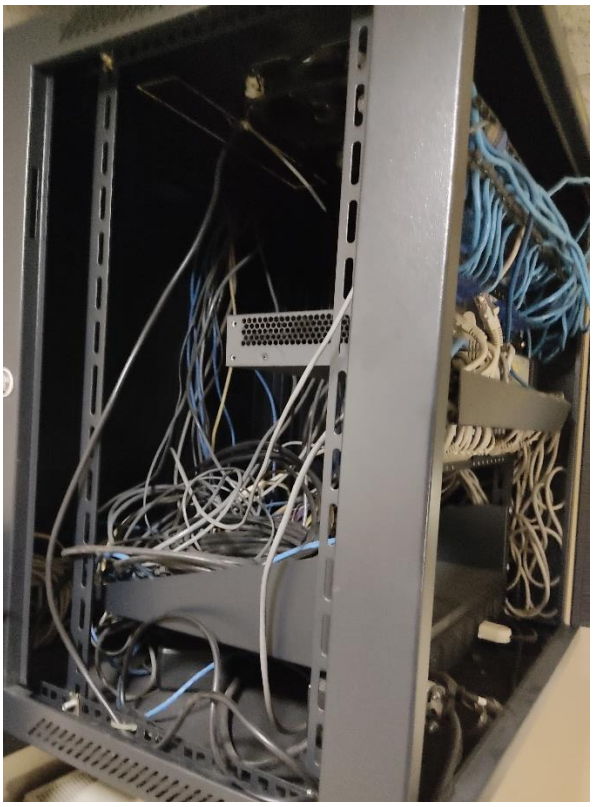
Sulca, I. C. (2018). *Redes definidas por Software (SDN).* Madrid. Obtenido de https://informatica.ucm.es/data/cont/media/www/pag-103596/transparencias/redes-por-software-SDN.pdf

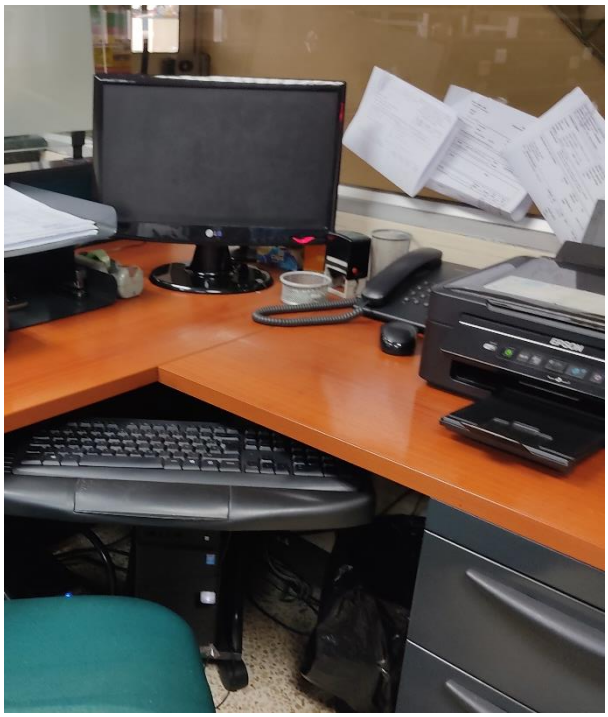
Velez Mejia, C. L. (2018). *ANÁLISIS DE SEGURIDAD EN REDES SDN (REDES DEFINIDAS POR SOFTWARE).* Medellín, Antioquia, Colombia. Obtenido de https://repository.unad.edu.co/bitstream/handle/10596/27165/%20clvelezm.pdf;jsessionid=AC076087A2AD439D3CBB0185760ED7E2.jvm1?sequence=1

Yagüe, C. (2019). *Qué es Apache Maven.* Obtenido de https://openwebinars.net/blog/que-es-apache-maven/

## ANEXOS





**ENTREVISTA.**

**¿Cuál es su función en la empresa y cuanto tiempo lleva trabajando en la misma?**

Mi función en la empresa es la de técnico de sistemas y asesor DTI, llevo trabajando en Eléctrico HAZ S.A. un periodo de 10 años.

**¿Cuáles son los dispositivos que conforman la red de la empresa Eléctrico HAZ S.A.?**

La red de la empresa es una red ethernet, muy simple, plana se podría decir, la cual se maneja sin un switch administrable, solo cuenta con uno común, tiene un servidor, proveedor de internet, conmutador para las líneas telefónicas, Router, y un promedio de 7 estaciones de trabajo.

**¿Cuál cree usted que es la problemática de la empresa?**

Yo creo que la problemática se enfoca al no tener ningún dispositivo de control, no dispone tampoco de una VPN ni una arquitectura para los hosts, los mismos que tienen una comunicación por dispersión en la que todos se comunican con todos y es lo que causa la saturación de la red, esto también hace que la seguridad de datos no sea la mas fuerte, al tener una red así igual queda una parte vulnerable a filtración de datos.

**¿Qué solución usted sugiere para solucionar la problemática de la empresa?**

Se podría implementar un firewall físico, pero el costo es elevado, así mismo como el mantenimiento y la implementación es tediosa, lo que puedo sugerir es un cambio a SDN, es más sencilla y práctica.

**¿Qué piensa usted sobre la funcionalidad actual de la empresa?**

Funciona de manera normal, pero también se puede mejorar, es bueno que se preste atención a las falencias de una organización, en este caso esta mejora ayudaría al rendimiento de la comunicación de la empresa ya que es lo primordial en las corporaciones actualmente.