



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Elaboración, recopilación, análisis y presentación de los  
KPIs de una red enmallada simulada en GNS3 por medio de  
herramientas de Excel, Python y Tableau.**

AUTOR:

Álvarez Granda, David Andrés

Trabajo de Integración Curricular previo a la obtención del título de  
**INGENIERO EN TELECOMUNICACIONES**

TUTOR:

Ms. C. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

16 de febrero del 2024



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**CERTIFICACIÓN**

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.  
**Álvarez Granda, David Andrés** como requerimiento para la obtención del  
título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

---

Ms. C. Palacios Meléndez, Edwin Fernando.

DIRECTOR DE CARRERA

---

Ms. C. Bohórquez Escobar, Celso Bayardo. Ph. D.

Guayaquil, a los 16 días del mes de febrero del año 2024



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Álvarez Granda, David Andrés**

**DECLARÓ QUE:**

El trabajo de titulación: **Elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3 por medio de herramientas de Excel, Python y Tableau**, previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 16 días del mes de febrero del año 2024

EL AUTOR

---

Álvarez Granda, David Andrés



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**AUTORIZACIÓN**

Yo, **Álvarez Granda, David Andrés**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **Elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3 por medio de herramientas de Excel, Python y Tableau**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 16 días del mes de febrero del año 2024

EL AUTOR

---

Álvarez Granda, David Andrés



# UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

## TRABAJO DE INTEGRACIÓN CURRICULAR INFORME SOFTWARE ANTIPLAGIO

### REPORTE DE COMPILATIO



Reporte Compilatio del trabajo de titulación de la Carrera Telecomunicaciones denominado: “Elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3 por medio de herramientas de Excel, Python y Tableau”, del estudiante Alvarez Granda, David Andrés se encuentra al 4% de coincidencias.

Atentamente,

Ing. Edwin Palacios, M. Sc

**DOCENTE-TUTOR**

---

## **DEDICATORIA**

Este trabajo de titulación se lo dedico principalmente a mi familia, quienes son un pilar fundamental en mi vida, mostrarles que puedo llegar a ser un profesional y un ser útil en la sociedad, porque ellos también forman parte indispensable al desarrollo de un profesional. Ustedes me han ofrecido confianza y motivación, lo cual me ha permitido mejorarme cada día más, y forjarme en sabiduría y conocimiento en el camino del aprendizaje.

Además, dedico este trabajo a todos los profesores que me han aportado sus conocimientos e ideologías para formarme como profesional en el ámbito académico y laboral. A mis amigos quienes siempre apoyándome y animándome a conseguir mis metas. También dedico este trabajo a todo investigador, científico y persona que desea aprender y mejorarse cada día más, juntos podemos cambiar este mundo volviéndolo más lógico y razonable, creando nuevas tecnologías y conceptos para aplicarlos en el desarrollo de la humanidad.

**EL AUTOR**

**ALVAREZ GRANDA, DAVID ANDRÉS**

## **AGRADECIMIENTOS**

En primer lugar, quiero agradecer a mi familia, mi mamita Sarita quién mi inculca el deseo del conocimiento, mi papá Rafael Marcelino quien me apoya afectivamente en mi vida, mi tío Carlos quién me fomentó la curiosidad por la carrera, mi papá Rafael Enrique el cual me ha apoyado en mis estudios y mi mascota Nate que es mi compañía en mis días, todos son mi familia la cual ha sido un pilar fundamental desde que tengo memoria de mi educación tanto académica como personal, dirigiéndome en mi camino hacia mi desarrollo como persona en la sociedad. Ellos han mostrado total apoyo y confianza en mí lo cual se evidenció a lo largo de los años, dándome su sabiduría y conocimiento para optar por buenas decisiones y poder resolver los problemas que se presenten.

También quiero agradecer a mis maestros los cuales siempre ha estado disponibles para ayudar o resolver alguna duda referente a la carrera, también prestos a ofrecerme su sabiduría y entendimiento para crear a un profesional. Un agradecimiento a mi tutor quien es mi mentor en el camino del desarrollo de este trabajo. A mis mejores amigos, a mi amigo Gilmar quienes me ha acompañado en este trayecto de aprendizaje y a mis seres queridos que me han apoyado a ser mejor.

EL AUTOR

ALVAREZ GRANDA, DAVID ANDRÉS



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_

**Ms. C. Bohórquez Escobar, Celso Bayardo. Ph. D.**  
DIRECTOR DE CARRERA

f. \_\_\_\_\_

**Ms. C. UBILLA GONZALEZ, RICARDO XAVIER**  
COORDINADOR DEL ÁREA

f. \_\_\_\_\_

**M. Sc. MEDINA MOREIRA, WASHIGTON ADOLFO. Ph. D.**  
OPONENTE



## Índice General

Resumen .....	XV
Abstract .....	XVI
Capítulo 1: Descripción General del Trabajo de Titulación.....	2
1.1. Introducción.....	2
1.2. Antecedentes .....	4
1.3. Definición del Problema.....	5
1.4. Justificación del Problema.....	5
1.5. Objetivos del Problema de Investigación.....	6
1.5.1. Objetivo General.....	6
1.5.2. Objetivos Específicos.....	6
1.6. Hipótesis. ....	6
1.7. Metodología de Investigación.....	6
Capítulo 2: Fundamentación Teórica .....	8
2.1. Introducción al networking.....	8
2.1.1. Capas del modelo OSI.....	8
2.1.2. Capas del modelo OSI y TCP/IP.....	10
2.1.3. Direcciones IP y MAC. ....	11
2.2. Métodos de subnetting. ....	12
2.2.1. Clases de direcciones IPv4.....	13
2.2.2. Subneteo IP.....	13
2.2.3. VLSM.....	14
2.3. Acceso a la red. ....	15
2.3.1. Acceso por medio de SSH.....	15
2.3.2. Acceso por medio de consola. ....	17
2.3.3. Ventajas del acceso por consola. ....	17

2.4.	Protocolo de enrutamiento OSPF.....	17
2.4.1.	Tipos de paquetes OSPF.....	19
2.4.2.	Tipos de base de datos en OSPF.....	21
2.4.3.	Estados operativos de OSPF.....	21
2.4.4.	ID de proceso y la wildcard.....	22
2.5.	Introducción al simulador GNS3.....	23
2.5.1.	Requerimientos Generales .....	24
2.5.2.	Hardware emulado .....	25
2.5.3.	Instalación de IOS router .....	25
2.5.4.	Ipterm .....	26
2.6.	Introducción básica a Python. ....	27
2.6.1.	Librería Netmiko. ....	27
2.6.2.	Librería Openpyxl. ....	28
2.6.3.	Librería Pandas. ....	29
2.6.4.	Librería Paramiko. ....	29
2.6.5.	Comandos básicos de Python. ....	30
2.7.	Introducción a PyCharm.....	31
2.7.1.	Entorno básico de PyCharm. ....	31
2.7.2.	Ventajas del programa PyCharm. ....	33
2.7.3.	Características del programa PyCharm. ....	33
2.8.	Introducción a Tableau.....	34
2.8.1.	Funcionalidades de Tableau.....	35
2.8.2.	Ventajas de Tableau.....	35
2.8.3.	Características de Tableau.....	36
	Capítulo 3: Diseño, Implementación y resultados .....	37
3.1.	Elaboración de la topología de la red enmallada en GNS3.....	37
3.1.1.	Diseño de la topología. ....	39

3.1.2.	Conexión de routers por OSPF.....	42
3.1.3.	Configuración de SSH. ....	42
3.2.	Diseño de escenarios de simulación en GNS3.....	43
3.2.1.	Escenario 1: Todos los routers con prioridad 1. ....	43
3.2.2.	Escenario 2: Envío de 10 Megabytes a las interfaces. ....	44
3.2.3.	Escenario 3: Ruta OSPF al apagar interfaces.....	47
3.3.	Extracción de KPIs por medio de Python. ....	53
3.3.1.	Script de KPIs de Memoria y Versión.....	53
3.3.2.	Script de KPIs de estado de interface. ....	57
3.3.3.	Script de KPIs de Chasis. ....	62
3.3.4.	Script de KPIs de protocolos de enrutamiento. ....	65
3.3.5.	Script de KPIs de neighbor OSPF.....	68
3.4.	Creación y análisis de dashboards con Tableau. ....	71
3.4.1.	Dashboard de KPIs de cpu y memoria.....	71
3.4.2.	Dashboards de KPIs de interfaces.....	73
3.4.3.	Dashboards de KPIs de Chasis. ....	81
3.4.4.	Dashboards de KPIs de route source. ....	83
3.4.5.	Dashboards de KPIs de OSPF neighbor.....	86
	Conclusiones.....	90
	Recomendaciones.....	90
	Bibliografía.....	92

## Índice de Figuras

Figura 1. Modelo OSI entre dos computadoras.....	9
Figura 2. Modelo de red TCP/IP. ....	11
Figura 3. División de una dirección IPv4.....	14
Figura 4. Interfaz de gráfica de PUTTY.....	16
Figura 5. Enrutamiento de una red por OSPF.....	18
Figura 6. Logotipo de GNS3. ....	24
Figura 7. Tarjetas de red del IOS de un router.....	26
Figura 8. Logotipo de Python.....	27
Figura 9. Conexión al router por medio de Netmiko.....	28
Figura 10. Logotipo del programa PyCharm.....	31
Figura 11. Entorno de trabajo en PyCharm.....	32
Figura 12. Logotipo de Tableau.....	35
Figura 13. Topología de la Red Enmallada en GNS3.....	41
Figura 14. Escenario 1 router DR/BDR.....	43
Figura 15. Diagrama de flujo del escenario 1.....	44
Figura 16. Escenario 2 envío de 10 Megabytes.....	45
Figura 17. Diagrama de flujo del escenario 2.....	46
Figura 18. Pruebas de interface Ipterm.....	47
Figura 19. Diagrama de flujo del escenario 3.....	47
Figura 20. Primera ruta definida por OSPF.....	48
Figura 21. Segunda ruta definida por OSPF.....	50
Figura 22. Tercera ruta definida por OSPF.....	52
Figura 23. Diagrama de flujo del script KPIs de memoria y cpu.....	53
Figura 24. Diagrama de flujo del script KPIs de estado de enlace.....	57
Figura 25. Diagrama de flujo del script KPIs de chasis.....	62
Figura 26. Diagrama de flujo del script de KPIs de protocolos de enrutamiento.....	65
Figura 27. Diagrama de flujo del script de KPIs de neighbor OSPF.....	68
Figura 28. Excel de los KPIs de memoria y cpu.....	71
Figura 29. Dashboard de Memoria y Modelo.....	72
Figura 30. Dashboard de %CPU usado y Uptime.....	73
Figura 31. Excel de los KPIs de %CPU usado y Uptime.....	74

Figura 32. Dashboard de MTU y Duplex.....	75
Figura 33. Dashboard de Ancho de Banda .....	76
Figura 34. Dashboard de Ruta 1 % Input y Output rate .....	77
Figura 35. Dashboard de Ruta 1 Número de Input Errors.....	77
Figura 36. Dashboard de Ruta 2 % Input y Output rate. ....	78
Figura 37. Dashboard de Ruta 2 Número de Input Errors.....	79
Figura 38. Dashboard de Ruta 3 % Input y Output rate .....	80
Figura 39. Dashboard de Ruta 3 Número de Input Errors.....	80
Figura 40. Excel de los KPIs de las Interfaces .....	81
Figura 41. Dashboard del Inventario de la Red.....	82
Figura 42. Excel de los KPIs de Route Source .....	83
Figura 43. Dashboard del Número de Routers vecinos.....	84
Figura 44. Dashboard del Número de Overhead .....	85
Figura 45. Dashboard del Número de Memoria .....	86
Figura 46. Excel de los KPIs de Router Neighbor. ....	86
Figura 47. Dashboard de Numero de Router Vecinos.....	88
Figura 48. Dashboard de Router DR/BDR. ....	88
Figura 49. Dashboard de Redes aprendidas por OSPF.....	89

## Índice de Tablas

Tabla 1. Clases de direcciones IPv4.....	13
Tabla 2. Tipos de mensajes LSA .....	20
Tabla 3. Tipos base de datos OSPF. ....	21
Tabla 4. Tipos estados operativos de OSPF.....	22
Tabla 5. Datos para redes del RTYGU01 .....	37
Tabla 6. Datos para redes del RTMLG01 .....	38
Tabla 7. Datos para redes del RTDUR01 .....	38
Tabla 8. Datos para redes del RTGYE01.....	38
Tabla 9: Datos para redes del RTUIO01 .....	38
Tabla 10. Datos para redes del RTCNC01 .....	39
Tabla 11. Datos para redes del RTRIO01 .....	39

## Resumen

El presente trabajo de titulación propone la elaboración de una red enmallada en GNS3, la cual está configurada con el protocolo de enrutamiento OSPF (protocolo de enrutamiento dinámico). Esta red se compone de 7 routers los cuales poseen el router ID de acuerdo con la IP de su interface Loopback, luego se extraen los KPIs del chasis, route source, ospf neighbors, interfaces, cpu y memoria entre otros.

Los KPIs varían de acuerdo con los tres escenarios propuestos y por medio de scripts de Python se genera un Excel, con el archivo de Excel se cargan los datos de las variables a Tableau donde se cambia formatos, calcula y se gestiona de acuerdo a su ubicación en los ejes correspondientes para elaborar los dashboards referentes a los KPIs mostrando gráficos de forma interactiva para el usuario consecutivamente se realiza el respectivo análisis de la red para obtener los resultados de los escenarios propuestos.

**Palabras claves:** GNS3, OSPF, KPIs, Python, Tableau, Scripts, Dashboards, Enrutamiento.

## **Abstract**

This degree work proposes the elaboration of a mesh network in GNS3, which is configured with the routing protocol OSPF (dynamic routing protocol). This network is composed of 7 routers which have the router ID according to the IP of its Loopback interface, then the KPIs of the chassis, route source, ospf neighbors, interfaces, cpu and memory and some others are extracted.

The KPIs are changed according to the three proposed scenarios and by means of Python scripts an Excel is then generated, and with the Excel file the data of the different variables are loaded to Tableau where the formats are changed, they are calculated and managed according to their location in the corresponding axes to elaborate the dashboards referring to the KPIs showing graphs in an interactive way for the user, consecutively the respective analysis of the network is performed to obtain the results of the proposed scenarios.

**Keywords:** GNS3, OSPF, KPIs, Python, Tableau, Scripts, Dashboards, Routing.



## **Capítulo 1: Descripción General del Trabajo de Titulación**

En este capítulo, se muestra la descripción general del proyecto de trabajo de integración curricular.

### **1.1. Introducción.**

Una red en malla se conforma por una serie de nodos que se comunican directamente entre sí, transmitiendo información desde nodos diversos hasta alcanzar su destino a través de múltiples saltos. En este tipo de red, no se requiere una entidad centralizada para supervisar su funcionamiento, lo que se denomina una operación distribuida. Si, en cambio, existe una entidad que gestiona las condiciones de operación de la red, se habla de una configuración centralizada. (Martinez & Kelsey, 2007).

Las redes enmalladas son un componente fundamental en el mundo de las comunicaciones y la infraestructura de tecnología de la información. Estas redes se caracterizan por tener una topología que conecta todos los dispositivos entre sí, lo que resulta en una estructura altamente redundante y tolerante a fallos. A medida que las organizaciones y empresas dependen cada vez más de la conectividad y la disponibilidad constante de servicios, las redes enmalladas han adquirido una importancia significativa.

Las redes en malla en telecomunicaciones ofrecen ventajas notables, como una alta redundancia y fiabilidad, ya que cada dispositivo está conectado directamente a otros, facilitando la continuidad de la comunicación incluso si un enlace falla. Esta topología proporciona una gran flexibilidad en la comunicación, permitiendo múltiples rutas para la transmisión de datos, lo que puede optimizar el rendimiento. La gestión y el mantenimiento de redes enmalladas pueden ser desafiantes debido a la cantidad de dispositivos interconectados. (Rico-Bautista et al., 2014).

En este contexto, el análisis de los Key Performance Indicators (KPI) se vuelve crucial. Los KPIs son métricas que evalúan el rendimiento y la eficiencia de las redes enmalladas, ofreciendo información valiosa sobre la

calidad del servicio, latencia, ancho de banda, utilización de recursos de red y otros aspectos clave. El monitoreo y la optimización de estos KPI son esenciales para garantizar un funcionamiento óptimo de la red y para tomar decisiones informadas.(Alemany et al., 2021).

Un software de simulación ampliamente utilizado en el análisis de redes enmalladas es GNS3 (Graphical Network Simulator 3). GNS3 es una herramienta poderosa que permite crear, configurar y probar redes en un entorno virtual. La simulación en GNS3 proporciona una plataforma segura para experimentar con configuraciones de red sin afectar la infraestructura real, lo que resulta beneficioso para la formación, la resolución de problemas y el diseño de redes complejas.(Grgurevic et al., 2021).

Los KPIs desempeñan un papel crucial en la evaluación de redes enmalladas dentro de GNS3, ya que permiten a los administradores y profesionales de TI analizar el rendimiento de la red simulada y ajustar sus configuraciones según sea necesario. La capacidad de medir y supervisar KPIs como el rendimiento de los enlaces, latencia, tasa de error y el flujo de tráfico en tiempo real es fundamental para garantizar que la red cumpla con los estándares de calidad y eficiencia esperados.(Kara et al., 2023).

En este contexto, es relevante explorar la importancia del análisis de KPIs en el contexto de las redes enmalladas y cómo GNS3, como una herramienta de simulación versátil, puede desempeñar un papel fundamental en la evaluación y optimización de estas redes. Este análisis es crucial para la toma de decisiones informadas y para garantizar que las redes enmalladas cumplan con los requisitos de rendimiento en un entorno de comunicaciones cada vez más complejo y exigente.(Nedyalkov & Georgiev, 2023)

El manejo y el entendimiento de los KPIs de una red es el modo más eficiente de analizar los recursos y el estado de una red, para ello es necesario representarlo de manera gráfica para un mayor entendimiento, por ende, es de gran relevancia el uso de las funciones de las herramientas de programación y de representación gráfica. (Dieudonne & Wang, 2023)

## 1.2. Antecedentes

Las telecomunicaciones en especial el área de networking son un campo amplio de la tecnología que evoluciona rápidamente. Por lo que es necesario tener diversas maneras para analizar y gestionar la red para un mejor desempeño de recursos. Las redes enmalladas es un modelo de topología muy esencial en las redes de datos, garantizando un mejor sistema redundante en el que cada nodo tiene conexión directa a otros nodos, evita que cuando caiga un nodo no afecte el desempeño de la red.(Bustamante & Hincapié, 2005).

El uso del simulador GNS3 es muy importante ya que permite crear, diseñar e implementar redes de una variedad de dispositivos tanto en software como el hardware configurando diversos aspectos para obtener un ambiente más realista, para un mejor análisis de los posibles casos que puedan suceder, otorgando múltiples herramientas por su diversidad al configurar y usar elementos de una red, además ofrece el uso de máquinas virtuales y al usuario cargar el IOS(sistema operativo de los routers Cisco) en el dispositivo.(Escobar León & Pineda Suárez, 2022).

En las redes enmalladas existen diversos KPIs los cuales proporcionan información muy importante a los gestores de la red, por lo que su entendimiento es relevante al momento de realizar el análisis de una red, dando herramientas para optimizar recursos y la toma de decisiones en base a resultados obtenidos al extraer información, obteniendo información de forma más detalla y especifica con mejores resultados para el rendimiento de la red.(López-Carmona et al., 2023)

En la actualidad, los lenguajes de programación juegan un papel fundamental en el desarrollo de tecnología, uno de los que más resaltan es Python (lenguaje de programación de alto nivel), el cuales ofrece a su usuario múltiples herramientas, librerías y características las cuales son útiles al momento de crear aplicaciones en diversos ámbitos. Python en redes de telecomunicaciones se ha integrado de manera importante por lo que los gestores de la red pueden utilizarlo para automatizar la gestión de la red como

también extraer información de diversos equipos al mismo tiempo en la red aplicada.(González Lattuada & Maggiolo Rocca, 2017).

### **1.3. Definición del Problema.**

En la actualidad, existe una carencia de herramientas y metodologías efectivas para la evaluación del rendimiento de redes enmalladas mediante KPIs. Aunque la simulación de redes enmalladas, especialmente a través de herramientas como GNS3, ofrece una solución potencial, la gestión eficaz de estas simulaciones es un desafío crítico. Además, la representación de los KPIs generados por estas simulaciones en GNS3 plantea dificultades adicionales, lo que dificulta la optimización del rendimiento de la red enmalla.

Por lo tanto, surge la pregunta fundamental: ¿De qué manera se utilizaría herramientas de programación y diseño de gráficos para representar los KPIs de una red enmallada simulada en GNS3 superando las limitaciones actuales en la gestión y visualización de datos de rendimiento en entornos simulados?

### **1.4. Justificación del Problema.**

En la actualidad la tecnología ha evolucionado en gran escala sobre todo en área de networking, por ello la simulación de redes en GNS3 es crucial para la planificación y gestión de redes de comunicaciones. Sin embargo, la falta de representación gráfica de los KPIs los cuales son los indicadores claves de rendimiento dificulta a los gestores de la red el monitoreo en tiempo real, la optimización de recursos y la toma de decisiones informadas.

En el proyecto propuesto, con el uso de diversas herramientas de programación y de presentación gráfica se podrá obtener una visualización más clara y detallada de los indicadores de rendimiento, dando una comprensión más concisa a los gestores de la red para una mejor toma de decisión o gestión de la red que se desea extraer.

El uso de las funciones de la herramienta de programación Python permitirá extraer en un archivo de Excel los KPIs de la red enmallada simulada en GNS3, para posteriormente procesar la información de manera gráfica y

simplificada por medio de Tableau (software para análisis de datos), el cual que es una manera más eficiente de presentar la información. Por ello el proyecto contará con procesos los cuáles se ejecutarán de manera ordenada para una aplicación óptima y efectiva.

## **1.5. Objetivos del Problema de Investigación.**

### **1.5.1. Objetivo General.**

Desarrollar un proceso integral para la elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3, haciendo uso eficiente de herramientas como Excel, Python y Tableau.

### **1.5.2. Objetivos Específicos.**

1. Describir los fundamentos generales del networking, indicadores KPIs y herramientas de simulación.
2. Diseñar los escenarios de simulación de una red enmallada para la obtención de las métricas KPIs utilizando el simulador GNS3.
3. Recopilar datos generados en la simulación de la red por medio de scripts de Python.
4. Generar los datos recopilados en documentos de Excel con los KPIs de cada uno de los equipos para su presentación en Tableau.
5. Analizar los resultados obtenidos del estado de la red tanto físico como lógico.

## **1.6. Hipótesis.**

La implementación de un programa en Python permitirá extraer los KPIs de la una red enmallada en un archivo de Excel, consecuentemente se procesará los datos en Tableau para una presentación más eficiente del estado de una red.

## **1.7. Metodología de Investigación.**

En el presente trabajo de investigación se va a elaborar una red enmallada en GNS3, la cual se extraerán los KPIs por medio del lenguaje de programación Python, el cual a su vez genera un archivo de Excel con los datos recopilados de las métricas solicitadas de manera ordenada y detallada,

por consiguiente, se subirán los datos a Tableau donde se obtendrá de manera gráfica los datos ingresados, dando como fin obtener un mejor entendimiento y presentación del estado de la red para la presentación de los resultados obtenidos al momento de la recopilación.

La cronología del presente trabajo de investigación es de carácter progresivo porque se elaboró, recopiló, analizó y presentó los resultados obtenidos de los KPIs de una red enmallada en el simulador GNS3 de manera ordenada y detallada, para obtener los resultados y representarlos de manera gráfica, para un mejor análisis del estado de una red de una manera más completa y concisa.

## **Capítulo 2: Fundamentación Teórica**

### **2.1. Introducción al networking.**

En la actualidad el networking es el tejido que conecta todo el mundo digital, permitiendo aprovechar al máximo la revolución tecnológica en la que se vive. Es una disciplina que se encuentra en constante evolución que sigue transformando la forma de vida en lo personal, laboral y de comunicación de los usuarios a nivel mundial.

En un mundo cada vez más interconectado, el networking se ha convertido en un elemento esencial para la comunicación y la colaboración, en forma de redes locales en hogares u oficinas, o en la vasta red global que es Internet, el networking permite que los dispositivos electrónicos se comuniquen entre sí y accedan a una amplia variedad de servicios y recursos.(Jimenez et al., 2023).

El networking implica la capacidad de compartir recursos, dispositivos, información y programas tanto local como globalmente. Además, se centra en garantizar la fiabilidad de los datos mediante la disponibilidad de opciones de almacenamiento redundante. La búsqueda de una relación costo/beneficio favorable es esencial en este contexto. Además, el objetivo principal es permitir la rápida y eficiente transmisión de información entre usuarios que se encuentran distantes geográficamente.(Darín, 2016).

#### **2.1.1. Capas del modelo OSI.**

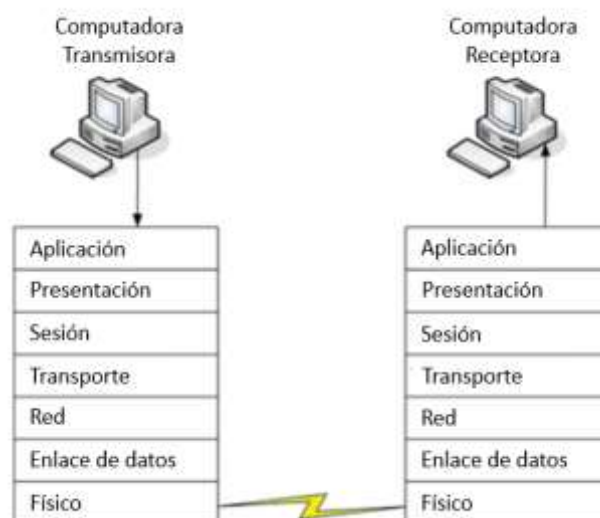
El modelo OSI (Open Systems Interconnection) fue elaborado por la Organización Internacional para la Estandarización, que tiene como función principal admitir que múltiples sistemas de comunicación puedan interconectarse por medio de protocolos, para el correcto funcionamiento de la red creada.(Hernández et al., 2017).

El modelo OSI está compuesto de 7 capas las cuales tiene su propio funcionamiento y servicio que presta a cada proceso de la red establecida, estas capas son:

- La capa de aplicación: Ofrece los servicios que las aplicaciones utilizan para permitir la comunicación entre usuarios a través de la red, permitiendo una mayor integración de funciones.
- La capa de presentación: Establece el formato de los datos que se intercambian entre las aplicaciones, proporcionando una serie de servicios para procesar información.
- La capa de sesión: Proporciona los mecanismos de gestión de interacción entre aplicaciones, ofrece inicio y fin de sesión.
- La capa de transporte: Permite la segmentación de paquetes para un mejor envío de mensajes. Aquí se encuentra el protocolo UDP que no es orientado a la conexión y el TCP que si es orientado a la conexión, los cuales son importantes en una red. (Kamble & Kotwal, 2023).
- La capa de red: Define la ruta que seguirán los datos desde su origen hasta su destino por medio redes interconectadas mediante routers.
- La capa de enlace de datos: Maneja el direccionamiento físico en cualquier topología de red, es decir mantiene y activa la conexión.
- La capa física: Controla las señales (medio) a través de las cuales viajan los datos en la red.

Es importante el conocimiento de estas capas para el entendimiento de los protocolos y los servicios de un sistema de comunicación. En la Figura 1 se puede observar las capas del modelo OSI entre 2 computadoras.

Figura 1. Modelo OSI entre dos computadoras



Fuente: (Panek, 2020)



### **2.1.2. Capas del modelo TCP/IP.**

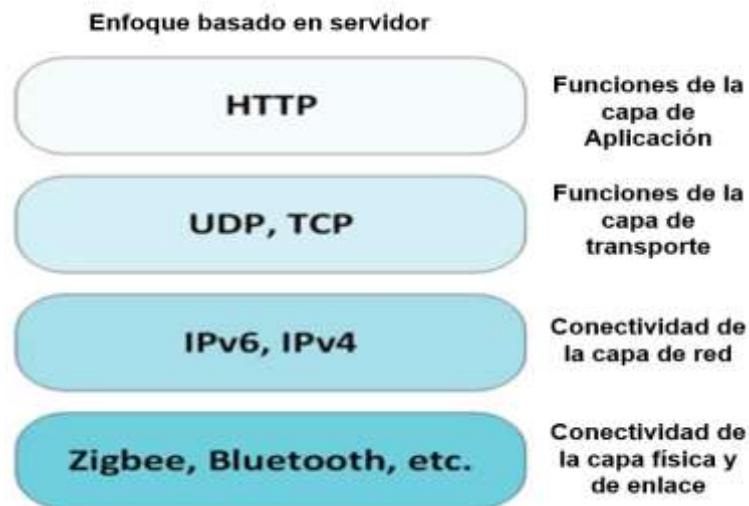
El modelo TCP/IP hace referencia al protocolo de capa de transporte TCP (Transmission Control Protocol) y el protocolo de red IP (Internet Protocol). Este modelo accede al intercambio de datos en una red, de acuerdo al protocolo que esté ubicado en dicha red, fue creado a partir del proyecto ARPANET que luego se convirtió en Internet. (Oviedo et al., 2018).

A diferencia del modelo OSI el modelo TCP/IP consta de 4 capas ubicadas jerárquicamente y más utilizado para la práctica laboral o el desarrollo de nuevas redes. Estas capas son:

- Capa de acceso a la red: Proporciona acceso a la red de forma física, define el tipo de datos que se van a enrutar, la manera en que se va a ejecutar el enrutamiento, chequeo de errores, etc. Es lo equivalente a la capa física y de enlaces de datos del modelo OSI.
- Capa de red: Al igual que la capa 3 del modelo OSI, esta capa se encarga de gestionar direcciones IP y de enrutar los paquetes desde el origen al destino por medio de la red. Esta capa es crucial porque abarca datos fundamentales para la conexión a la red, se utilizan protocolos de enrutamiento como OSPF o BGP, y direcciones IPv4 e IPv6 que se asignan a cada host. (Tyagi, 2020)
- Capa de transporte: Similar a la capa 4 del modelo OSI, encargado de designar puertos asociado a aplicaciones, ofrecer información del estado de la conexión y datos de enrutamiento. Por medio de los protocolos TCP y UDP para asegurar la transmisión de datos desde el origen hacia el destino.
- Capa de aplicación: Ofrece aplicaciones de red como TELNET, FTP para la comunicación con capas inferiores por medio de protocolos de TCP y UDP. El usuario interactúa con la red para el acceso a servicios como navegación web, transferencia de archivos, etc.

Es importante conocer cómo funcionan las capas del modelo TCP/IP jerárquicamente para un mejor análisis de la red con sus respectivas funciones y protocolos. En la Figura 2 se observa las capas del modelo TCP/IP relacionadas con su protocolo respectivo.

Figura 2. Modelo de red TCP/IP.



Fuente: (Lombardi et al., 2021)

### 2.1.3. Direcciones IP y MAC.

La dirección IP es la dirección o ID digital que se otorga a un dispositivo informático para que pueda comunicarse con otros equipos. Evidentemente, para conseguir la dirección IP del ordenador debe tener una tarjeta de red o llamada NIC (Network Interface card. Existen dos tipos de direcciones IP que son IPv4 e IPv6.(Siahaan, 2017).

La dirección IPv4 se compone de 4 bloques de números decimales cuyo valor tiene un rango entre 0 y 255, es una dirección de 32 bits que se reparten como 4 octetos. La dirección IPv6 es una dirección que tiene 128 bits, se divide ocho bloques de 16 bits cada uno, se representa de forma hexadecimal. Cada dirección IPv4 viene en conjunto con su máscara, y en IPv6 se utiliza los prefijos. Algunos ejemplos de direcciones IPv4 e IPv6 con su máscara y prefijos correspondientes son:

- Para IPv4: 192.168.1.2 255.255.255.0
- Para IPV6: 2001:0db8:85a3:0000:0000:8a2e:0370:7334/64

Los dispositivos de comunicaciones tienen un identificador único llamado dirección MAC (Media Access Control). La dirección MAC identifica el dispositivo como la fuente de una trama transmitida y permite el procesamiento de la trama a su llegada. Una dirección MAC se compone de 6 bytes, la IEEE RA asigna un bloque de direcciones Organizationally Unique

Identifiers (OUIs), que consta de 3 bytes, estos bloques se venden a fabricantes específicos. Posteriormente, cada fabricante asigna los 3 bytes restantes para garantizar la unicidad global de cada controlador de interfaz de red (NIC) que fabrica. (Teca & Natkaniec, 2023).

El formato de la dirección MAC se estructura de la siguiente manera: los 3 primeros bytes representan el identificador único de la organización (OUI), mientras que los 3 últimos bytes denotan el controlador de interfaz de red (NIC). Una dirección MAC se compone de 6 grupos cada uno con 2 valores expresados en hexadecimal, cada dispositivo consta con una NIC o también llamada tarjeta de interfaz de red que contiene una dirección MAC la cual es única entre todos los dispositivos.

## **2.2. Métodos de subnetting.**

El subnetting es esencial en redes, implica dividir una red IP en subredes más pequeñas para gestionar direcciones eficientemente y reforzar la seguridad junto con una mejor distribución de recursos. Mediante el ajuste de la máscara de subred y la variación de la longitud del prefijo, se pueden destinar partes específicas de las direcciones a subredes concretas.

El proceso de subnetear mejora la eficiencia en la asignación de direcciones, optimizando el rendimiento y facilitando la gestión de la red. Además, el subnetting permite implementar políticas de seguridad más detalladas al restringir la comunicación entre subredes, fortaleciendo la protección y la confiabilidad al usuario de su privacidad. (Putri & Mogi, 2020).

El subnetting proporciona flexibilidad, eficiencia y seguridad en la administración de direcciones IP y recursos de red en entornos complejos, siendo una práctica esencial con gran relevancia en el diseño y gestión de redes, dando la opción de crear parámetros que definen la calidad de una red como la calidad de servicio, control del flujo del tráfico, lista de control de acceso entre otros. Las dos maneras de realizar subnetting en IPv4 es el Subneteo IP y el VLSM.(Kingsley, 2023).

### 2.2.1. Clases de direcciones IPv4.

Las direcciones IPv4 se dividen en cinco clases: A, B, C, D y E. Las clases A, B y C son las más utilizadas y están destinadas a redes específicas, las cuales están en función del tamaño. Las clases D y E están reservadas para el uso en casos especiales como la transmisión de multidifusión y fines experimentales. La asignación de direcciones IP en clases proporciona un marco para la administración y gestión de direcciones en red. (Gonzalez-Ramos et al., 2019).

Es importante conocer el rango de cada clase de las direcciones IPv4 para optar por la dirección adecuada al diseñar y administrar una red, y optimizar redes que pueden ser ocupadas a futuro. En la Tabla 2.1 se visualiza las clases de direcciones IPv4 con sus respectivos números de redes y host.

Tabla 1. Clases de direcciones IPv4.

Clase de IPv4	Primeros bits	Primer octeto	Número de redes	Host por red
Clase A	0	1-126	126	16777214
Clase B	10	128-191	16384	65534
Clase C	110	192-223	2097152	254
Clase D	1110	224-239	n/a	n/a
Clase E	11110	240-254	n/a	n/A

Fuente: (Sisat et al., 2013)

### 2.2.2. Subneteo IP.

El Subneteo IP es muy relevante para entender cómo se reparte una red más grande en subredes las cuáles tienen su sección del sistema autónomo. En el subnetting se utiliza la máscara de red que indica la porción de la red en una dirección IP. (Kumar & Shinde, 2016)

Para realizar un Subneteo IP se debe realizar un procedimiento para un aumento en la eficiencia de la gestión de la red:

- Primero se observa que cada octeto contiene números del 0 al 255, que se representa en binario.
- Se analiza la IP de la red con la máscara que mostrará el número de host disponibles. En este ejemplo se muestra una IP de red:

10.10.10.10/8

192.168.1.0 255.0.0.0

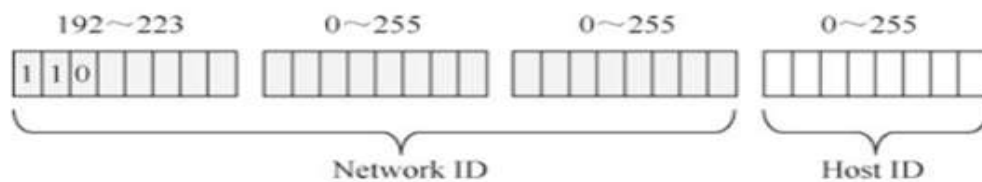
- Analizar cuantas subredes se requieren implementar. Usando el ejemplo anterior se necesita implementar siete subredes por ende buscamos un número aproximado a la cantidad de subnets necesarias, este número debe ser el exponente que elevamos el dos.
- Se observa que el número más cercano a 7 es  $2^3$ . Es decir, existirán 8 subredes. Se toma 3 bits de la porción de host de la siguiente manera:

$$11111111.00000000.00000000.00000000 = /8$$

$$11111111.11100000.00000000.00000000 = /11 \text{ expresado como } 255.224.0.0$$

- Asignar la IP de red de cada subred con su respectivas máscara para conocer su distribución. En la Figura 3 se observa la división de una dirección IPv4 en parte de red y host, especificando su parte red y host.

Figura 3. División de una dirección IPv4.



Fuente: (Co & Huawei Technologies Co., Ltd., 2022)

### 2.2.3. VLSM.

La máscara de subred de longitud variable, conocida como VLSM (Variable Length Subnet Mask), que consiste en especificar una máscara de subred diferente, para el mismo número de red en una subred distinta. La VLSM divide el espacio de direcciones IP en una jerarquía de subredes de distintos tamaños, permitiendo así crear subredes con un número de hosts muy diferente sin desperdiciar un gran número de direcciones.

Mediante el método VLSM, los gestores o administradores de red son capaces de utilizar un protocolo de enrutamiento sin clase (dirigen el tráfico sin dividir las direcciones IPs por clase) En pocas palabras, el VLSM es una agrupación de varias subredes que se encuentran en una gran red de direcciones.(Basira et al., 2015).

Para calcular las subredes dentro de una red por medio del método de VLSM es de gran relevancia aplicar de la siguiente manera para un desarrollo óptimo de la distribución de la red:

- Observar cual es la dirección IP de la red con su respectiva máscara, la cual es un dato importante para su pertinente división. Luego se analiza cuantos hosts requiere cada subred, y se las ordena de mayor a menor. En este ejemplo se quiere dividir una red máscara 24 en 3 subredes de 120, 60, 30 correspondientes.(Tamrin et al., 2023)

162.168.1.0/24 → Red<sub>1</sub> = 120 host Red<sub>2</sub> = 60 host Red<sub>3</sub> = 30 host

- Analizar el total de host disponibles en la red principal, como es máscara 24 tiene una capacidad de 256 direcciones IP. Para calcular que máscara es necesario para cada subred utilizar los bits prestados en relación por la porción de host.

Red = 192.168.1.0 → Máscara = 11111111.11111111.11111111.00000000

Red<sub>1</sub> = 120 host → Máscara = 11111111.11111111.11111111.10000000

Red<sub>2</sub> = 60 host → Máscara = 11111111.11111111.11111111.11000000

Red<sub>3</sub> = 120 host → Máscara = 11111111.11111111.11111111.11100000

- Convertir las máscaras obtenidas de formato binario a decimal, consecuentemente determinar la dirección de la subred que es la primera dirección IP y la última está destinada a broadcast, dando como resultado el rango de direcciones IP dentro de cada subred.

### **2.3. Acceso a la red.**

El acceso a un dispositivo de red, como un router, es esencial para la administración y configuración eficiente de la infraestructura de una red. Dos de los métodos más comunes para interactuar con un router son a través de la interfaz de línea de comandos (CLI) por consola y mediante protocolos de acceso de forma remota. Estos métodos proporcionan a los administradores la capacidad de realizar tareas críticas, como la configuración de parámetros de red, gestión de políticas de seguridad, gestión de QoS.(Song et al., 2023).

#### **2.3.1. Acceso por medio de SSH.**

Es importante configurar correctamente los routers, ya que esto ayudará a prevenir ataques y a mantener la seguridad y confidencialidad del tráfico de red. Utilizar el protocolo Telnet para acceder remotamente a un router no es suficientemente seguro. Por lo tanto, es muy recomendable ingresar por

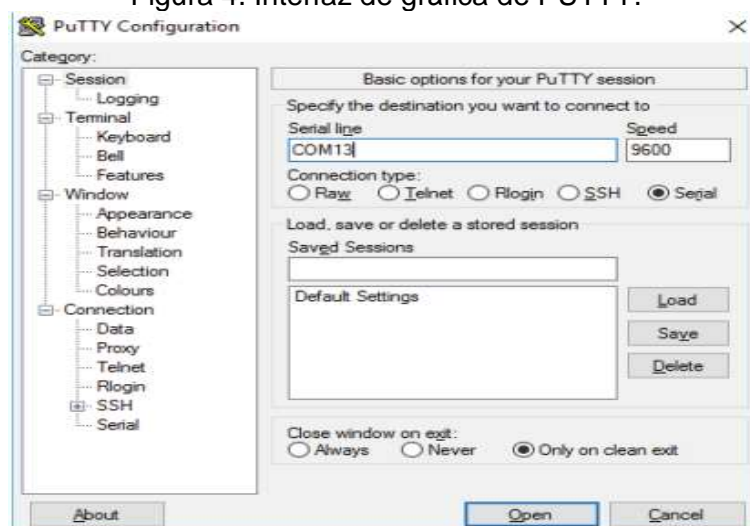
medio de SSH (Secure Shell) para acceder al router de forma segura y confiable.(Lau, 2021).

SSH es un protocolo de red que proporciona a los administradores una forma segura de acceder a un ordenador remoto. El protocolo SSH provee una autenticación y encriptación sólidas, y preserva la confidencialidad e integridad de las comunicaciones. Los administradores de red recurren con frecuencia a SSH para administrar remotamente la red, ejecutar comandos y mover archivos de un ordenador a otro.(Ylonen et al., 2015).

Para las conexiones remotas a los equipos es necesario utilizar emuladores los cuales permiten acceder con la dirección IP de la interfaz de un equipo o la dirección IP loopback. PuTTY es un emulador de terminal y cliente SSH de código abierto ampliamente utilizado en entornos de administración de sistemas y redes.

Su versatilidad en el soporte de varios protocolos de red, incluyendo SSH, Telnet. PuTTY facilita la conexión segura a sistemas remotos a través de su cliente SSH integrado, cifrando la comunicación para garantizar la confidencialidad de los datos. Su portabilidad, capacidad para guardar sesiones configuradas y herramientas adicionales para transferencia segura. En la Figura 4 se observa el entorno de trabajo de un emulador PUTTY para la conexión de dispositivos.

Figura 4. Interfaz de gráfica de PUTTY.



Fuente: (Guamán, 2017).

### **2.3.2. Acceso por medio de consola.**

El acceso por consola es un método esencial para configurar dispositivos de red, proporcionando una conexión directa mediante un puerto de consola físico y un cable serial. Se encuentra comúnmente en entornos empresariales, implica el uso de software de emulación de terminal para establecer una interfaz de línea de comandos (CLI) con el dispositivo. A través de esta conexión, los administradores pueden ingresar comandos específicos para la configuración y el monitoreo detallado del equipo, lo que es particularmente útil en situaciones de resolución de problemas y configuración inicial. La seguridad física del acceso por consola es crítica, ya que garantiza la integridad de la conexión directa al dispositivo y previene manipulaciones no autorizadas.(Nuñez, 2013).

### **2.3.3. Ventajas del acceso por consola.**

El acceso por consola ofrece ventajas como la capacidad de resolver problemas directamente y acceder a configuraciones avanzadas, aunque su uso puede limitarse una vez que la red está operativa, prefiriendo configuraciones remotas a través de interfaces de red para la gestión continua, para este tipo de acceso es necesario utilizar un cable de consola el cual se conecta a un puerto de consola para la conexión directa al dispositivo, por ello también es necesario un control presencial, lo que disminuye la posibilidad del ingreso de algún intruso que quiera ingresar a la red por medio protocolos de acceso.(Cañizares & Alejandro, 2020).

## **2.4. Protocolo de enrutamiento OSPF.**

El protocolo de enrutamiento Open Shortest Path First o llamado en sus siglas OPSF fue creado por el Grupo de Trabajo de Ingeniería de Internet (IETF) para redes de IP. OSPF es un protocolo estándar abierto fundamentado en el protocolo de enrutamiento de estado de enlace, el cuál se centra en escoger el camino mas corto para llegar al destino.

En el protocolo de enrutamiento de estado de enlace, cada enrutador de la red que funciona con OSPF conserva la misma topología denominada base de datos de estado de enlace. Los enrutadores emplean la base de datos de



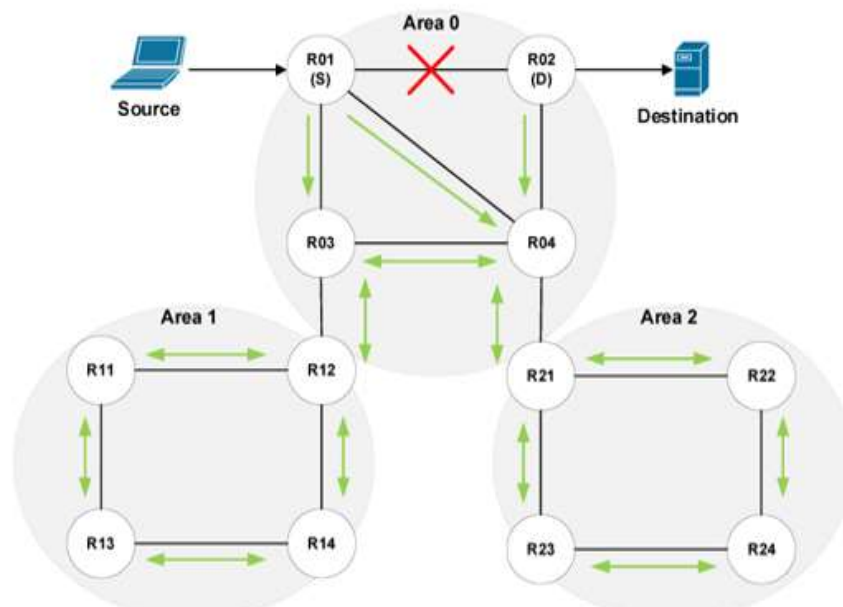
estado de enlace para determinar el trayecto más corto a cada red de destino utilizando el algoritmo SPF/Dijkstra, por ende, se le denomina a OSPF un protocolo de enrutamiento dinámico, este es muy utilizado en redes más complejas.(Mishra & Sahoo, 2020).

En el protocolo OSPF también llamado protocolo de estado de enlace abierto, los routers se agrupan en áreas. Según el tamaño de la red, puede existir una o más áreas en una red OSPF, siendo el área por defecto el área 0 o backbone. La agrupación de routers en áreas contribuye a disminuir las actualizaciones de la red, lo que la hace más eficiente.

En OSPF todas las áreas se conectan al área 0 para enviar los datos. Los routers que se encuentran dentro de un área se denominan routers de área, los routers que conectan a otras áreas a través del área 0 se conocen como routers de frontera de área (ABR), y los routers que se conectan a otro sistema autónomo se denominan routers de frontera de sistema autónomo (ASBR).(Araf, 2023).

En la Figura 5 se observa cómo se conecta todas las áreas de una red OSPF por medio del área 0 o backbone, analizando las posibles rutas para llegar a su destino.

Figura 5. Enrutamiento de una red por OSPF.



Fuente: (Papan et al., 2020)

### 2.4.1. Tipos de paquetes OSPF.

Los tipos de paquetes OSPF son cruciales para el funcionamiento eficaz y dinámica del protocolo de enrutamiento OSPF el cual fue creado para redes IP, este protocolo trabaja mediante la transmisión de varios paquetes entre los routers OSPF con la finalidad de poder intercambiar información fundamental sobre el estado de la red que se encuentran.

Cada uno de los paquetes en el protocolo de enrutamiento OSPF tiene un propósito determinado que permite a los routers OSPF descubrir routers vecinos, intercambiar información sobre bases de datos, solicitar y enviar detalles sobre el estado de los enlaces de la red y confirmar la recepción de información crítica. (Shamim, 2002).

En conjunto, estos paquetes contribuyen a la creación de una base de datos de estado de enlace consistente y actualizada, que permite el cálculo de las rutas más cortas y la toma de buenas decisiones de enrutamiento en entornos de red dinámicos. En este dinámico intercambio de información, los paquetes OSPF cumplen una función clave para mantener la convergencia de la red y garantizar una comunicación fluida y fiable entre los routers. Los tipos de paquetes OSPF son cinco y estos son:

- Paquete Hello: En el proceso OSPF se utilizan paquetes Hello para establecer y mantener relaciones de vecindad entre routers habilitados para OSPF.
- Paquete DBD: El paquete Data Base Description o en español paquete de descripción de base de datos (DBD) contiene un listado resumido de la base de datos de estado de enlace del router emisor, que es utilizado por el router receptor para realizar una comprobación cruzada con su base de datos de estado de enlace local. El paquete DBD permite al enrutador receptor conocer las redes que conoce del enrutador emisor y las que no conoce.(T et al., 2023).
- Paquete LSA: El paquete Link State Advertisement o en español paquete de anuncio de estado de enlace, es un anuncio individual acerca de redes específicas. Hay varios tipos de paquetes LSA que ayudan a construir la base de datos OSPF. En la Tabla 2.2 se

encuentra los tipos de paquetes de anuncio de estado de enlace que las redes utilizan comúnmente. (Sachinidis et al., 2023).

Tabla 2. Tipos de mensajes LSA

Tipo de LSA	Nombre	Descripción
1	Router LSA	Anuncio sobre una red. Inundaciones dentro de un área.
2	Network LSA	Anuncio sobre todas las rutas del mismo segmento Ethernet. Este tipo de LSAs son generados por el DR.
3	ABR Summary LSA	Anuncio de la ruta resumida de otra área. La ruta resumida se envía al área troncal. Este tipo de LSA es generado por el ABR.
4	ASBR Summary LSA	Anuncia la ubicación del ASBR. Es la dirección IP del router ASBR.
5	AS-External LSA	Anuncio de rutas externas resumidas desde fuera del Sistema Autónomo. Este tipo es generado por el ASBR.

Fuente: (Shaikh & Greenberg, 2004)

- Paquete LSR: El paquete Link State Request o Solicitud de Estado de Enlace es enviado por el router receptor sobre cualquier red en el DBD que no conocía.
- Paquete LSU: Link State Update o Actualización del estado del enlace, es la respuesta (contiene uno o más LSAs) a un paquete LSR. Contiene la información de estado de enlace que el router receptor solicitó en el proceso.
- Paquete LSAck: El paquete Link State Acknowledgement o Acuse de Recibo de Estado de Enlace, se envía por el router OSPF como confirmación de la recepción del LSU.

En OSPF el DR (Designated Router) o Router Designado es el router seleccionado dentro de un segmento de red actuando como punto focal en la comunicación OSPF, minimiza la sobrecarga al intercambiar información de estado de enlace solo con routers vecinos. El BDR (Backup Designated Router) o Router Designado de Respaldo, designa a un router que asume el papel de DR en caso de que falle el DR principal. (Li et al., 2012).

El ABR (Area Border Router) o Router de Borde de Área es el router encargado de conectar diferentes áreas OSPF en un sistema autónomo. El ASBR (Autonomous System Border Router) o Router de Borde del Sistema

Autónomo intercambia información de enrutamiento entre el área OSPF.(Herrera & Nunez, 2020).

### 2.4.2. Tipos de base de datos en OSPF.

En las redes OSPF cada vez que se envía un mensaje para ejecutar un proceso se crea una base de datos OSPF el cual determina varios parámetros esenciales para realizar gestiones de trayectorias. En la Tabla 3 podemos observar una descripción de cada base de datos en OSPF.

Tabla 3. Tipos base de datos OSPF.

Base de datos	Tabla	Descripción
Base de datos de adyacencia	Tabla de vecinos	Lista los routers vecinos en el que el router estableció comunicación bidireccional. En Cisco ejecutamos show ip ospf neighbor.
Base de datos de estado de enlace	Tabla de topología	Muestra los routers de la red. En routers Cisco se aplica el comando show ip OSPF database.
Base de datos de reenvío	Tabla de ruteo	Lista de rutas generadas cuando de ejecuta el algoritmo en la base de estado de enlace, la tabla de enrutamiento es única por cada router. En routers Cisco se utiliza el comando show ip route

Fuente: (Clausen et al., 2004)

### 2.4.3. Estados operativos de OSPF.

El protocolo OSPF utiliza un conjunto definido de estados operativos para gestionar la formación y el mantenimiento de las relaciones de vecindad entre routers en una red. Estos estados operativos, esenciales para la convergencia eficiente del protocolo, incluyen el estado Down, estado Init, estado Two-Way, estado Exstart, el estado Exchange, estado Loading y el estado Full. Cada uno de estos estados representa una fase específica en el proceso de establecimiento de vecindad OSPF, desde la detección inicial de vecinos hasta la sincronización completa de bases de datos de enrutamiento.

Es necesario comprender estos estados operativos porque es una parte fundamental para optimizar el rendimiento y la estabilidad de las redes OSPF, garantizando una eficaz distribución de información de enrutamiento en

entornos de red dinámicos. En la Tabla 4 se puede observar una descripción de cada uno de estos estados.

Tabla 4. Tipos estados operativos de OSPF

Estado	Descripción
Down State	Es el estado de activado, se encuentra en reposo para el siguiente estado.
Init State	Cuando una interfaz recibe su primer paquete Hello.
Two-Way State	Cuando se establece una comunicación entre 2 routers bidireccional, se elige un DR y BDR.
Exstart State	En eres punto a punto, los 2 routers deciden quine es el que inicia el intercambio y el número de secuencia del paquete inicial.
Exchange State	Los routers intercambian paquetes DBD.
Loading State	Se utiliza los LSR y LSU para obtener información de ruta adicional.
Full State	La base de datos de estado de enlace del router esta totalmente sincronizada.

Fuente: (Tiwari & Sahoo, 2007)

#### 2.4.4. ID de proceso y la wildcard.

La implementación efectiva de OSPF implica considerar varios aspectos clave, entre ellos el uso de áreas, el empleo de wildcard masks y la asignación de identificadores de proceso y estas se describen así:

- El área 0 conocida como área backbone, ejerce un papel fundamental al proporcionar conectividad entre otras áreas facilitando la gestión.
- La utilización de wildcard masks ofrece una flexibilidad significativa al definir rutas de filtrado las cuales permiten especificar patrones de dirección IP para la inclusión o exclusión de rutas específicas, lo que resulta esencial en la configuración de OSPF para adaptarse a los requisitos de enrutamiento específicos de cada red.(Ding et al., 2023).
- La asignación de identificadores de proceso en OSPF es importante para evitar conflictos y garantizar la coexistencia armoniosa de múltiples instancias del protocolo en una red lo que contribuye a una implementación coherente y eficiente de OSPF, mejorando la escalabilidad, la seguridad y la administración de la red en general.(Pearson IT Certification, 2012).

Para obtener la máscara wildcard es necesario tener la máscara de la dirección IP de la red a la que desea obtener la conversión, para luego realizar

la resta de la máscara total menos el inverso de la máscara de la red. En los siguientes ejemplos se explica cómo se llega al valor necesitado.

Máscara de la dirección IP de la red =/24 → 255.255.255.0

Wilcard = 255.255.255.255 – 255.255.255.0

Wilcard = 0.0.0.255

Máscara de la dirección IP de la red =/30 → 255.255.255.252

Wilcard = 255.255.255.255 – 255.255.255.252

Wilcard = 0.0.0.3

## 2.5. Introducción al simulador GNS3.

En el campo dinámico y en constante evolución de las redes de computadoras, las herramientas de simulación como GNS3 han emergido como recursos esenciales para el diseño, la prueba y el análisis de redes complejas, por ello es relevante explorar la utilidad y la aplicación de GNS3, un simulador de red de código abierto que permite a los usuarios replicar con precisión configuraciones de red sin necesidad de hardware físico. El GNS3 facilita un entorno de aprendizaje y experimentación rico y accesible para los profesionales y estudiantes de redes, con mayor fiabilidad a la generación de variables y obtención de resultados.(Castillo-Velazquez & Huerta, 2021).

La plataforma de simulación GNS3 (Graphical Network Simulator-3), es una plataforma de simulación de redes con la que los usuario diseñan, configuran y prueban redes de forma virtual. Esta herramienta es muy utilizada en la industria de redes y entornos de educación para probar las configuraciones de red y comprender el comportamiento de los dispositivos de red. (Nedyalkov, 2023).

Hay diversas razones concretas que justifican el uso del simulador GNS3 y que tienen relación con las capacidades específicas las cuales son propias de esta plataforma:

- Permite utilizar imágenes de disco de sistemas operativos de dispositivos de red reales, como routers y switches, etc. Los modelos de red IP creados de esta forma son lo más cercanos a las redes IP reales creadas a partir de dichos dispositivos de red.

- Capacidad de conexión de los modelos creados a redes IP reales o a Internet.
- Posee la opción de obtener una versión gratuita.

Todas estas capacidades convierten a GNS3 en una herramienta idónea para el estudio de las redes IP a través de la construcción de los modelos de redes IP a partir de imágenes de discos de dispositivos de red reales. La aplicación GNS3 posibilita la elaboración de todo tipo de redes IP, desde modelos muy simples hasta modelos muy sofisticados. En la Figura 6 se puede observar el logotipo del programa de simulación GNS3.

Figura 6. Logotipo de GNS3.



Fuente:(GNS3, 2023).

### **2.5.1. Requerimientos Generales**

Para ejecutar GNS3, es fundamental considerar los requerimientos de sistema que dependen de la complejidad de la red. Una configuración de sistema base podría incluir un procesador de 1.5 GHz y 4 GB de RAM, con un espacio libre de disco de 250 MB, que se considera suficiente para montajes de red y simulaciones básicas. Sin embargo, para proyectos que integren una variedad más amplia de dispositivos virtuales y topologías de red complejas, se recomienda un sistema con capacidades superiores. (Neumann, 2015).

Debido a que la simulación de redes extensas y la incorporación de elementos adicionales, como dispositivos de Juniper o la utilización de IOS avanzados, demandan mayor potencia de procesamiento y memoria. Así, un hardware más avanzado no solo maneja mejor las cargas de trabajo más exigentes, sino que también mejora la eficiencia y la fluidez de la simulación,

lo que resulta esencial para la experimentación y el aprendizaje en entornos de red complejos.(Sanchez et al., 2020).

### **2.5.2. Hardware emulado**

GNS3 se distingue por su capacidad para crear laboratorios de redes virtuales, que incluyen una variedad de enrutadores, conmutadores y PCs. Su ventaja principal se hace evidente cuando se utiliza junto con Cisco IOS, ya que GNS3 no se limita a replicar los comandos o características de Cisco, sino que emplea un hipervisor de fondo para emular el hardware que ejecuta Cisco IOS.(Neumann, 2015) .

Esto significa que solo el hardware emulado puede ejecutar una imagen real de IOS en la PC. Por lo tanto, cualquier comando o característica compatible con una versión de IOS está a disposición para tus diseños de red. Esta funcionalidad es lo que diferencia a GNS3 de otros programas de simulación de redes que ofrecen entornos, comandos y escenarios limitados.(Golightly et al., 2023)

### **2.5.3. Instalación de IOS router**

En el programa de simulación de GNS3 se puede utilizar IOS de routers o equipos de red de acuerdo con el modelo y marca, permitiendo una mayor experiencia al usuario al momento de implementar y diseñar una red, estas imágenes se pueden conseguir de manera legal o copiando dentro de un router que se posea utilizando el comando copy para mover la imagen al flash del router.(Neumann, 2015).

Después de obtener las imágenes de los IOS de los router se procede a crear una carpeta que contengan los archivos de los IOS, luego se abre el programa GNS3 y se aplica los siguientes pasos:

- En el entorno de GNS3 nos dirigimos a la pestaña Edit.
- Luego presionamos la opción Preferences.
- Se abrirá una pestaña con diversas opciones de la cual se elige IOS routers.
- Consecutivamente se presiona en la opción New.



- Elegir la opción New image y luego browse.
- Se elige la carpeta donde están todos los IOS descargados, luego la imagen que se necesita.
- Se escoge el tipo de tarjetas con los puertos que se requiere implementar en el router.

Es importante que se logre distinguir el tipo de tarjeta con los puertos de red que se quiere utilizar, en la Figura 7 se puede observar un ejemplo de cómo se presenta esta opción en el GNS3.

Figura 7. Tarjetas de red del IOS de un router



Fuente:(Nedyalkov, 2023).

#### 2.5.4. Ipterm

El Ipterm es una caja de herramientas de red basada en el sistema operativo Debian. Contiene las siguientes utilidades: net-tools, iproute2, ping, traceroute, curl, host, Iperf3, mtr, socat, ssh client, tcpdump y las herramientas de pruebas multicast msend/mreceive, herramientas indispensables para realizar las pruebas de medición.

Iperf es una herramienta de mediciones activas del ancho de banda máximo que se puede alcanzar en redes IP. Permite ajustar diversos parámetros relacionados con la sincronización, los búferes y los protocolos TCP, UDP, SCTP con IPv4 e IPv6. En cada prueba comunica el ancho de banda, las pérdidas y otros indicadores. Iperf es creado por NLANR/DAST.

Iperf3 es una nueva versión de Iperf la cual ha sido desarrollado principalmente por ESnet / Lawrence Berkeley National Laboratory la cuál es muy útil en entornos de simulación en GNS3.(Gueant, 2023).

## **2.6. Introducción básica a Python.**

Python es un lenguaje de programación interpretado, conocido por su facilidad de uso y su poderosa biblioteca estándar. Su sintaxis elegante y su tipado dinámico lo hacen ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, incluyendo web, análisis de datos, ciencia y educación. Python promueve la mejora continua y el intercambio de módulos y herramientas, lo que lo convierte en una opción robusta y flexible para programadores de todos los niveles.(Manca & Bonnici, 2023).

El lenguaje de programación Python es una potente herramienta de uso general ampliamente empleada en aplicaciones de Internet, desarrollo de software, ciencia de datos, aprendizaje automático. La filosofía de Python está basada en el código abierto y una comunidad en constante evolución. Su sintaxis es simple, y el código escrito se ejecuta en cualquier plataforma. Existen numerosas librerías para casi cualquier tarea, y a menudo estos paquetes se pueden obtener de forma gratuita.(Ryzhkov et al., 2023).

Es importante al momento de elegir Python el tipo de programa o aplicación que se desea crear, las librerías a utilizar para tener los recursos listos para su uso, para tener una idea clara de cómo se utilizará el código En la Figura 8 se puede observar el logotipo que representa el lenguaje de programación Python.

Figura 8. Logotipo de Python



Fuente:(Nedyalkov, 2023).

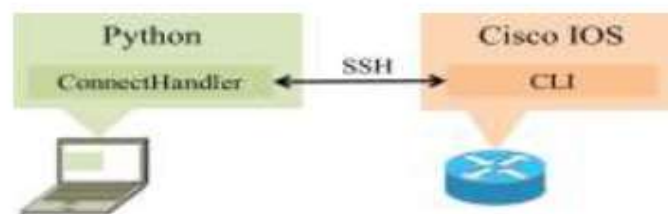
### **2.6.1. Librería Netmiko.**

La fuerza de Python radica en el empleo eficiente de sus bibliotecas especializadas con el menor esfuerzo posible y mediante comandos de

scripting muy abreviados. Python proporciona la biblioteca Netmiko, que simplifica la manipulación SSH de dispositivos de red y proporciona una salida de código de texto enriquecido al programador, lo que le permite centrarse en la configuración del dispositivo en términos de detalle SSH de baja complejidad.(Elezi & Karras, 2022).

Desde la librería de Netmiko, se importa un conjunto de funciones de fábrica, una de las cuales es ConnectHandlerclass, que elige la clase Netmiko correcta en función de la herramienta especificada y es una librería típica para dispositivos de varias empresas. En la Figura 9 se puede observar cómo se conecta Netmiko por medio de ConnectHandlerclass al IOS del router por CLI para realizar una correcta conexión de manera remota por SSH para mayor seguridad. (Elezi & Karras, 2022).

Figura 9. Conexión al router por medio de Netmiko



Fuente:(Elezi & Karras, 2022)

### 2.6.2. Librería Openpyxl.

Openpyxl es una biblioteca de Python diseñada para leer y escribir archivos Excel, proporcionando una forma intuitiva y amigable para manipular grandes conjuntos de datos requeridos por el usuario. Es especialmente útil en escenarios del mundo real, como la importación de nuevos productos a una base de datos o la exportación de datos para campañas de marketing, sin la necesidad de conocimientos técnicos avanzados.(Openpyxl / Openpyxl - GitLab, 2023).

La biblioteca facilita la creación de hojas de cálculo desde el principio, permitiendo a los usuarios instalar el paquete y comenzar a trabajar con archivos Excel en poco tiempo. Los desarrolladores pueden crear, leer y escribir en celdas de hojas de cálculo, adaptar estilos y formatos, e incluso trabajar con fórmulas y gráficos.(OpenPyXL, 2023).

Además, Openpyxl ofrece métodos para iterar a través de los datos de una hoja de cálculo y convertirlos en estructuras de datos de Python más útiles, como listas o diccionarios, e incluso transformarlos en clases de Python para una mejor estructuración de los datos, aprovechando la funcionalidad de las clases de datos disponibles desde Python 3.7, esto ayuda a crear nuevas funciones a partir de Python a Excel.

### **2.6.3. Librería Pandas.**

Pandas es una valiosa biblioteca de programación en Python que se ha convertido en una herramienta fundamental en el ámbito de la ciencia de datos y el análisis de datos. Su funcionalidad central radica en las estructuras de datos clave: el DataFrame, una tabla bidimensional con etiquetas en filas y columnas, y la Serie, una estructura unidimensional similar a una columna en una hoja de cálculo.(Burns, 2021).

Estas estructuras permiten cargar, manipular y analizar datos de manera eficiente, facilita tareas como la carga de datos desde diversas fuentes, la selección de columnas específicas, y la aplicación de filtros para extraer información relevante, ofrece funciones para realizar operaciones estadísticas y de agregación en los datos, proporcionando resúmenes descriptivos útiles.(García & María, 2019).

### **2.6.4. Librería Paramiko.**

Paramiko, una librería de Python especializada en implementaciones de los protocolos SSH, se destaca por posibilitar conexiones seguras y comunicación con servidores remotos. Su utilidad principal radica en la automatización de tareas en sistemas distribuidos y la ejecución de operaciones en entornos remotos de manera segura.

La librería Paramiko ofrece una implementación completa de clientes y servidores SSH, así como soporte para el protocolo SFTP para transferencias de archivos seguras, ideal para extracción de datos. La gestión de claves SSH, que incluye la generación y uso de claves públicas y privadas para autenticación, es otra característica clave de Paramiko.(Zadka, 2019).

La librería es versátil en términos de autenticación, admitiendo métodos como contraseña, clave privada y autenticación basada en teclado interactivo, facilita la creación de túneles SSH y el reenvío de puertos. Paramiko es compatible con Python 2 y 3, lo que simplifica su integración en diversos proyectos de programación.

Un ejemplo básico de la implementación de la librería Paramiko implica la conexión a un servidor SSH, la ejecución de un comando y la gestión de la salida correspondiente. Es esencial destacar que, en entornos de producción, se aconseja emplear la autenticación basada en claves por motivos de seguridad, y Paramiko proporciona herramientas efectivas para su manejo.(Choi, 2021).

#### **2.6.5. Comandos básicos de Python.**

Python, un lenguaje que destaca por su sintaxis clara y eficiente, proporciona estructuras de control esenciales para la programación dinámica. Estas estructuras de control son fundamentales para dirigir el flujo de un programa, permitiendo la implementación de lógicas complejas de una manera sencilla y comprensible y estas son:

- if: Permite la ejecución condicional de un bloque de código si se cumple una determinada condición.
- else: Se ejecuta como alternativa al bloque if cuando la condición inicial no se cumple.
- elif: Ofrece condiciones adicionales como parte de una cadena de if.
- Expresiones condicionales: Funcionan como operadores ternarios para evaluar una condición en una sola línea.
- for: Facilita la iteración sobre una secuencia, como una lista o un rango mejorando la eficiencia del código.
- while: Este comando permite ejecutar un bloque de código siempre y cuando la condición sea verdadera.
- continue: Omite el resto del código dentro de un bucle y continúa con la siguiente iteración.

- **break:** Termina un bucle antes de que haya recorrido todos los elementos, ofreciendo un control de flujo más preciso.

## 2.7. Introducción a PyCharm.

Es un entorno de desarrollo integrado (IDE) el cual posibilita la modificación de texto, la eliminación de errores y la completa interpretación del código Python. Esta herramienta dispone de funciones de completado e inspeccionado de código para una mayor funcionalidad al usuario. Del mismo modo, este instrumento efectúa informes de errores al instante, arreglos rápidos, refactorizado de código automático y funciones de navegación completas.(Lopez-Carreño et al., 2023).

Además, IDE cuenta con una consola interactiva de Python la cual es compatible con la distribución Anaconda (para corrección de errores en Python) e incorpora distintas librerías y paquetes científicos, tales como Matplotlib (gráficos en 2D y 3D) y NumPy (matrices multidimensionales) Este programa informático ha permitido la verificación y la validación del modelo ejecutado en Google Colab. (Arteaga, 2023).

PyCharm está disponible en una versión gratuita Community y una versión Professional de pago, esta última con características adicionales. En la Figura 10 se observa el logotipo del programa PyCharm.

Figura 10. Logotipo del programa PyCharm



Fuente:(Jetbrians, 2023)

### 2.7.1. Entorno básico de PyCharm.

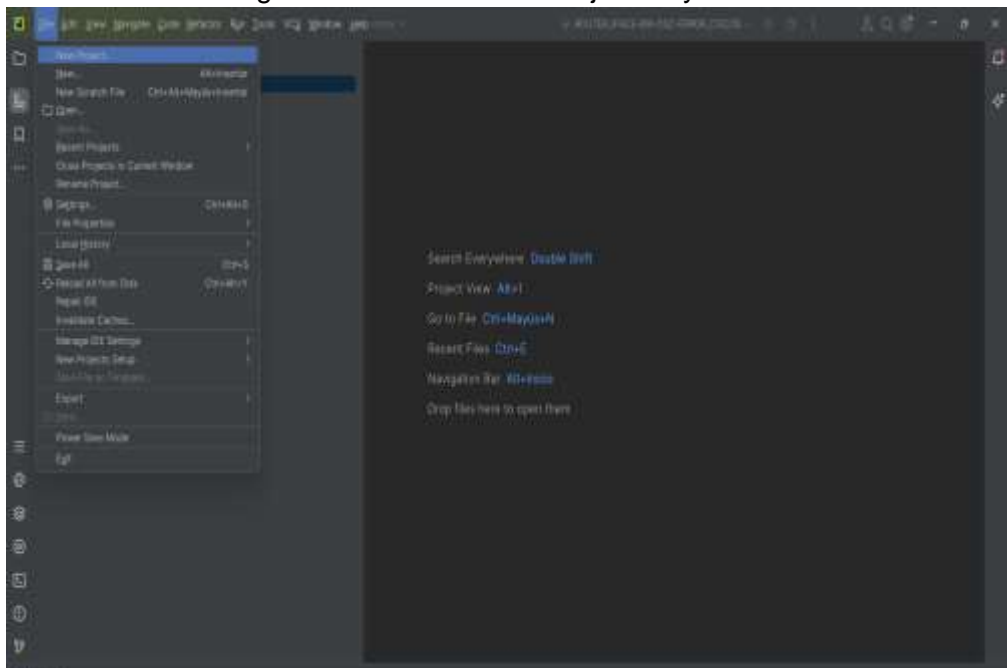
El entorno básico de PyCharm incluye varias características y herramientas clave que lo hacen un IDE eficiente y fácil de usar para programadores de Python Este entorno es altamente personalizable y se puede adaptar a las necesidades específicas del desarrollador para el cliente de una empresa.(Gupta, 2022).

Es de gran importancia conocer el funcionamiento de cada parte del entorno de trabajo de PyCharm y estas son:

- Editor de Código: Ofrece resaltado de sintaxis, autocompletado y sugerencias de código, mejorando la legibilidad en la escritura.
- Navegación y Búsqueda: Facilita la navegación por el código y la búsqueda rápida de archivos, clases, métodos, usos, etc.
- Herramientas de Depuración y Ejecución: Incluye un depurador gráfico para la ejecución y depuración de código desde el IDE.
- Consola de Python: Ofrece una consola integrada para ejecutar scripts de Python y experimentar con código en tiempo real.
- Gestión de Proyectos: Permite la organización fácil de proyectos con una vista estructurada de archivos y carpetas.
- Control de Versiones Integrado: Soporta sistemas de control de, facilitando el seguimiento de cambios y la colaboración.
- Terminal Integrado: Incluye un terminal integrado para ejecutar comandos del IOS y scripts directamente desde el IDE.

Este entorno de trabajo es factible para el entendimiento de IDE En la Figura 11 se puede observar un entorno de trabajo de PyCharm.

Figura 11. Entorno de trabajo en PyCharm



Fuente: (Gupta, 2022)

### **2.7.2. Ventajas del programa PyCharm.**

El programa PyCharm es un Entorno de Desarrollo Integrado (IDE) avanzado y específicamente diseñado para Python, uno de los lenguajes de programación populares, este programa es utilizado en el lenguaje de programación Python y fue desarrollado por JetBrains.(Shang et al., 2023).

PyCharm ofrece varias ventajas que lo hacen destacar:

- **Análisis de Código y Depuración:** Incluye herramientas de análisis de código, lo que ayuda a identificar errores y sugerir correcciones. Su potente depurador facilita la identificación y solución de problemas en el código.
- **Soporte para Frameworks Web:** PyCharm soporta varios frameworks de desarrollo web como Django, Flask, y Pyramid, lo que lo hace ideal para desarrollo web en Python.
- **Integración con Herramientas y Servicios:** Ofrece integración con sistemas de control de versiones como Git, así como con bases de datos y servicios en la nube.
- **Ambientes Virtuales:** Facilita la gestión de diferentes entornos de desarrollo Python, permitiendo a los usuarios trabajar con múltiples versiones de Python y librerías sin conflictos.
- **Personalización y Extensibilidad:** Permite personalizar su entorno según las necesidades del usuario y es extensible mediante plugins.
- **Edición y Refactorización de Código:** Provee una amplia gama de herramientas para la edición de código y refactorización, mejorando la eficiencia en la escritura y mantenimiento del código.

### **2.7.3. Características del programa PyCharm.**

PyCharm se presenta como un completo entorno de desarrollo integrado que está meticulosamente adaptado para la programación en Python. Este IDE destaca por su suite integrada de herramientas de desarrollo, soporte para una gama de tecnologías web y científicas, y una interfaz altamente personalizable que es consistente a través de plataformas. Facilita la escritura de código limpio y eficiente, y está equipado con funcionalidades para la



depuración y pruebas, que lo hace indispensable para los desarrolladores que buscan optimizar su flujo de trabajo y aumentar su eficiencia. (Jetbrians, 2023).

PyCharm ofrece diversas características las cuales son de gran relevancia su entendimiento y estas son:

- Asistencia inteligente en la codificación: Ofrece autocompletado inteligente, inspecciones de código, destacado de errores y soluciones rápidas con refactorización automática y capacidades de navegación.
- Herramientas de desarrollo integradas: Incluye un depurador integrado, corredor de pruebas, perfilador de Python, terminal integrada, integración con sistemas de control de versiones, herramientas de base de datos, y capacidades de desarrollo remoto con intérpretes remotos.
- Desarrollo web: Proporciona soporte especializado para frameworks de desarrollo web de Python, lenguajes de plantilla específicos.
- Personalizable y multiplataforma: Funciona en Windows, macOS y Linux con una sola clave de licencia, con esquemas de colores personalizables, atajos de teclado y emulación de VIM.

## **2.8. Introducción a Tableau.**

Tableau es una herramienta de visualización de datos potente y líder en la industria. Permite a los usuarios transformar datos en bruto en visualizaciones comprensibles, como gráficos, dashboards interactivos. Diseñado para ser accesible a usuarios sin conocimientos técnicos avanzados, Tableau facilita el análisis de grandes cantidades de datos.

El programa Tableau soporta una amplia gama de fuentes de datos, desde hojas de cálculo simples hasta bases de datos complejas. Su capacidad para realizar análisis en profundidad y compartir visualizaciones lo hace valioso en toma de decisiones basada en datos en diversos sectores empresariales y sociales. (Neteris, 2023).

La herramienta de Tableau tiene múltiples ventajas en la actualidad lo que la convierte en lo idóneo para el análisis de datos. En la Figura 12 se puede visualizar el logo de Tableau.

Figura 12. Logotipo de Tableau



Fuente: (Neteris, 2023)

### **2.8.1. Funcionalidades de Tableau.**

Tableau sobresale en la creación rápida de análisis avanzados y visualizaciones impresionantes, sin necesidad de desarrollos técnicos complejos. Ofrece una gran versatilidad en su configuración, pudiendo operar tanto en servidores locales, en dispositivos individuales de usuarios, como en la nube. Su capacidad para integrar datos de múltiples fuentes y su uso intuitivo facilitan análisis ágiles y colaborativos.

Tableau proporciona acceso inmediato a datos de variados sistemas, bases de datos y hojas de cálculo. Permite a los usuarios gestionar y compartir sus análisis, fomentando la independencia de los departamentos de TI y agilizando la toma de decisiones. Además, su escalabilidad y las opciones personalizables de seguridad aseguran un manejo eficiente tanto de los recursos como de la información sensible.(Neteris, 2023).

### **2.8.2. Ventajas de Tableau.**

El programa de visualización de datos Tableau ofrece diversas ventajas las cuales son esenciales para determinar el uso y la funcionalidad que se le dará en determinados aspectos y estas son:

- **Eficiencia:** Facilita análisis rápidos y efectivos para abordar interrogantes empresariales complejas, contribuyendo a la expansión de negocios mediante una mejor toma de decisiones, permite realizar a las empresas análisis más complejos de manera rápida y efectiva.
- **Análisis visual avanzado:** Ofrece una interpretación clara de los datos, transforma datos crudos en información visualmente accesible, facilitando una mayor interpretación de una gran cantidad de datos.
- **Intuitividad:** Su facilidad de uso permite una rápida adaptación por parte de los usuarios. Esta facilidad de uso democratiza el acceso a

herramientas de análisis de datos, permitiendo que un rango más amplio de empleados en una organización.

- Costo: Representa una opción económica, con un precio inferior al promedio del mercado, lo cual lo hace atractivo para el cliente.
- Confiabilidad: Es una plataforma segura, escalable y de confianza, ofreciendo solidez en términos de seguridad y escalabilidad.

### **2.8.3. Características de Tableau.**

Tableau, una herramienta líder en visualización de datos y análisis de negocios destaca por su versatilidad y potencia. Ofrece características como dashboards interactivos, opciones de colaboración, y aplicaciones móviles para un acceso conveniente. Su enfoque en la seguridad de datos, actualizaciones en tiempo real, y la integración con herramientas analíticas avanzadas que hacen de Tableau una solución adaptable para diversas necesidades empresariales que lo vuelve atractivo al usuario.(Neteris, 2023).

### Capítulo 3: Diseño, implementación y resultados

En este capítulo se detalla y describe la creación de la red enmallada en GNS3, de la cual se extraerán los KPIs por medio de Python, el cual generará un Excel con los datos relacionados y estos documentos se subirán a Tableau para la visualización en un entorno gráfico.

#### 3.1. Elaboración de la topología de la red enmallada en GNS3.

Es muy importante que se tenga el diseño de la topología de la red para mayor comprensión sobre las partes al momento de implementarlo. En este caso se tendrá 2 redes las cuales sus routers estarán entrelazados entre sí, para un mayor redundancia creando un modelo enmallado, cada red representará una región y sus subredes una ciudad estratégica.

En la primera red, se encontrará con 3 routers los cuales se conectan al router central que a su vez estará conectada con el otro router central de la otra red el cual permitirá el paso de datos de la Red 1 a la Red 2 que tiene conectado 2 routers enlazados entre sí. En esta red se usará una nomenclatura indicando el tipo de elemento y la ciudad que estará presente. Por lo tanto se utilizarán los siguientes: RTYGU01, RTMLG01, RTDUR01, RTGYE01, RTUIO01, RTCNC01, RTRI001.

Luego de realizar las conexiones físicas de cada router y determinar los puertos que se utilizarán, se procede a subneatear de acuerdo con la necesidad de usuarios en la red. Como ejemplo se visualiza las redes del router RTYGU01 que en la tabla 5 se puede visualizar ya subneteada.

Tabla 5. Datos para redes del RTYGU01

Interface	Máscara	Prefijo	Wildcard	Red
G0/0	255.255.255.252	/30	0.0.0.3	10.10.10.12
G2/0	255.255.255.252	/30	0.0.0.3	10.10.10.16
G6/0	255.255.255.252	/30	0.0.0.3	40.40.40.0
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.5

Fuente: Autor.

Después de subnetear y definir parámetros de la red incluyendo la interface Loopback, se procede a realizar el mismo procedimiento para el router RTMLG01 el cual se observa en la Tabla 6 dando los datos requeridos

para poder realizar su respectiva configuración de manera ordenada evitando errores en su conexión.

Tabla 6. Datos para redes del RTMLG01

Interface	Máscara	Prefijo	Wildcard	Red
G0/0	255.255.255.252	/30	0.0.0.3	10.10.10.12
G3/0	255.255.255.252	/30	0.0.0.3	10.10.10.20
G6/0	255.255.255.252	/30	0.0.0.3	10.10.10.24
F5/1	255.255.255.224	/27	0.0.0.31	192.168.1.64
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.4

Fuente: Autor.

El tercer router que se determinará los datos que se obtendrá al momento de subnetear será el RTDUR01, el cual se podrá visualizar los datos requeridos en la Tabla 7 de forma específica.

Tabla 7. Datos para redes del RTDUR01

Interface	Máscara	Prefijo	Wildcard	Red
G1/0	255.255.255.252	/30	0.0.0.3	10.10.10.16
G3/0	255.255.255.252	/30	0.0.0.3	10.10.10.20
G4/0	255.255.255.252	/30	0.0.0.3	10.10.10.28
F5/1	255.255.255.224	/27	0.0.0.31	192.168.1.96
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.2

Fuente: Autor.

Una vez terminado el router RTDUR01, se procede a determinar los parámetros a subnetear en el router central de la esta red, el cual es el RTGYE01 que se puede visualizar en la Tabla 8 sus resultados. Este router es el que permite el paso de tráfico a la otra red.

Tabla 8. Datos para redes del RTGYE01

Interface	Máscara	Prefijo	Wildcard	Red
G0/0	255.255.255.252	/30	0.0.0.3	10.10.10.32
G4/0	255.255.255.252	/30	0.0.0.3	10.10.10.28
G6/0	255.255.255.252	/30	0.0.0.3	10.10.10.24
F5/0	255.255.255.192	/26	0.0.0.63	192.168.1.0
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.3

Fuente: Autor.

Terminado la red 1 se procede a la red 2 con el router que se conecta con el RTGYE01, es decir el RTUIO01, en la Tabla 9 se puede visualizar los datos de la red obtenidos al subnetear.

Tabla 9: Datos para redes del RTUIO01

Interface	Máscara	Prefijo	Wildcard	Red
G0/0	255.255.255.252	/30	0.0.0.3	10.10.10.32
G1/0	255.255.255.252	/30	0.0.0.3	20.20.20.4
G2/0	255.255.255.252	/30	0.0.0.3	20.20.20.0
G6/0	255.255.255.252	/30	0.0.0.3	30.30.30.0
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.6

Fuente: Autor.

Luego de terminar el router RTUIO01, se procede a subnetear las redes del router RTCNC01 y se las muestra en la Tabla 10, la cual se observa diversos parámetros para tomar en cuenta al momento de configurar el router y su red.

Tabla 10. Datos para redes del RTCNC01

Interface	Máscara	Prefijo	Wildcard	Red
G1/0	255.255.255.252	/30	0.0.0.3	20.20.20.4
G3/0	255.255.255.252	/30	0.0.0.3	20.20.20.8
G6/0	255.255.255.252	/30	0.0.0.3	30.30.30.0
F5/0	255.255.255.192	/26	0.0.0.63	1921.68.2.0
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.7

Fuente: Autor.

Luego de subnetear el router RTCNC01, se procede con el router RTRIO01, el cual se definirá en la Tabla 11 los parámetros de red necesarios para configurar en el router.

Tabla 11. Datos para redes del RTRIO01

Interface	Máscara	Prefijo	Wildcard	Red
G2/0	255.255.255.252	/30	0.0.0.3	20.20.20.0
G3/0	255.255.255.252	/30	0.0.0.3	20.20.20.8
G6/0	255.255.255.252	/30	0.0.0.3	40.40.40.5
Loopback	255.255.255.255	/32	0.0.0.0	1.1.1.8

Fuente: Autor.

### 3.1.1. Diseño de la topología.

Cada router se figura como una ciudad estratégica para el tráfico de datos dependiendo del servicio que se necesita. Para configurar el nombre de cada router se aplica en el modo privilegiado y luego al modo de configuración global con el siguiente comando, a continuación, se muestra la configuración de cada router:

```

RTYGU01#configure terminal.
RTYGU01(config)#hostname RTYGU01
RTYGU01 (config)#do wr
RTDUR01#configure terminal.
RTDUR01(config)#hostname RTDUR01
RTDUR01(config)#do wr
RTMLG01#configure terminal.
RTMLG01 (config)#hostname RTMLG01
RTMLG01(config)#do wr
RTGYE01#configure terminal.

```

```
RTGYE01 (config)#hostname RTGYE01
RTGYE01 (config)#do wr
RTUIO01#configure terminal.
RTUIO01 (config)#hostname RTUIO01
RTUIO01 (config)#do wr
RTRIO01#configure terminal.
RTRIO01 (config)#hostname RTRIO01
RTRIO01 (config)#do wr
RTCNC#configure terminal.
RTCNC (config)#hostname RTCNC
```

Cada empresa o sector privado tiene su propias reglas las cuales se debe da a conocer a cualquier trabajador o cliente que ingrese a la red; para ello se configura en el router por medio del siguiente comando.

```
RTGYE01#configure terminal
RTGYE01 (config-if) #banner motd ""
```

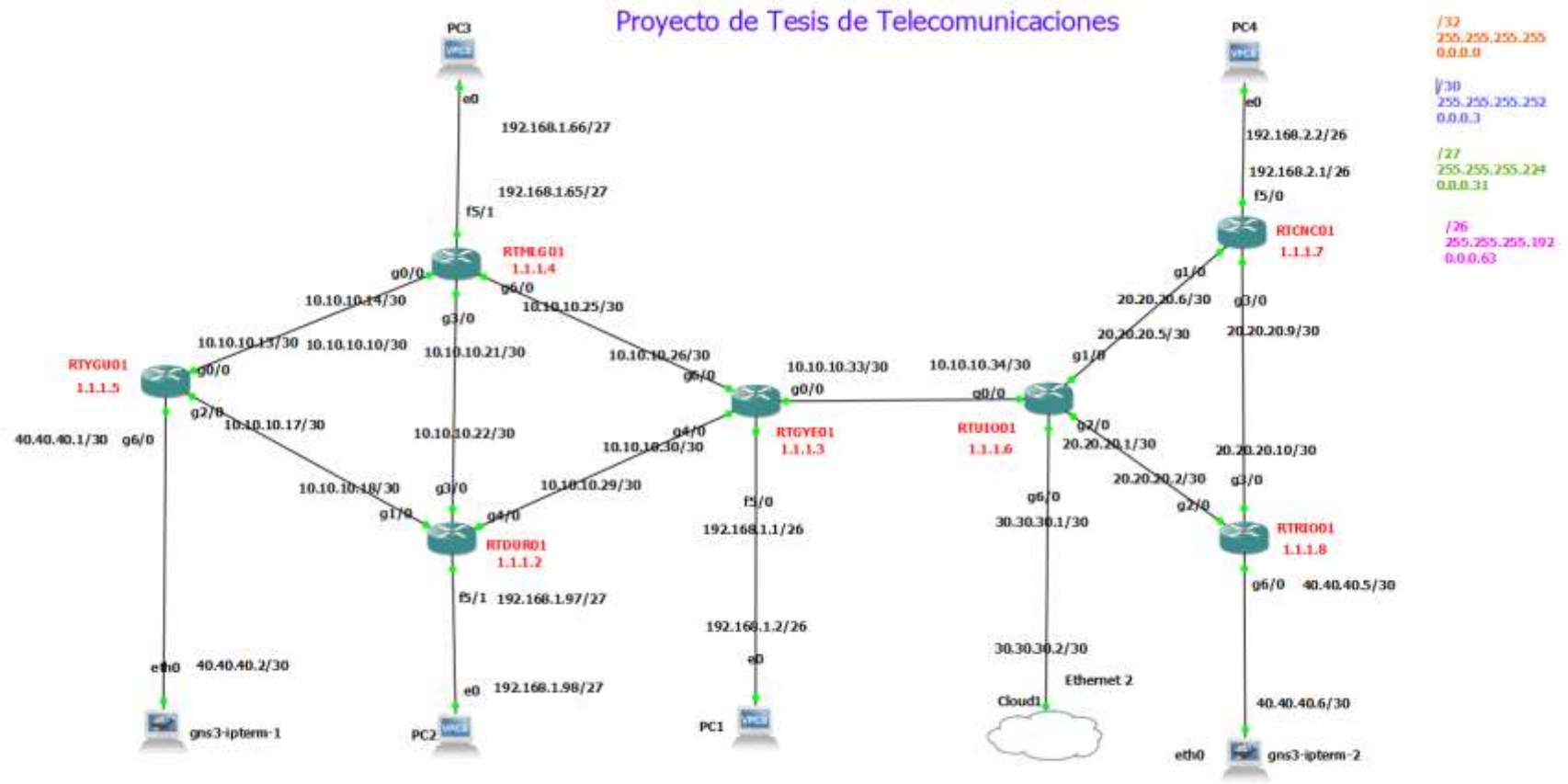
Es necesario la configuración de cada interfaz de un router, poniendo la velocidad, el modo de transmisión, la descripción, la IP de la interface junto a su máscara y lo encendemos utilizando los siguientes comandos:

```
RTGYE01#configure terminal
RTGYE01 (config)#interface gigabitEthernet 0/0
RTGYE01 (config-if) #description XXXXX
RTGYE01 (config-if) #duplex full/auto/half
RTGYE01 (config-if)#speed 10/100/1000/auto
RTGYE01 (config-if)#ip address 10.10.10.33 255.255.255.252
RTGYE01 (config-if) #no shutdown
```

Para realizar su comprobación al conectar y configurar con el otro router al que se conecta, se realiza una prueba de conexión con el comando ping para poder asegurarse de que la conexión este establecida entre cada router. En la Figura 13 se puede observar el diseño de la topología.

```
RTGYE01#ping 10.10.10.34
RTGYE01#show run
```

Figura 13. Topología de la Red Enmallada en GNS3



David Andrés Alvarez Granda

Elaborado por: Autor.



### 3.1.2. Conexión de routers por OSPF.

Para establecer el enrutamiento por OSPF es necesario determinar algunos datos para implementar al momento de configurar cada router, como es el Router-ID que en este caso es la IP loopback del router, y las redes de los routers vecinos con su respectiva wildcard, por consiguiente, ejecutamos el siguiente comando en cada router con los datos correspondiente, un ejemplo de este formato es el RTGYE01 el cual es:

```
RTGYE01#configure terminal
RTGYE01 (config) #router ospf 1
RTGYE01 (config-router) #router-id 1.1.1.3
RTGYE01 (config-router) # network 1.1.1.3 0.0.0.0 area 0
RTGYE01 (config-router) # network 10.10.10.24 0.0.0.3 area 0
RTGYE01 (config-router) #network 10.10.10.28 0.0.0.3 area 0
RTGYE01 (config-router) #network 10.10.10.32 0.0.0.3 area 0
RTGYE01 (config-router) #network 192.168.1.0 0.0.0.63 area 0
```

### 3.1.3. Configuración de SSH.

Cada router tendrá conexión de acceso SSH para poder ingresar al equipo de manera remota, lo cual es una pieza fundamental para la extracción de KPIs por medio de Python, por eso es necesario la conectividad. Primero se configura la IP de dominio, luego la cantidad de bits para encriptación deseada, consecutivamente se configura el número de intentos de autenticación para ingresar, se ejecuta los siguientes comandos:

```
RTGYE01#configure terminal
RTGYE01 (config) #ip domain name suc3.com
RTGYE01 (config) #crypto key generate rsa
1024(se escoge la cantidad que desea)
RTGYE01 (config) #ip ssh authentication-retries 3
RTGYE01 (config) #username david privilege 15 secret 2001
```

Consecutivamente se configura las líneas virtuales las cuales son las que permite en acceso de manera remota, se activa SSH, se pone un tiempo de espera de inactividad, ingreso local y la versión de SSH en este caso es la versión 2 y se guarda utilizando los siguientes comandos:

```

RTGYE01 (config) #line vty 0 15
RTGYE01 (config-line) #transport input ssh
RTGYE01 (config-line) #exec-timeout 60
RTGYE01 (config-line) #login local
RTGYE01 (config-line) #exit
RTGYE01 (config) #ip ssh version 2
RTGYE01 (config) #do wr
RTGYE01 (config) #do show run

```

### 3.2. Diseño de escenarios de simulación en GNS3.

Para obtener variaciones de los KPIs al momento de extraer es necesario generar escenarios, los cuales podrían ser situaciones que suelen suceder de manera común para ello es necesario conocer la naturaleza del detalle para interpretar su posible resultado.

#### 3.2.1. Escenario 1: Todos los routers con prioridad 1.

En este escenario se deja los routers en su enrutamiento por defecto lo cual define la prioridad 1, cuando todos tiene prioridad 1, el de mayor dirección IP del Neighbor ID es la que determinará si es DR y la menor determinará si es BDR en ambos casos pueden existir más de una clase perteneciente al grupo mencionado. En la Figura 14 se muestra el escenario 1 que al extraerse en un Excel los datos se observa cómo se asocia una IP en relación con otra para asignarse como DR o BDR cuando todos los routers tiene prioridad 1.

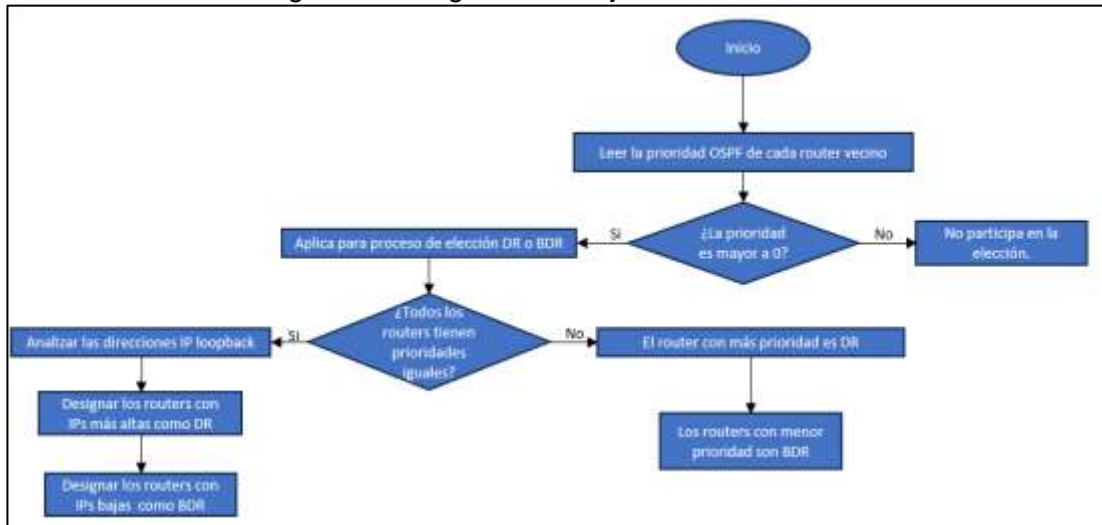
Figura 14. Escenario 1 router DR/BDR

	A	B	C	D	E	F	G
1	Router IP	Neighbor ID	Pri	State	Dead Time	Address	Interface
2	1.1.1.2	1.1.1.3	1	FULL/DR	00:00:36	10.10.10.30	GigabitEthernet4/0
3	1.1.1.2	1.1.1.4	1	FULL/DR	00:00:33	10.10.10.21	GigabitEthernet3/0
4	1.1.1.2	1.1.1.5	1	FULL/DR	00:00:37	10.10.10.17	GigabitEthernet1/0
5	1.1.1.3	1.1.1.6	1	FULL/DR	00:00:38	10.10.10.34	GigabitEthernet0/0
6	1.1.1.3	1.1.1.2	1	FULL/BDR	00:00:39	10.10.10.29	GigabitEthernet4/0
7	1.1.1.3	1.1.1.4	1	FULL/DR	00:00:39	10.10.10.25	GigabitEthernet6/0
8	1.1.1.4	1.1.1.3	1	FULL/BDR	00:00:36	10.10.10.26	GigabitEthernet6/0
9	1.1.1.4	1.1.1.2	1	FULL/BDR	00:00:33	10.10.10.22	GigabitEthernet3/0
10	1.1.1.4	1.1.1.5	1	FULL/DR	00:00:35	10.10.10.13	GigabitEthernet0/0
11	1.1.1.5	1.1.1.2	1	FULL/BDR	00:00:39	10.10.10.18	GigabitEthernet2/0
12	1.1.1.5	1.1.1.4	1	FULL/BDR	00:00:39	10.10.10.14	GigabitEthernet0/0
13	1.1.1.6	1.1.1.7	1	FULL/DR	00:00:35	20.20.20.6	GigabitEthernet1/0
14	1.1.1.6	1.1.1.8	1	FULL/DR	00:00:39	20.20.20.2	GigabitEthernet2/0
15	1.1.1.6	1.1.1.3	1	FULL/BDR	00:00:37	10.10.10.33	GigabitEthernet0/0
16	1.1.1.7	1.1.1.8	1	FULL/DR	00:00:34	20.20.20.10	GigabitEthernet3/0
17	1.1.1.7	1.1.1.6	1	FULL/BDR	00:00:32	20.20.20.5	GigabitEthernet1/0
18	1.1.1.8	1.1.1.7	1	FULL/BDR	00:00:31	20.20.20.9	GigabitEthernet3/0
19	1.1.1.8	1.1.1.6	1	FULL/BDR	00:00:38	20.20.20.1	GigabitEthernet2/0

Elaborado por: Autor.

Para un mayor entendimiento del escenario de todos los routers con prioridad 1 es necesario representarlo con un diagrama de flujo que se muestra en la Figura 15 mostrando las decisiones que toma el router para asignarse como DR o BDR.

Figura 15. Diagrama de flujo del escenario 1



Elaborado por: Autor.

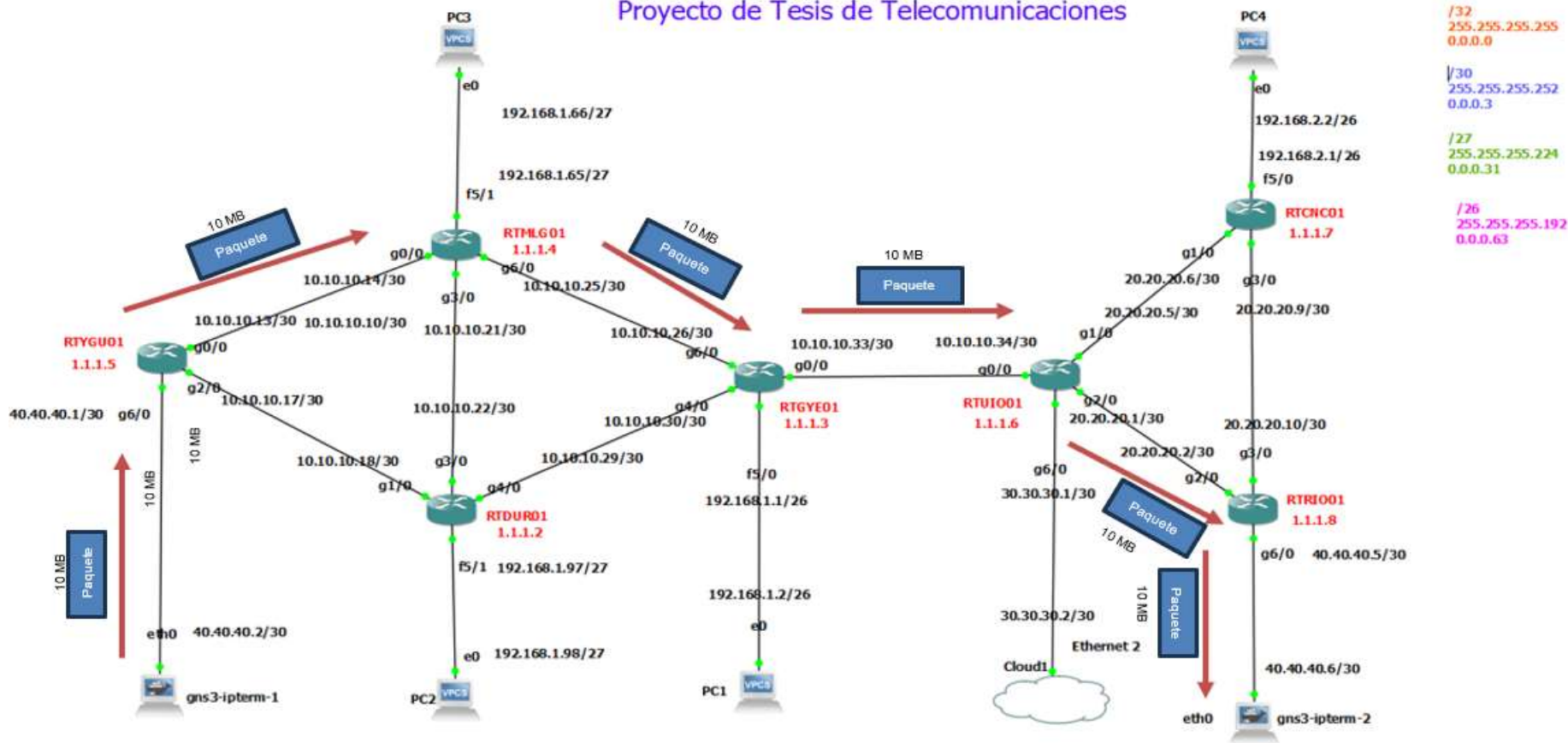
### 3.2.2. Escenario 2: Envío de 10 Megabytes a las interfaces.

Para pruebas de conectividad en una interface, es necesario generar tráfico para obtener KPIs necesarios al enviar datos. En la Figura 3.4 se visualiza el escenario en el que se envía 10 Megabytes y los saltos que realiza para llegar al destino solicitado. En este escenario se utilizará la herramienta Ipterm la cual es de gran utilidad para realizar pruebas de conectividad, ancho de banda, etc. Por ello se realiza los siguientes pasos:

- Primero se abre la topología de la red en el programa GNS3.
- Se descarga GNS3 VM y Virtual Box para realizar la instalación en el programa de GNS3. Para ello se da clic en la pestaña nuevo de Virtual Box donde se añade la VM GNS3. Luego en el entorno de simulación de GNS3 se da clic en Edit, Preferences, luego en GNS3 VM, se elige la opción que se requiera para correr la máquina virtual.
- Se añade 2 end devices de la categoría de Ipterm los cuales se usarán para las pruebas de conectividad.
- Luego se da clic derecho sobre el dispositivo, elegir la opción auto start console , luego se presiona en Edit. En la Figura 16 se puede visualizar el envío de paquetes en la topología de la red.

Figura 16. Escenario 2 envío de 10 Megabytes

Proyecto de Tesis de Telecomunicaciones

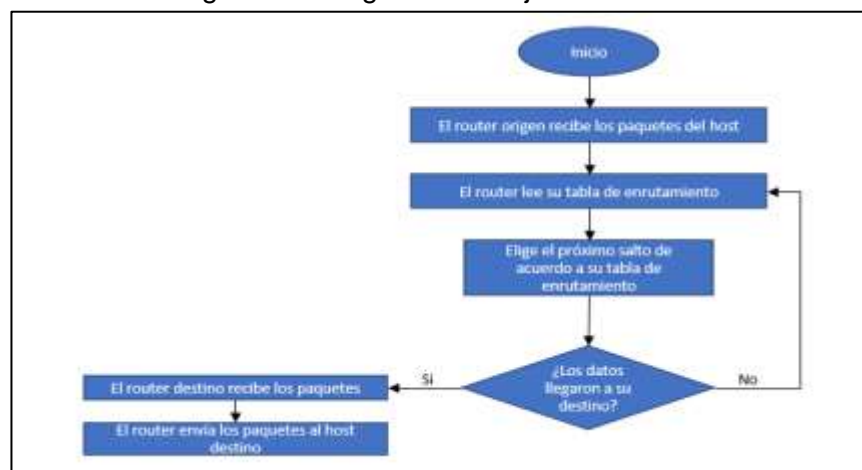


David Andrés Alvarez Granda

Elaborado por: Autor

- Se reemplaza la dirección IP, gateway y máscara los cuales están configurados por defecto, luego se elimina el carácter # para que se tome esa línea de comando al momento de iniciar el equipo, consecutivamente se da clic en aplicar. En el Ipterm 2 se tendrá la IP 40.40.40.6 con máscara 255.255.255.252 y el gateway correspondiente que es el 40.40.40.5. En el Ipterm 1 se tiene la IP 40.40.40.2 con máscara 255.255.255.252 con el gateway 40.40.40.1
- Luego se conecta a la interface de cada router que se necesita en este caso es el router RTYGU01 para el Ipterm1 y el RTRIO01 para el Ipterm 2. En el router RTYGU01 se utilizará la interface g6/0 , con la dirección IP 40.40.40.1 con máscara /30 con la descripción Connection RTYGU01 to Customers. En el router RTRIO01 se utilizará la interface g6/0 con la IP 40.40.40.5 con máscara /30 con la descripción Connection RTRIO01 to Customers.
- Se realiza pruebas de conectividad con el comando ping más la IP de la interface del otro dispositivo.
- Se define el tipo de paquete que se enviará, el tiempo que se ejecutará las prueba , la tasa de bits para enviar tráfico. Es necesario determinar cuál end device será el servidor y cual dispositivo será el cliente el cual es donde se ejecutará el comando. En la Figura 17 se observa el diagrama de flujo del envío del paquete desde el origen al destino.

Figura 17. Diagrama de flujo del escenario 2



Elaborado por: Autor.

Una vez realizado la instalación y configuración del end device Ipterm para las prueba de interfaces, se define que el Ipterm2 será el servidor para

activarlo se ejecuta el comando iperf3 -s. El Ipterm 1 se utilizará como cliente el cual usará paquetes UDP de 10 Megabits por segundo con el comando iperf3 -u -t 600 -b 10M -c 40.40.40.6 el cuál comenzará el envío de paquetes. En la Figura 18 se visualiza el envío de paquetes de un dispositivo a otro.

Figura 18. Pruebas de interface Ipterm

```

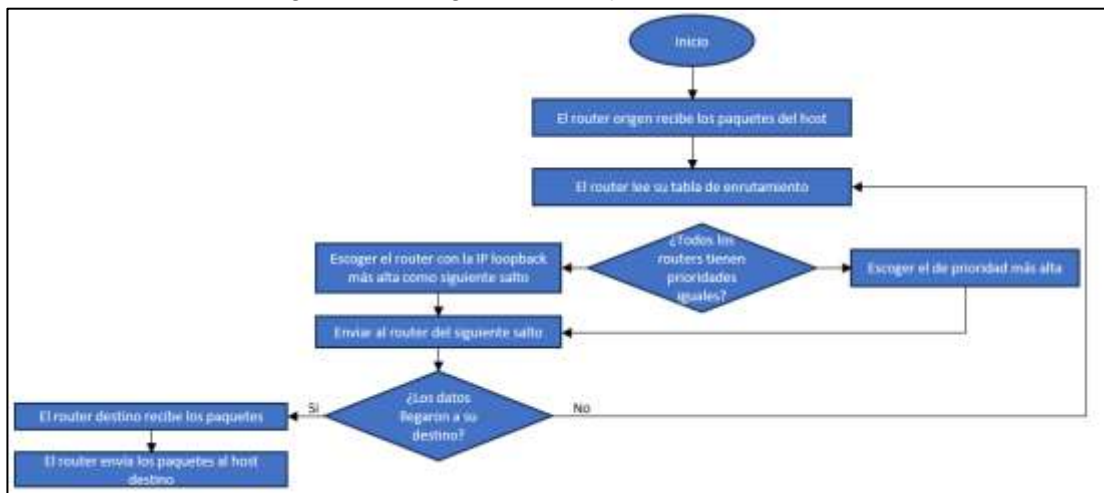
root@gnsl-ipterm-2:~# iperf3 -s
Server listening on 5201 (test #1)
Accepted connection from 40.40.40.2, port 51656
5) local 40.40.40.6 port 5201 connected to 40.40.40.2 port 4218?
ID Interval Transfer Bitrate 11111111 11111111 11111111
5) 0.00-1.00 sec 789 Kbytes 6.46 Mbits/sec 2.806 ms 0/558 (0%)
5) 1.00-2.00 sec 1.04 Mbytes 8.72 Mbits/sec 1.423 ms 12/264 (1.0%)
5) 2.00-3.00 sec 1.02 Mbytes 8.53 Mbits/sec 2.248 ms 87/821 (11%)
5) 3.00-4.00 sec 1.15 Mbytes 9.63 Mbits/sec 1.583 ms 150/911 (15%)
5) 4.00-5.00 sec 1.09 Mbytes 9.15 Mbits/sec 1.782 ms 136/926 (15%)
5) 5.00-6.00 sec 1.04 Mbytes 8.76 Mbits/sec 1.385 ms 26/782 (3.3%)
5) 6.00-7.00 sec 1.13 Mbytes 9.48 Mbits/sec 1.694 ms 60/886 (7.7%)
5) 7.00-8.00 sec 945 Kbytes 7.74 Mbits/sec 2.759 ms 66/736 (9%)
5) 8.00-9.00 sec 909 Kbytes 7.91 Mbits/sec 1.982 ms 184/869 (21%)
  
```

Elaborado por: Autor.

### 3.2.3. Escenario 3: Ruta OSPF al apagar interfaces.

En OSPF se escoge la ruta con menor costo o peso para llegar a su destino, en este caso todos los routers tiene prioridad 1, por ende, se analiza que camino toma el router para llegar a su destino, para ello se analiza la topología y las direcciones IP loopback. En la Figura 19 se observa el diagrama de flujo del escenario 3 que se utilizarán para generar 3 rutas.

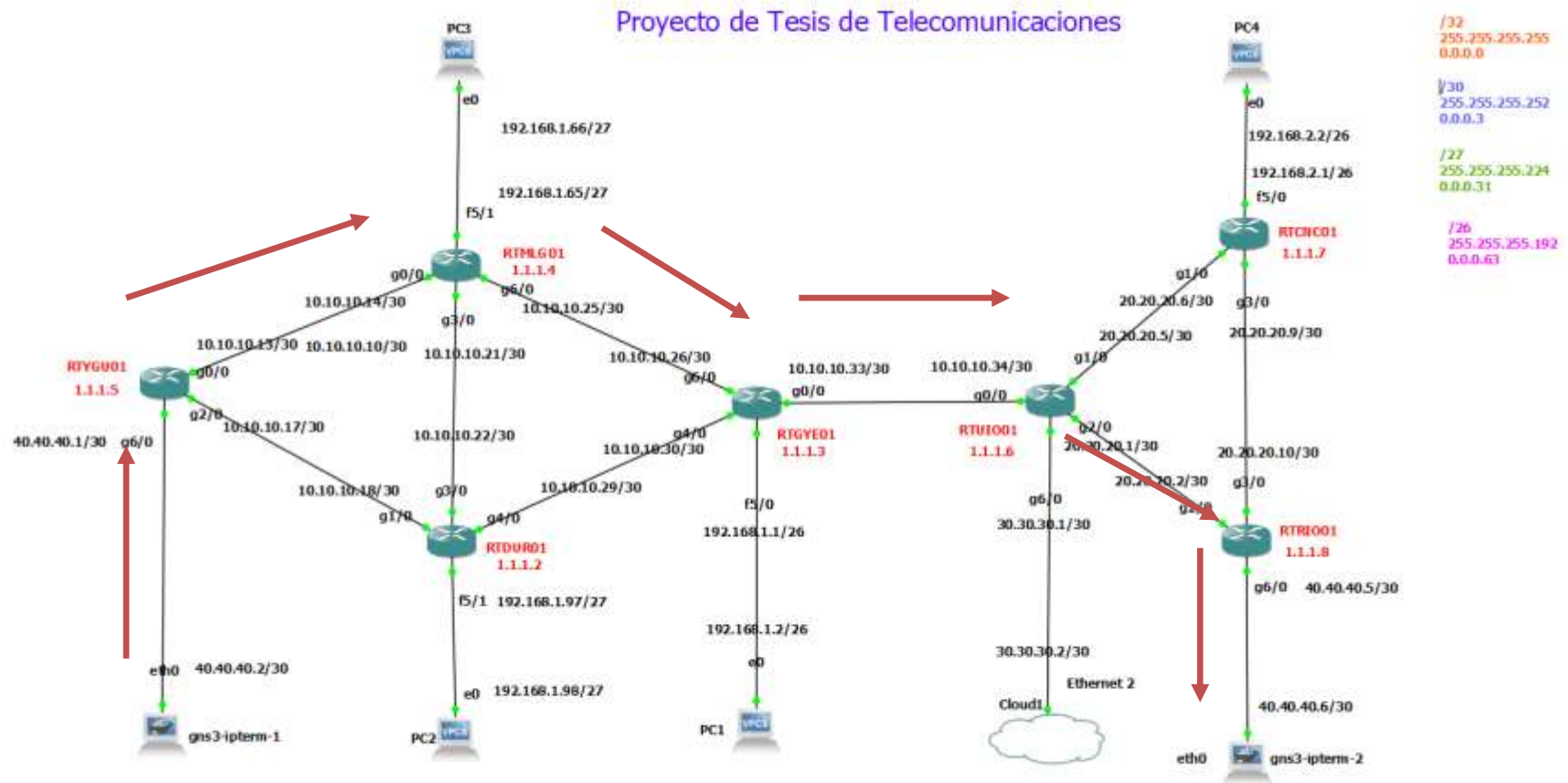
Figura 19. Diagrama de flujo del escenario 3



Elaborado por: Autor.

Define como DR al router con mayor prioridad, pero si existen más de 2 routers DR se escoge el de la IP más alta. Por ello se conoce la primera ruta es la que se muestra en la Figura 20 el cual es el primer trayecto del envío de datos del Ipterm1 hacia el Ipterm2.

Figura 20. Primera ruta definida por OSPF



David Andrés Alvarez Granda

Elaborado por: Autor.

Se visualiza que el tráfico de RTYGU01 tiene 2 alternativas las cuales son la conexión con RTDUR01 o la conexión con RTMLG01 de lo cual escoge la de la prioridad mayor en este caso son iguales, por ende, escogerá la de IP mayor la cual es RTMLG01. Para generar la segunda ruta se apaga la interface g0/0 del router RTMLG01 para forzar el tráfico vaya por el router RTDUR01 para ello se ejecuta los siguientes comandos:

```
RTMLG01#configure terminal
RTMLG01 (config)#interface gigabitEthernet 0/0
RTMLG01 (config-if) # shutdown
RTMLG01 (config-if) #do wr
```

Una vez apagada la interface g0/0 del router RTMLG01 el tráfico se topa las siguientes alternativas de salto la conexión con el RTDUR01 a RTMLG01 o la conexión RTDUR01 hacia RTGYE01. Para forzar que vaya por la segunda alternativa para crear una trayectoria diferente se debe apagar la interface g3/0 del RTMLG01 mediante los siguientes comandos:

```
RTMLG01#configure terminal
RTMLG01 (config)#interface gigabitEthernet 3/0
RTMLG01 (config-if) # shutdown
RTMLG01 (config-if) #do wr
```

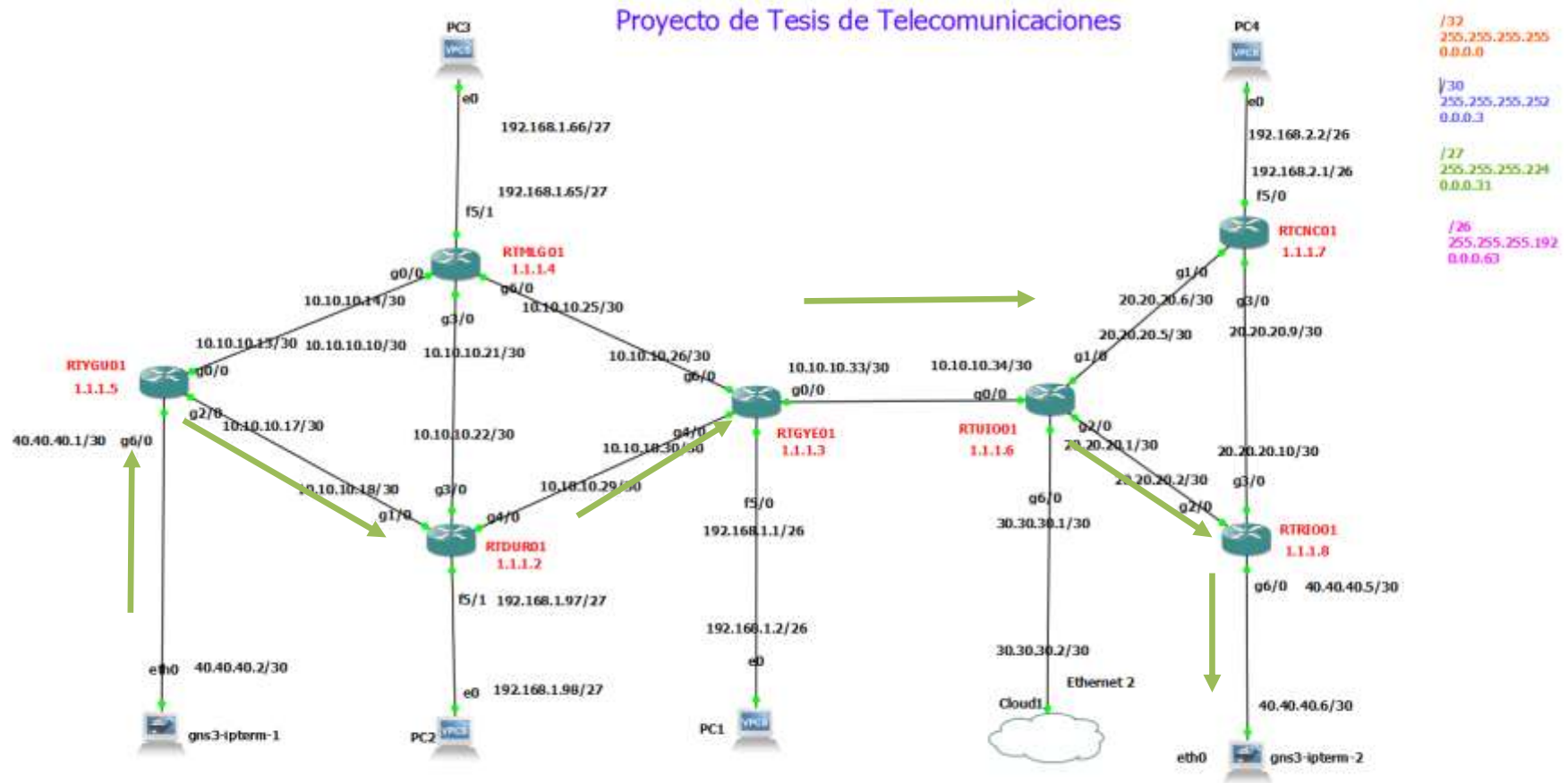
Luego el tráfico pasará desde el RTGYE01 hacia el RTUIO01 , al llegar tendrá dos caminos que son la conexión con RTCNC01 o la conexión RTRIO02 de lo cual conocemos que irá por el RTRIO01 ya que es el camino más corto, con la IP más alta y también es DR, pero para asegurarse se apaga la interface g1/0 del RTCNC01 con los siguientes comandos:

```
RTCNC01#configure terminal
RTCNC01 (config)#interface gigabitEthernet 1/0
RTCNC01 (config-if) # shutdown
RTCNC01 (config-if) #do wr
```

En la Figura 21 se observa cual es la ruta en forma gráfica que toma la data al realizar los pasos que anteriormente fueron solicitados a configurar.



Figura 21. Segunda ruta definida por OSPF



David Andrés Alvarez Granda

Elaborado por: Autor.

Para la tercera ruta la data comienza en el end device lpterm 1 ,la data entra por la interfaz g6/0 del RTYGU01 y tiene 2 alternativas que es la conexión del RTYGU01 con el RTMLG01 y la conexión del RTYGU01 con el RTDUR01, en este caso se requiere que pase por el RTMLG01 por ello apagamos la interface g1/0 del RTDUR01 con los siguientes comandos:

```
RTDUR01#configure terminal
RTDUR01 (config)#interface gigabitEthernet 1/0
RTDUR01 (config-if) # shutdown
RTDUR01(config-if) #do wr
```

Luego la data se encuentra en el RTMLG01, pero se requiere que pase al RTDUR01 para ello se apaga la interface g6/0 del RTGYE01 que conecta el RTMLG01 con el RTGYE01 para forzar que el flujo de datos del próximo salto al RTDUR01, se ejecuta los siguientes comandos.

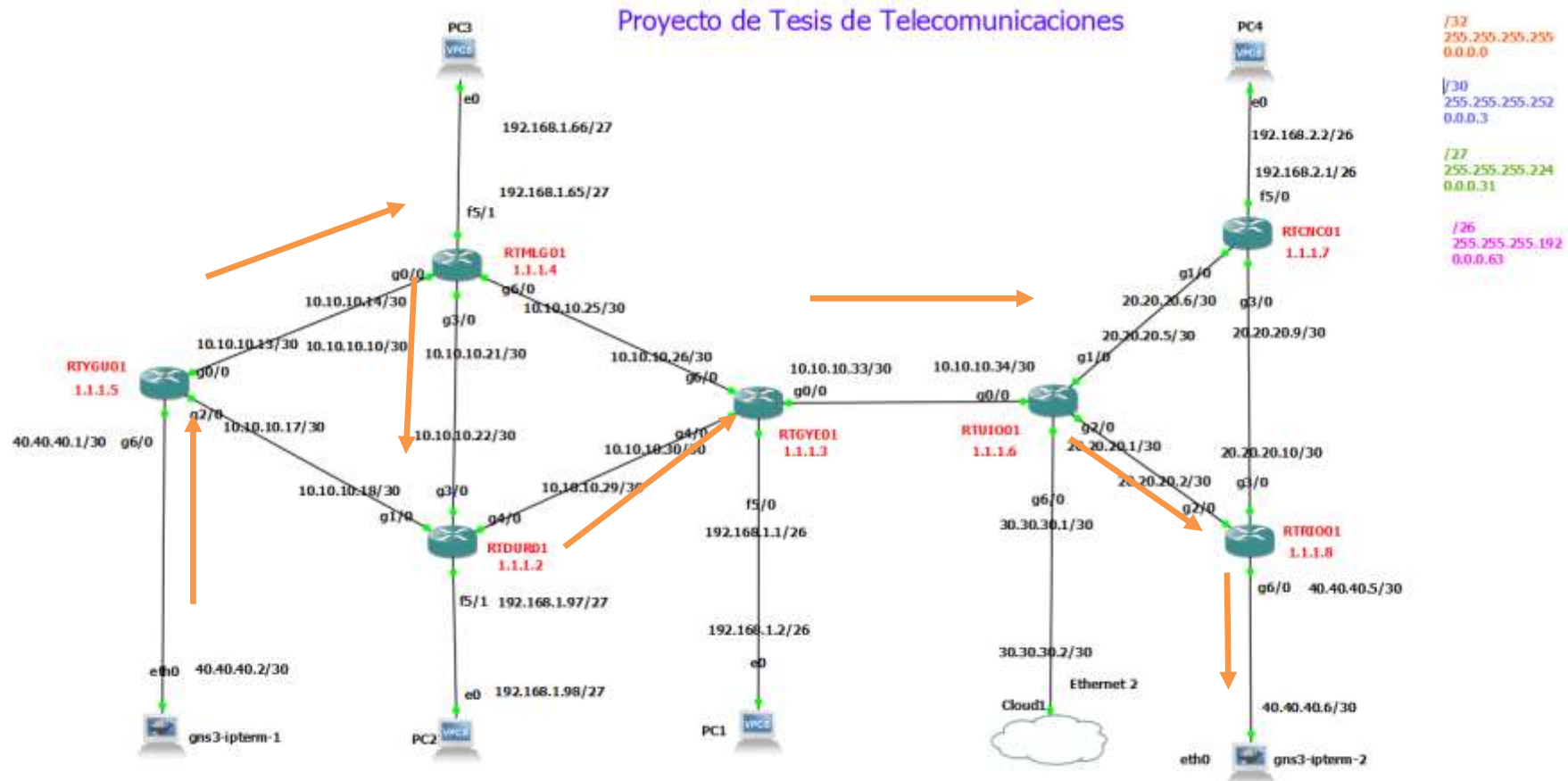
```
RTGYE01#configure terminal
RTGYE01 (config)#interface gigabitEthernet 6/0
RTGYE01 (config-if) # shutdown
RTGYE01 (config-if) #do wr
```

Cuando llega al RTDUR01 la data pasa al RTGYE01 por medio del enlace que conecta el RTDUR01 con RTGYE01. El RTGYE01 direcciona el flujo de datos al RTUIO01 el cual tiene 2 opciones RTCNC01 o RTRIO01, se requiere utilizar el RTCNC01, por ende, se apaga al interface g2/0 del RTRIO01 para que la data fluya al RTCNC01. Cuando los datos lleguen al RTCNC01 este direccionará al RTRIO01 el cual llevará el tráfico al end device lpterm 2, para ello se ejecuta los siguiente comandos:

```
RTRIO01#configure terminal
RTRIO01 (config)#interface gigabitEthernet 42/0
RTRIO01 (config-if) # shutdown
RTRIO01 (config-if) #do wr
```

Existen más rutas, pero las mencionadas son de utilidad para conocer que los enlaces funcionan correctamente en OSPF. En la Figura 22 se observa la tercera ruta generada por OSPF.

Figura 22. Tercera ruta definida por OSPF



David Andrés Alvarez Granda

Elaborado por: Autor.

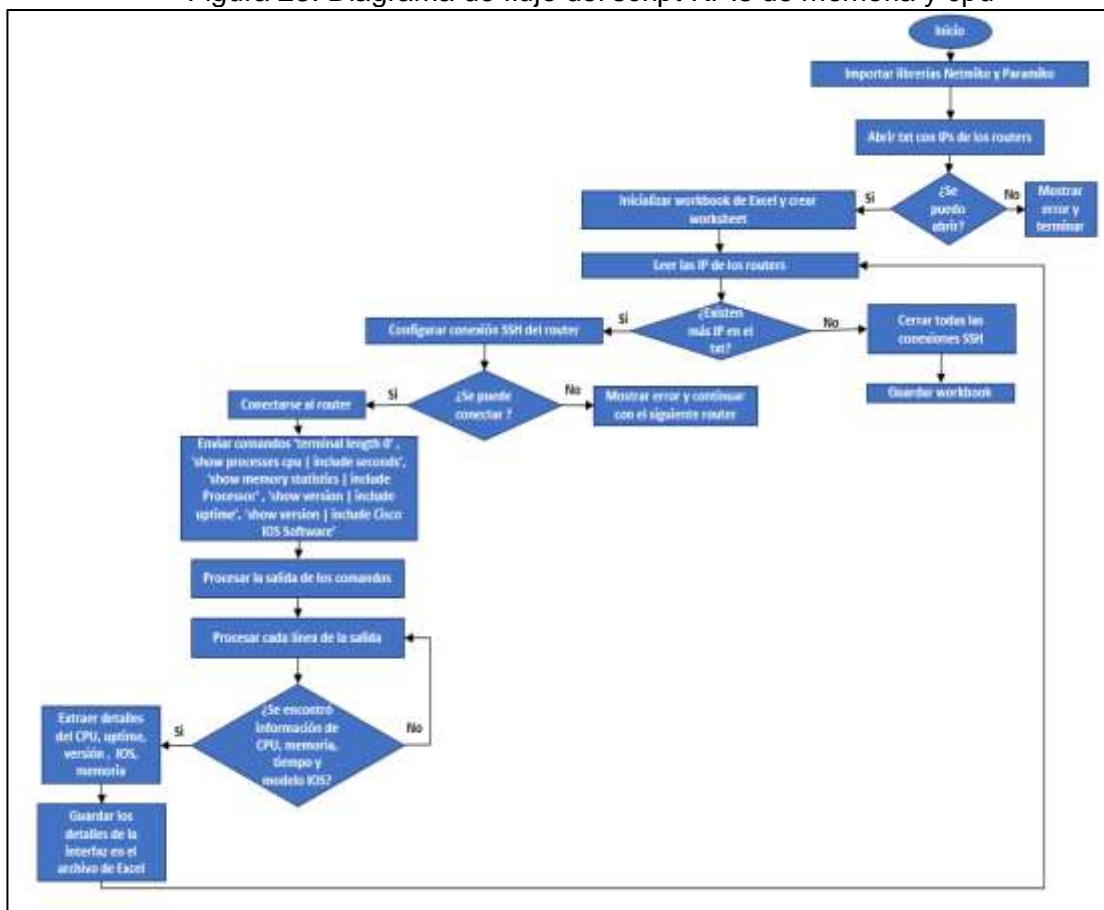
### 3.3. Extracción de KPIs por medio de Python.

Es muy importante para realizar la extracción que la PC del usuario pueda hacer ping a cualquier IP de la red para pruebas de conectividad de cada router, junto al acceso por SSH. En este caso se utiliza el programa PyCharm para ejecutar los scripts de Python para la extracción de KPIs con las librerías de Python netmiko, openpyxl, paramiko y pandas. Para su instalación se ejecuta los comandos `pip install paramiko`, `pip install paramiko`, `pip install paramiko`, `pip install netmiko`, `pip install pandas` `pip install openpyxl`.

#### 3.3.1. Script de KPIs de Memoria y Versión.

Para la creación de este script es necesario conocer dónde va a obtenerse la información de la cual se va a extraer los datos requeridos para su respectiva recopilación para ello creamos un blog de notas con las direcciones IPs loopback de cada router las cuales deben estar configuradas en cada equipo, las librerías, para la extracción de los KPIs. En la Figura 23 se observa el diagrama de flujo del script de KPIs de memoria y versión.

Figura 23. Diagrama de flujo del script KPIs de memoria y cpu



Elaborado por: Autor

Es importar las librerías se usará para las acciones requeridas en el script, también que comandos se utilizará al ingresar al elemento para extraer las los diferentes valores de los KPIs solicitados y almacenarlos en variables, se necesitará mencionar el usuario y contraseña para el ingreso y el formato sería el siguiente:

```
# Importar las bibliotecas necesarias
from netmiko import ConnectHandler # Biblioteca para manejar conexiones a
dispositivos de red
import openpyxl # Biblioteca para trabajar con archivos Excel
# Cargar los nombres de host/IP de los routers desde un archivo
with open('ips_rt_cisco.txt') as f:
    hostnames = f.read().splitlines()
# Inicializar el libro de trabajo de Excel
workbook = openpyxl.Workbook()
worksheet = workbook.active
worksheet.title = 'CPU-MEM-UPTIME' # Puedes mejorar el nombre según tus
necesidades
# Agregar encabezados de columna
worksheet.append(['Router IP', 'CPU % USED', 'MEM_USED', 'UPTIME',
'MODEL', 'VERSION'])
# Configurar la conexión SSH con los routers
for hostname in hostnames:
    device = {
        'device_type': 'cisco_xr',
        'ip': hostname,
        'username': 'david',
        'password': '2001',
        'global_delay_factor': 2, # Aumentar el delay factor
        'session_log': f"log_{hostname}.txt", # Activar el log
        'timeout': 40 # Configurar el timeout
    }
    # Conectar al router
    try:
        net_connect = ConnectHandler(**device)
```

```

    print(f"Conexión exitosa al router {hostname}")
except Exception as e:
    print(f"Error al conectarse al router {hostname}: {e}")
    continue
# Ejecutar los comandos
output = net_connect.send_command('terminal length 0') # Para que la
información que se obtiene del comando se muestre totalmente
output_cpu = net_connect.send_command('show processes cpu | include
seconds')
output_mem = net_connect.send_command('show memory statistics |
include Processor')
output_uptime = net_connect.send_command('show version | include
uptime')
output_ios_model = net_connect.send_command('show version | include
Cisco IOS Software')

```

Luego de tener en cuenta los valores que se requiere por medio de comandos que se ejecutará la correr el programa se debe tener en cuenta que se debe unir todos los resultados antes de procesarlos por ello se usa:

```

# Variables para almacenar
current_row = len(worksheet['A']) + 1 # Asegura que los nuevos datos se
agreguen en la siguiente fila disponible.
cpu_found = mem_found = uptime_found = ios_model_found = False
cpu_v1 = mem_total = mem_usedfinal = mem_used = uptime = ios = model
= None
# Se crea una lista con las variables almacenadas las cuales poseen un
salto de línea.
lines = output_cpu.split("\n") + output

```

Al extraer la información se realiza su procesamiento por ello se utiliza condicionantes para extraer el KPI solicitado y almacenarlo en una variable, luego de procesarlo se lo agrega a las celdas en Excel de acuerdo con el orden deseado con los siguientes comandos:

```

# Iterar a través de las líneas de la salida del comando
for line in lines:

```

```

if line.startswith('CPU utilization'):
    cpu = line.split()[11] # También puede ser -1
    cpu_v1 = cpu.split('%')[0]
    cpu_found = True
if line.startswith('Processor'):
    mem_total = int(line.split()[2])
    mem_used = int(line.split()[3])
    mem_found = True
if "uptime" in line:
    uptime_info = line.split()[3:]
    uptime = ' '.join(uptime_info) # Convierte la lista a una cadena
separada por espacios
    uptime_found = True
if line.startswith('Cisco IOS Software'):
    model = line.split()[3] # También puede ser -1
    ios = line.split(",")[2].split()[1]
    ios_model_found = True
if cpu_found and mem_found and uptime_found and ios_model_found:
    mem_usedfinal = (mem_used / mem_total) * 100
    # Agregar los valores a las celdas
    worksheet.cell(row=current_row, column=1, value=hostname)
    worksheet.cell(row=current_row, column=2, value=cpu_v1)
    worksheet.cell(row=current_row, column=3, value=mem_usedfinal)
    worksheet.cell(row=current_row, column=4, value=uptime)
    worksheet.cell(row=current_row, column=5, value=model)
    worksheet.cell(row=current_row, column=6, value=ios)
    # Incrementar la fila actual
    current_row += 1
    # Restablecer las variables
    cpu_v1 = mem_total = mem_usedfinal = mem_used = uptime = ios =
model = None
    cpu_found = mem_found = uptime_found = ios_model_found = False
    # Cerrar la conexión
    net_connect.disconnect()

```

Una vez obtenido los KPIs y procesarlos en una hoja de Excel se procede a guardarlos en un Libro de Excel el cual se agregará al directorio configurado para ello se utiliza los siguientes comandos:

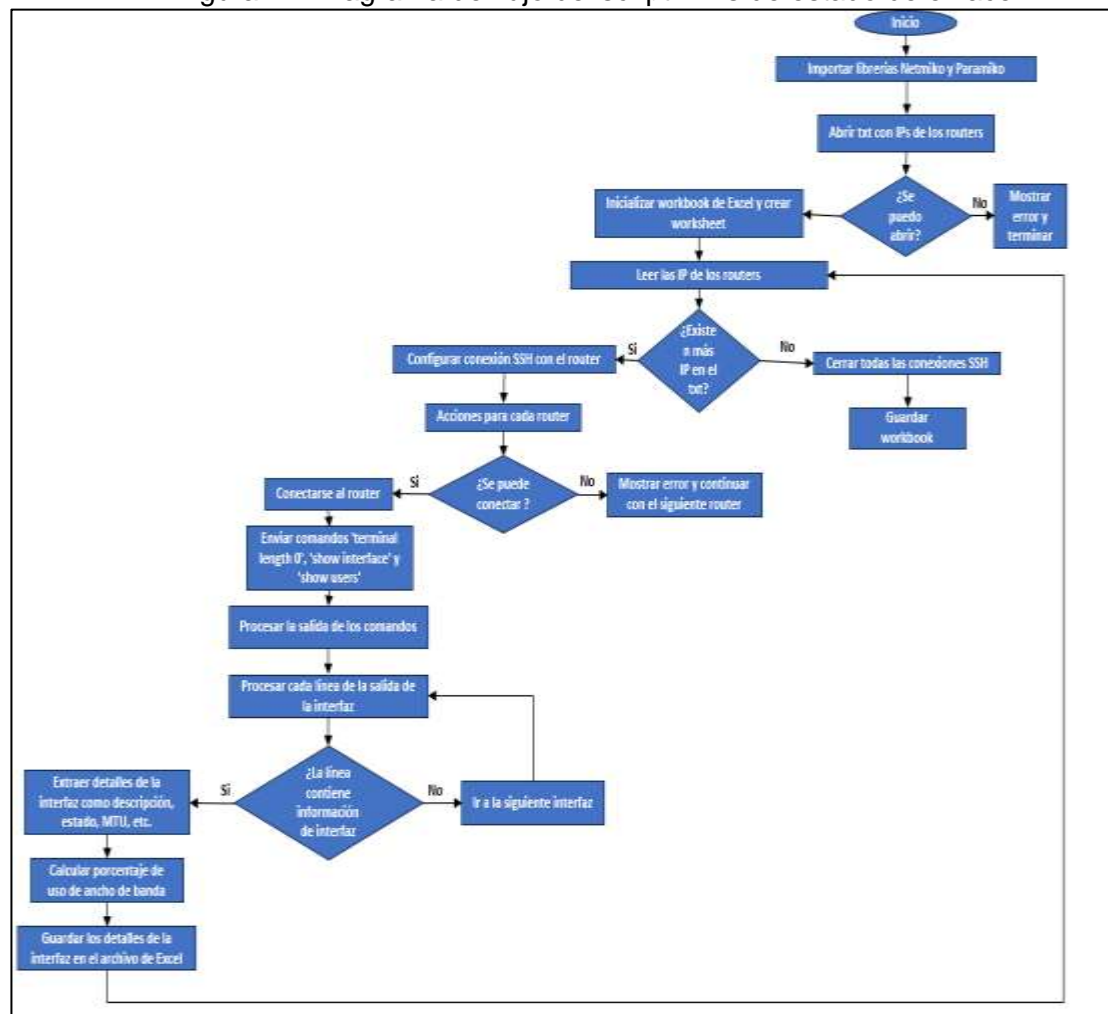
# Guardar el libro de Excel

```
workbook.save('ROUTER_CPU_MEM_UP-TIME)#Es el nombre que se guardará el archivo de Excel
```

### 3.3.2. Script de KPIs de estado de interface.

En este script, se debe reconocer los comandos que se utiliza para mostrar la información necesaria para su extracción y almacenamiento en variables especificadas en el script, también las IPs loopback de cada router que se requiera recopilar para ello las adjuntamos en un blog de notas, sus librerías para su proceso, y los comandos a utilizar. En la Figura 24 se observa el diagrama de flujo del script de KPIs de estado del interface.

Figura 24. Diagrama de flujo del script KPIs de estado de enlace



Elaborado por: Autor



Luego determinamos los comandos que se utilizarán para obtener las líneas de lectura las cuales serán almacenadas en variables, el usuario y contraseña para el ingreso remoto por lo ello el formato sería el siguiente:

```
# Importar las bibliotecas necesarias
from netmiko import ConnectHandler
import openpyxl # Cargar los nombres de host/IP de los routers desde un
archivo ips
with open('ips_rt_cisco.txt') as f:
    hostnames = f.read().splitlines()
# Inicializar el libro de trabajo de Excel
workbook = openpyxl.Workbook()
worksheet = workbook.active
worksheet.title = 'IFACE INFORM'
# Agregar encabezados de columna
worksheet.append(['Router IP', 'IFACE', 'DESC', "STATUS", 'MTU', "
DUPLEX", "BW_GIGA", "INPUT_RATE%", "OUTPUT_RATE%",
"INPUT_ERROR", "OUTPUT_ERROR", 'DROPS'])
# Configurar la conexión SSH con los routers
for hostname in hostnames:
    # Definir los parámetros del dispositivo para la conexión SSH
    device = {
        'device_type': 'cisco_xe',
        'ip': hostname,
        'username': 'david',
        'password': '2001',
        'global_delay_factor': 2, # Aumentar el factor de retraso para permitir
que los comandos se ejecuten completamente
        'session_log': f"log_{hostname}.txt", # Activar el registro de sesiones
para guardar la salida de los comandos en un archivo
        'timeout': 40 # Configurar el tiempo máximo de espera para la conexión
    }
    # Conectar al router
    try:
        net_connect = ConnectHandler(**device)
```

```

    print(f"Conexión exitosa al router {hostname}")
except Exception as e:
    print(f"Error al conectarse al router {hostname}: {e}")
    continue
# Ejecutar los comandos en el router
output = net_connect.send_command('terminal length 0') # Configurar la
longitud del terminal para obtener toda la salida
output_iface = net_connect.send_command('show interface') # Obtener
información de la interfaz
output_users = net_connect.send_command('show users') # Obtener
información de usuarios conectados
# Combinar la salida de los comandos en una variable
output = output_iface + output_users

```

Luego de guardar los resultados se crean variables las cuales serán utilizadas para la extracción de KPIs al momento de ejecutar el script, se los pone en 0 para que realicen inicialización y los códigos serían los siguientes

```

# Variables para almacenar
current_row = 2 # Rastrea la fila actual en la hoja de cálculo
iface_found = False # Indica si se ha encontrado información sobre una
interfaz
iface, desc, status, mtu, duplex, bw, input_rate, output_rate, in_error,
out_error, drops = None, None, None, None, None, None, None, None,
None, None, None
# Variables para almacenar información específica de cada interfaz,
inicializadas en None.
bw_giga, input_percent, output_percent = 0, 0, 0 # Variables para
almacenar información de ancho de banda y porcentajes de tasa,
inicializadas en 0.

```

Consecutivamente se crea los condicionantes para extraer los KPIs requeridos para almacenarlos en las variables las cuales deben cumplir con lo especificado; que luego se almacena en el archivo de Excel añadiendo fila por fila las interfaces de cada router y el código sería:

```

# Separar la salida del comando en líneas
lines = output.split('\n')
# Iterar a través de las líneas de la salida del comando
for line in lines:
    # Verificar si la línea contiene información sobre interfaces Ethernet
    if "Ethernet" in line or "FastEthernet" in line or "GigabitEthernet" in line:
        iface = line.split()[0] # Obtener el nombre de la interfaz
        iface_found = True # Indicar que se ha encontrado información sobre
una interfaz

        # Obtener la descripción de la interfaz si está presente
        if line.startswith(' Description:') and iface_found:
            desc = line.split("Description: ")[1]
        # Obtener el estado de la línea del protocolo de la interfaz
        if "line protocol" in line and iface_found:
            status = line.split()[-1]
        # Obtener información sobre el MTU y el ancho de banda de la interfaz
        if line.startswith(" MTU") and iface_found:
            mtu = line.split()[1]
            bw = line.split()[4]
            bw_giga = int(bw) / 1000000
        # Obtener información sobre el modo de duplex de la interfaz
        if "Duplex" in line and iface_found:
            duplex = line.split()[0]
        if "duplex" in line and iface_found:
            duplex = line.split()[0].replace(", ", "")
        # Calcular el porcentaje de tasa de entrada si está presente
        if line.startswith(" 5 minute input rate") and iface_found and bw_giga !=
0:
            input_rate = line.split()[4]
            input_percent = ((int(input_rate) / 1000000000) / bw_giga) * 100
        # Calcular el porcentaje de tasa de salida si está presente
        if line.startswith(" 5 minute output rate") and iface_found and bw_giga
!= 0:
            output_rate = line.split()[4]

```

```

        output_percent = ((int(output_rate) / 1000000000) / bw_giga) * 100
# Obtener información sobre errores de entrada si está presente
if "input error" in line and iface_found:
    in_error = line.split()[0]
# Obtener información sobre errores de salida si está presente
if "output error" in line and iface_found:
    out_error = line.split()[0]
# Obtener información sobre drops y actualizar la hoja de cálculo
if "bad etype drop" in line and iface_found:
    drops = line.split()[2]
# Agregar la información a la hoja de cálculo
worksheet.cell(row=len(worksheet['A']) + 1, column=1, value=hostname)
worksheet.cell(row=len(worksheet['B']), column=2, value=iface)
worksheet.cell(row=len(worksheet['C']), column=3, value=desc)
worksheet.cell(row=len(worksheet['D']), column=4, value=status)
worksheet.cell(row=len(worksheet['E']), column=5, value=mtu)
worksheet.cell(row=len(worksheet['F']), column=6, value=duplex)
worksheet.cell(row=len(worksheet['G']), column=7, value=bw_giga)
worksheet.cell(row=len(worksheet['H']), column=8, value=input_percent)
worksheet.cell(row=len(worksheet['I']), column=9,
value=output_percent)
worksheet.cell(row=len(worksheet['J']), column=10, value=in_error)
worksheet.cell(row=len(worksheet['K']), column=11, value=out_error)
worksheet.cell(row=len(worksheet['L']), column=12, value=drops)
# Reiniciar variables para la siguiente interfaz
current_row += 1
iface, desc, status, mtu, duplex, bw, input_rate, output_rate, in_error,
out_error, drops = None, None, None, None, None, None, None, None,
None, None, None
bw_giga, input_percent, output_percent = 0, 0, 0
iface

```

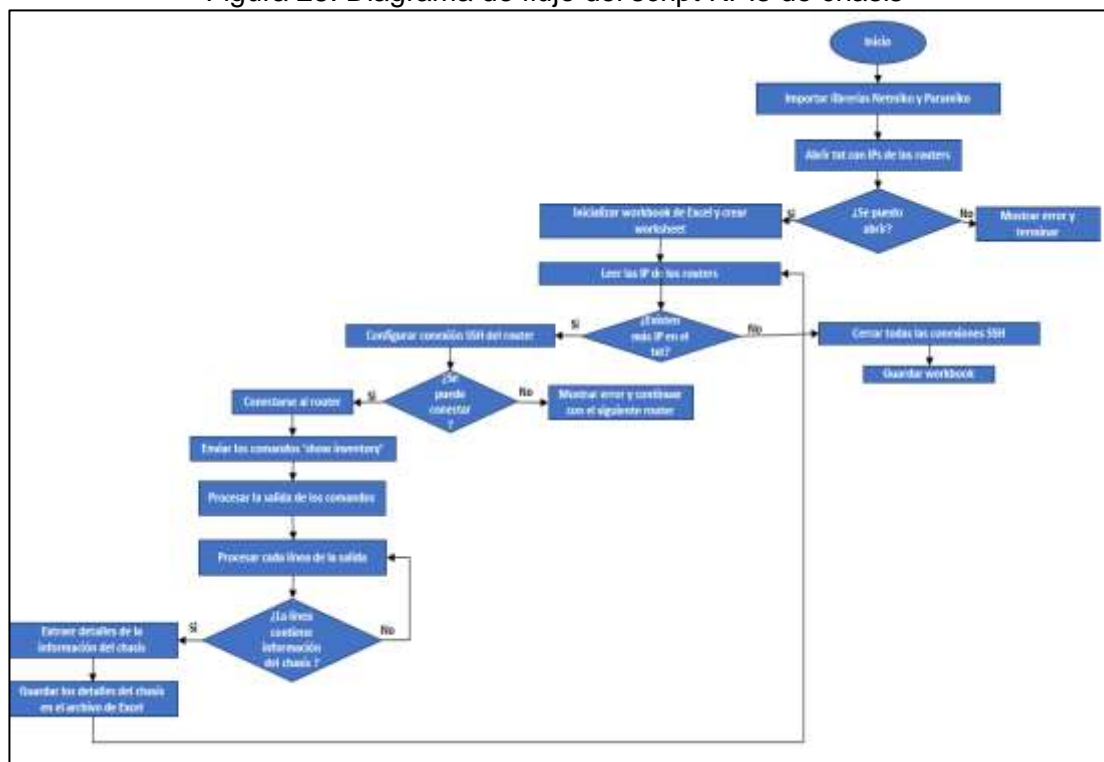
Luego de generar la hoja de Excel con los KPIs recopilados del router se ejecuta el siguiente comando para crear el libro de Excel y guardarlos con el nombre deseado usando este comando:

```
workbook.save('ROUTER_IFACE-BW-DSC-ERROR-INPUT/OUTPUT RATE.xlsx')
```

### 3.3.3. Script de KPIs de Chasis.

El script que se genera es para la extracción de características específicas de cada router, junto a su número serial y PID. Se busca reconocer las partes del elemento de forma física lo cual es muy factible al momento de su reparación o instalación. El script contiene diversas librerías para la conexión, extracción y procesamiento de datos de cada router. En la Figura 25 se observa el diagrama de flujo del script del KPI chasis.

Figura 25. Diagrama de flujo del script KPIs de chasis



Elaborado por: Autor

Es necesario conocer las librerías que se utilizarán y los comandos que se usarán para desplegar la información que se desea recopilar para su procesamiento, usamos el archivo de texto con las IPs loopback de los routers junto con su usuario y contraseña para acceder de forma remota, el formato es el siguiente:

```

# Importa la biblioteca Paramiko para establecer conexiones SSH
import paramiko
# Importa la biblioteca Pandas para manipulación de datos tabulares
import pandas as pd
# Define el comando a ejecutar en los routers y las cadenas a buscar en la
salida
cmd = "show inventory"
search_strings = ["NAME: ", "DESCR: ", "PID: ", "SN: "]
# Define el nombre del archivo que contiene las direcciones IP de los routers
filename = "ips_rt_cisco.txt"
# Lee las direcciones IP desde el archivo de texto y las almacena en la lista
'ips'
with open(filename) as f:
    ips = f.read().splitlines()
# Crea una lista vacía para almacenar la información de cada router
data = []
# Realiza la conexión y ejecuta el comando en cada router
for ip in ips:
    try:
        # Crea una conexión SSH
        client = paramiko.SSHClient()
        client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        client.connect(ip, username='david', password='2001')
        # Ejecuta el comando y lee la salida
        stdin, stdout, stderr = client.exec_command(cmd)
        output = stdout.readlines()
        # Procesa la salida para extraer información relevante

        Luego de recopilar la información y tenerla lista para su procesamiento,
        extraemos los KPIs por medio de condicionantes para consiguiente
        procesarlos en una hoja de Excel utilizando el siguiente código:
        # Busca las líneas que contienen la información que se desea extraer
        # Crea un diccionario para almacenar la información extraída, comenzando
        con la dirección IP del router

```

```

extracted_data = {"IP": ip}
# Itera sobre cada línea en la salida del comando ejecutado en el router
for line in output:
    # Itera sobre las cadenas de búsqueda definidas anteriormente
    for search_string in search_strings:
        # Verifica si la línea contiene la cadena de búsqueda
        if search_string in line:
            # Divide la línea en clave y valor utilizando la cadena de búsqueda
            key, value = line.split(search_string)
            # Remueve información adicional del valor, como espacios en blanco
            value = value.split(", ", 1)[0].strip()
            # Almacena la información en el diccionario extraído
            extracted_data[search_string.strip()] = value.strip()
        # Cuando se ha encontrado la información de todos los campos, agrega la
información a la lista
        if all(search_string.strip() in extracted_data for search_string in
search_strings):
            # Agrega la dirección IP al diccionario extraído
            extracted_data['IP'] = ip
            # Agrega una copia del diccionario extraído a la lista de datos
            data.append(extracted_data.copy())
            # Reinicia el diccionario para el próximo router
            extracted_data = {}
# Cierra la conexión SSH
client.close()
# Maneja excepciones relacionadas con la conexión SSH
except paramiko.AuthenticationException:
    print(f"La autenticación falló al conectar con {ip}")
    continue
except Exception as e:
    print(f"No se pudo conectar con {ip}: {e}")
    continue
# Si no se encontró información en ningún router
if not data:

```

```

print("No se encontró información en ningún router.")
# Convierte la lista en un DataFrame de pandas
df = pd.DataFrame(data)

```

Finalmente, luego de recopilar y procesar las variables para extraer los KPIs de cada router se procede a guardar el dataframe en un archivo de Excel por lo que se ejecuta el siguiente comando:

```

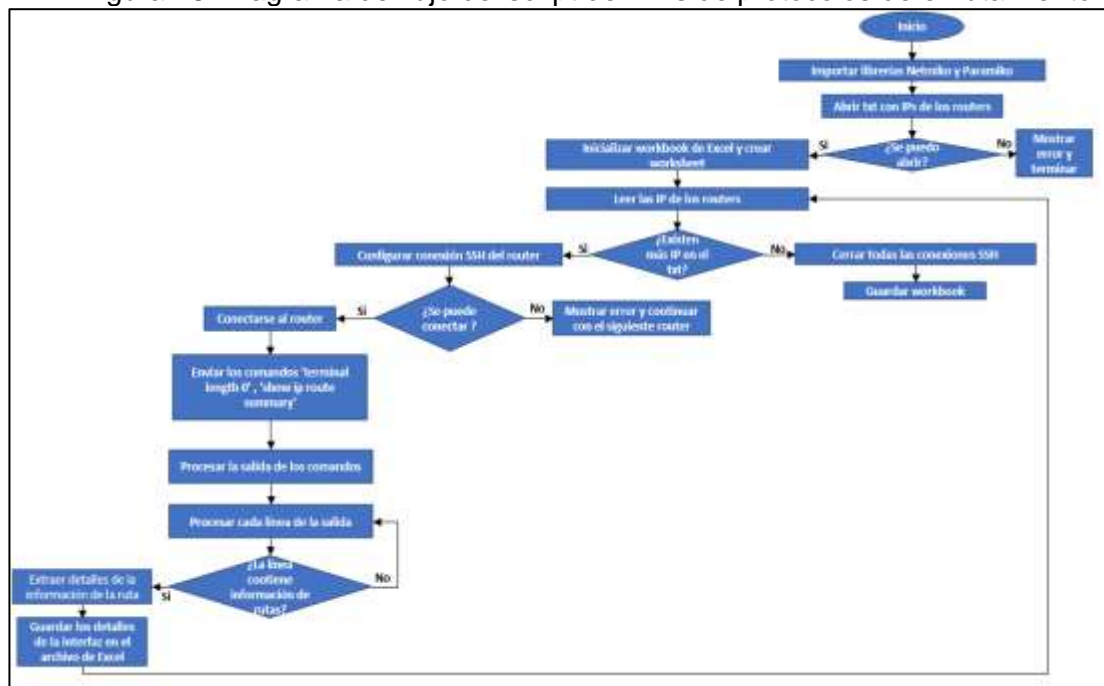
# Guardar el DataFrame en un archivo Excel
df.to_excel("HD_CHASSIS_CONTROL_LINE_SFP_INFO_CISCO_SW_280
22023_FV.xlsx", index=False)

```

### 3.3.4. Script de KPIs de protocolos de enrutamiento.

En este script se generará si una ruta es estática, directa o con algún protocolo de enrutamiento en este caso OSPF, con sus redes, subredes y algunas variables necesarias de mostrar, para ello es necesario analizar el proceso para la obtención de los datos. En la Figura 26 se observa el diagrama de flujo del script de los KPIs para protocolos de enrutamiento.

Figura 26. Diagrama de flujo del script de KPIs de protocolos de enrutamiento



Elaborado por: Autor

Para elaborar el script de los KPIs de protocolos de enrutamiento se debe tener en cuenta las librerías que se utilizarán, el usuario con su



contraseña, un archivo de texto en el que contenta las IPs Loopback de cada router para conectarse de manera remota, la estructura para acceso contiene los siguientes comandos:

```
# Importar las bibliotecas necesarias
from netmiko import ConnectHandler
import openpyxl

# Cargar los nombres de host/IP de los routers desde un archivo
with open('ips_rt_cisco.txt') as f:
    hostnames = f.read().splitlines()

# Inicializar el libro de trabajo de Excel
workbook = openpyxl.Workbook()
worksheet = workbook.active
worksheet.title = 'Routing_Info'

# Agregar encabezados de columna al archivo Excel
worksheet.append(['Router IP', 'Route Source', 'Networks', 'Subnets',
'Overhead', 'Memory (bytes)'])

# Configurar la conexión SSH con los routers
for hostname in hostnames:
    device = {
        'device_type': 'cisco_xe',
        'ip': hostname,
        'username': 'david',
        'password': '2001',
        'global_delay_factor': 2,
        'session_log': f"log_{hostname}.txt",
        'timeout': 60
    }
    try:
        # Intentar establecer conexión SSH con el router
        net_connect = ConnectHandler(**device)
        print(f"Conexión exitosa al router {hostname}")
        # Ejecutar comandos en el router para obtener información de rutas
        output = net_connect.send_command('terminal length 0')
        output_routes = net_connect.send_command('show ip route summary')
```

```

lines = output_routes.split("\n")
except Exception as e:
    print(f"Error al conectarse al router {hostname}: {e}")
    continue

```

Luego de guardar el resultado de los comandos ejecutados en cada router que se requiera en una variable, se procede a crear condicionantes para extraer los KPIs solicitados, que serán almacenados en nuevas variables a utilizar para posteriormente guardarlos en una hoja de Excel para ello se requiere estos códigos:

```
# Iterar a través de las líneas de la salida del comando
```

```
for line in lines:
```

```

    # Verificar si la línea contiene palabras clave "connected", "static", "ospf 1"
    if any(route in line for route in ["connected", "static", "ospf 1"]):

```

```

        # Dividir la línea en partes utilizando espacios en blanco como
separadores

```

```
        parts = line.split()
```

```
        # Comprobar que la línea contenga suficientes partes
```

```
        if len(parts) >= 5 and "ospf" not in line:
```

```
            route_source = parts[0] # Origen de la ruta
```

```
            networks = parts[1] # Número de redes
```

```
            subnets = parts[2] # Número de subredes
```

```
            overhead = parts[4] # Overhead
```

```
            memory = parts[5] # Memoria usada
```

```
            # Agregar la información de la ruta a la hoja de cálculo
```

```
            worksheet.append([hostname, route_source, networks, subnets,
overhead, memory])

```

```
        # Comprobar si la línea contiene "ospf" y ajustar el índice de las partes
```

```
        if len(parts) >= 5 and "ospf" in line:
```

```
            route_source = parts[0] # Origen de la ruta
```

```
            networks = parts[2] # Número de redes
```

```
            subnets = parts[3] # Número de subredes
```

```
            overhead = parts[5] # Overhead
```

```
            memory = parts[6] # Memoria usada
```

```

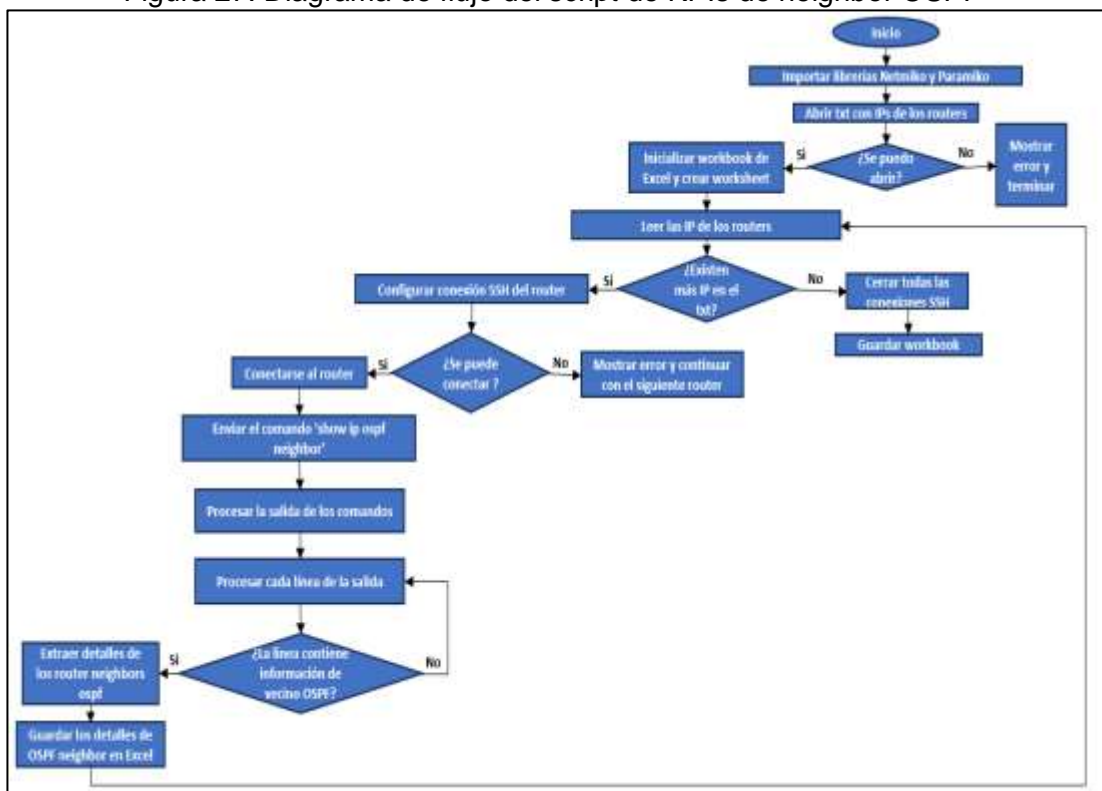
# Agregar la información de la ruta a la hoja de cálculo
worksheet.append([hostname, route_source, networks, subnets,
overhead, memory])
except Exception as e:
    print(f"Error al conectarse al router {hostname}: {e}")
try:
    # Guardar el libro de trabajo de Excel
    workbook.save('ROUTER_ROUTING_INFO_CISCOXR.xlsx')
except Exception as e:
    print(f"Error al guardar el libro de trabajo de Excel: {e}")

```

### 3.3.5. Script de KPIs de neighbor OSPF.

En el script que se crea, se busca observar y determinar los routers vecinos de cada router, junto a su interface, su IP de red, prioridad los cuales muestran cómo está configurado la base de datos de enrutamiento. Para ello es necesario conocer cómo se ejecuta el proceso de la extracción de los KPIs, en la Figura 27 se muestra el diagrama de flujo del script de KPIs de routers neighbor en el protocolo de enrutamiento OSPF.

Figura 27. Diagrama de flujo del script de KPIs de neighbor OSPF



Elaborado por: Autor

Para extraer los datos de los routers vecinos OSPF mediante un script es necesario tener la red configurada y que todo funcione correctamente, para posteriormente importar librerías, el usuario con su contraseña, un archivo de texto con las IPs Loopback de todos los routers y las variables que se guardarán al ejecutar el comando en el router, por ende, se escribe la siguiente parte del código:

```
# Importar las bibliotecas necesarias
from netmiko import ConnectHandler
import openpyxl

# Cargar los nombres de host/IP de los routers desde un archivo
ips_rt_cisco.txt
with open('ips_rt_cisco.txt') as f:
    hostnames = f.read().splitlines()
# Inicializar el libro de trabajo de Excel
workbook = openpyxl.Workbook()
worksheet = workbook.active
worksheet.title = 'Neighbors'
# Agregar encabezados de columna
worksheet.append(['Router IP', 'Neighbor ID', 'Pri', 'State', 'Dead Time',
'Address', 'Interface'])
# Configurar la conexión SSH con los routers
for hostname in hostnames:
# Definir los parámetros de conexión para cada dispositivo (router)
    device = {
        'device_type': 'cisco_xe', # Tipo de dispositivo Cisco XE
        'ip': hostname, # Dirección IP del router
        'username': 'david', # Nombre de usuario para la conexión SSH
        'password': '2001', # Contraseña para la conexión SSH
        'global_delay_factor': 2, # Factor de retardo global para la conexión
        'timeout': 40 # Tiempo máximo de espera para la conexión
    }
# Intentar establecer la conexión SSH con el router
    try:
        net_connect = ConnectHandler(**device)
```

```

    print(f"Conexión exitosa al router {hostname}")
except Exception as e:
    print(f"Error al conectarse al router {hostname}: {e}")
    continue # Pasar al siguiente router en caso de error
# Ejecutar el comando 'show ip ospf neighbor' en el router y obtener el
resultado
output = net_connect.send_command('show ip ospf neighbor')

```

Una vez establecida la conexión se debe utilizar condicionantes como if, for, any entre muchos más para poder extraer los KPIs de protocolos de enrutamiento y almacenarlos en variables las cuales se mostrarán en la hoja de Excel en el orden solicitado para ello es necesario realizar los siguientes códigos en el script :

```

# Procesar la salida del comando y actualizar el libro de trabajo de Excel
lines = output.split('\n') # Dividir la salida en líneas
for line in lines:
    if 'Neighbor ID' in line:
        continue # Saltar la línea de encabezado
    # Dividir la línea en columnas
    columns = line.split()
    if len(columns) >= 6:
        # Extraer información de las columnas
        neighbor_id, priority, state, dead_time, address, interface = columns[:6]
        # Agregar la información a la hoja de trabajo de Excel
        worksheet.append([hostname, neighbor_id, priority, state, dead_time,
address, interface])
    # Cerrar la conexión SSH
    net_connect.disconnect()

# Una vez creada la hoja de Excel con los KPIs correspondientes, se genera
el libro de Excel.
# Para ello se utiliza esta línea de comando:
# Guardar el libro de trabajo de Excel
workbook.save('Neighbors_Info.xlsx')

```

### 3.4. Creación y análisis de dashboards con Tableau.

El programa Tableau ofrece muchas herramientas y facilita la creación de dashboards, para ello, se necesita tener los archivos de Excel con los KPIs solicitados para poder adjuntarlo y generar tablas o gráficos estadísticos de acuerdo con los datos requeridos, obteniendo una forma más entendible para el análisis de la red.

#### 3.4.1. Dashboard de KPIs de cpu y memoria.

Para crear un dashboard de los KPIs de cpu y memoria, se tiene que extraer y generar un Excel por medio de Python en este escenario se dejó corriendo la simulación por 2 horas y 54 minutos de lo cual se obtiene el siguiente Excel que se muestra en la Figura 28 las variables necesarias.

Figura 28. Excel de los KPIs de memoria y cpu

	A	B	C	D	E	F
1	Router IP	CPU % USE	MEM_USED	UPTIME	MODEL	VERSION
2	1.1.1.2	1	46,987121	2 hours, 54 minutes	7200	15.2(4)S5
3	1.1.1.3	3	47,030976	2 hours, 54 minutes	7200	15.2(4)S5
4	1.1.1.4	3	46,988465	2 hours, 54 minutes	7200	15.2(4)S5
5	1.1.1.5	2	46,97527	2 hours, 54 minutes	7200	15.2(4)S5
6	1.1.1.6	3	46,994816	2 hours, 54 minutes	7200	15.2(4)S5
7	1.1.1.7	1	46,976865	2 hours, 54 minutes	7200	15.2(4)S5
8	1.1.1.8	2	46,988297	2 hours, 54 minutes	7200	15.2(4)S5

Elaborado por: Autor.

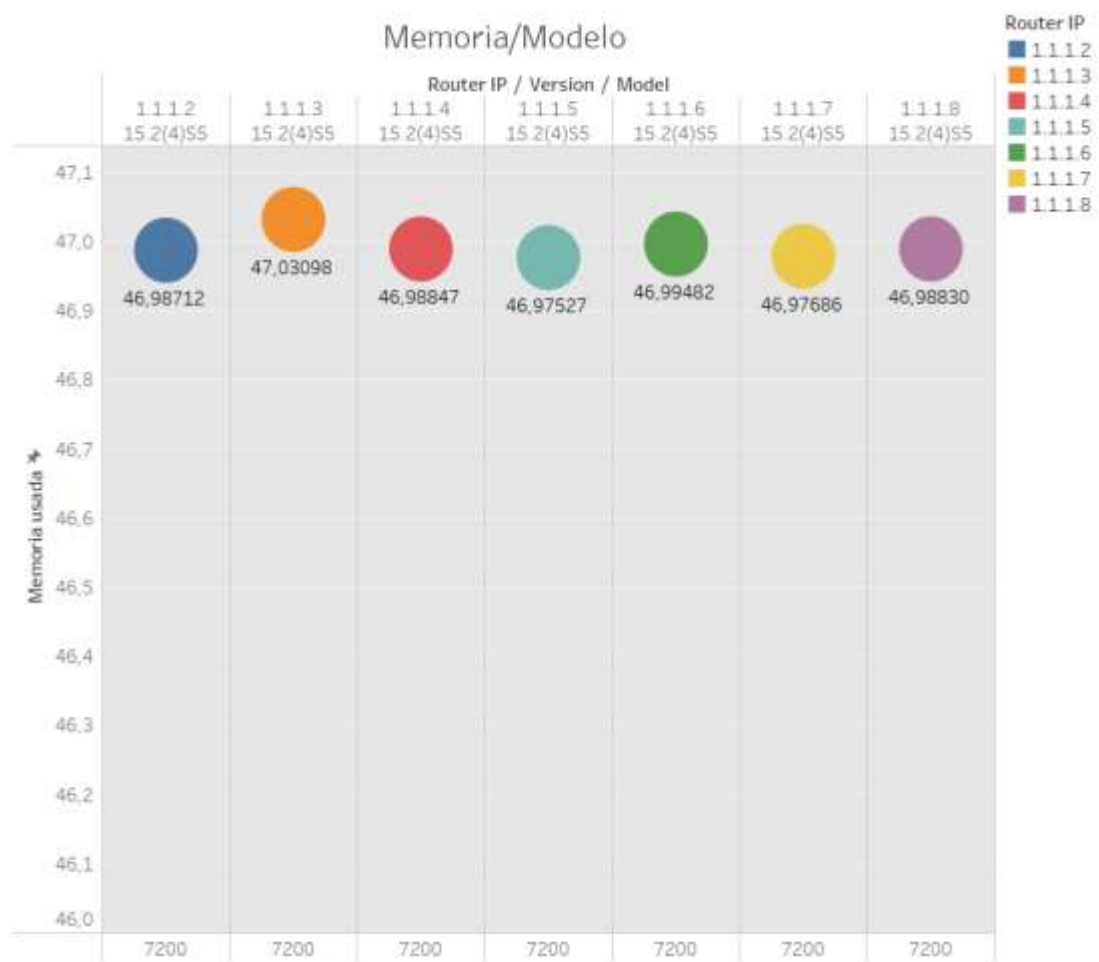
Luego de obtener el Excel con los KPIs requeridos se tiene el modelo, cpu, memoria utilizada, uptime de los routers solicitados se lo carga en Tableau para la creación de dashboards, realizando los siguientes pasos:

- Al abrir Tableau y dar clic en la pestaña archivo y luego en nuevo, para después seleccionar el Excel que contiene los KPIs.
- Dar formato número entero y clic derecho en CPU para crear un grupo calculado. Dar clic opción Hoja 1, para mostrar las variables.
- Para el primer dashboard el cual es de Memoria/Modelo se utiliza en la parte de filas la variable SUMA(Mem Used) y en la parte de columna de adjunta las variables Router IP, Version y Model.
- Para el segundo dashboard el cual es del %CPU usado y uptime, se utiliza en la parte de columnas la variable Router IP y Uptime mientras que en la fila es SUMA(%Cpu usado) .

- Luego se escoge en la parte de Mostrarme el tipo de grafica a utilizar, después en parte de marcas se elige el tipo de figura, el color y dar clic derecho a cada elemento de espacio para darle formato.

Después de crear los dashboards se realiza su respectivo análisis en la primera imagen que es la 29 se observa como todos los routers pertenecen al mismo modelo y versión. También se puede analizar que los routers RTGYE01 y el RTUIO01 poseen mayor memoria usada debido a que son los routers que conectan las dos redes.

Figura 29. Dashboard de Memoria y Modelo



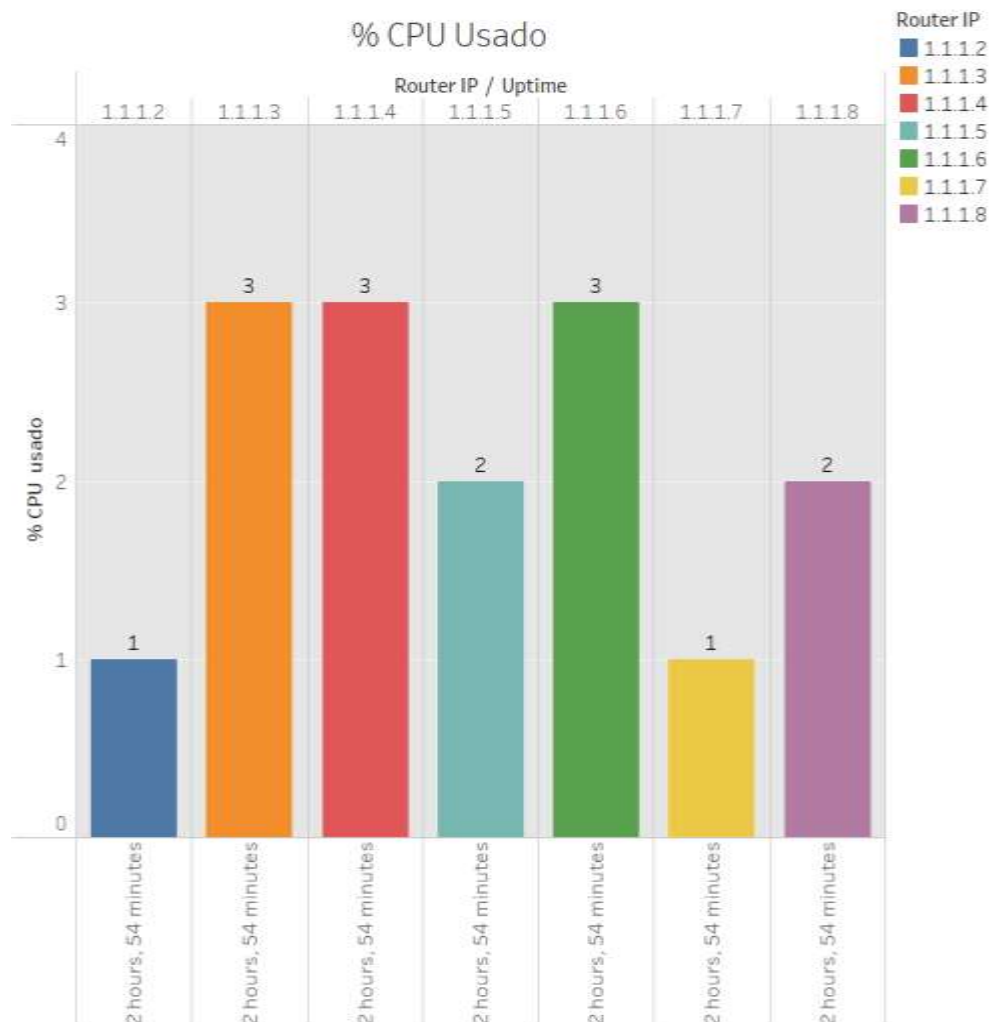
Elaborado por: Autor.

En el segundo dashboard que se visualiza en la Figura 3.9, se puede analizar que todos los routers tiene el mismo uptime, es decir ninguno se apagó y fueron encendidos al mismo tiempo, este KPI es de mucha importancia porque puede ocurrir errores como desconexión temporal del equipo, algún reinicio forzado o problemas eléctricos los cuales son muy

importantes reconocer al momento del análisis, también conocer el modelo del router para entender sus características y capacidades al momento de buscar repuestos o implementar alguna función.

Luego se analiza el CPU utilizado, el cual muestra que el RTGYE01, RTDUR01 y el RTUIO01 tienen 1% más que los demás routers, lo que da a concluir que el tráfico y el trabajo realizado por la red pasa más por esos routers, es decir utilizan más recursos en la Figura 30 se logra visualizar los datos requeridos.

Figura 30. Dashboard de %CPU usado y Uptime



Elaborado por: Autor.

### 3.4.2. Dashboards de KPIs de interfaces.

Para crear los dashboards de los KPIs de las interfaces, primero se debe crear los 2 escenarios los cuales afectarán a la interfaces de los routers. En el primer escenario se enviará por 15 minutos y por medio de 2 end devices



Ipterm paquetes UDP de 10 Megabytes por segundo los cuales generarán tráfico en las interfaces de los routers. En la Figura 31 se observa el archivo Excel con los KPIs extraídos del primer escenario.

Figura 31. Excel de los KPIs de %CPU usado y Uptime

	A	B	C	D	E	F	G	H	I	J	K	L
1	Router IP	IFACE	DESC	STATUS	MTU	DUPLEX	BW_GIGA	INPUT_RA	OUTPUT_E	INPUT_ERROR	OUTPUT_E	DROPS
2	1.1.1.2	Ethernet0/0		down	1500		0,01	0	0	0	0	
3	1.1.1.2	GigabitEthernet0/0		down	1500	Full	1	0	0	0	0	
4	1.1.1.2	GigabitEth Connectio		up	1500	Full	1	0	0	0	0	
5	1.1.1.2	GigabitEthernet2/0		down	1500	Full	1	0	0	0	0	
6	1.1.1.2	GigabitEth Connectio		up	1500	Full	1	0	0	0	0	
7	1.1.1.2	GigabitEth Connectio		up	1500	Full	1	0,0001	0,0001	0	0	
8	1.1.1.2	FastEthernet5/0		down	1500	Full-duple>	0,1	0	0	0	0	
9	1.1.1.2	FastEthern Connectio		up	1500	Full-duple>	0,1	0	0	0	0	
10	1.1.1.2	GigabitEthernet6/0		down	1500	Full	1	0	0	0	0	
11	1.1.1.3	Ethernet0/0		down	1500		0,01	0	0	0	0	
12	1.1.1.3	GigabitEth Connectio		up	4000	Full	1	0,0002	0,9629	0	0	
13	1.1.1.3	GigabitEthernet1/0		down	1500	Full	1	0	0	0	0	

Elaborado por: Autor.

Luego de extraer los KPIs en un Excel del primer escenario se procede por medio de Tableau a generar los dashboards necesarios para representar los parámetros de las interfaces; para ello se ejecuta los siguientes pasos :

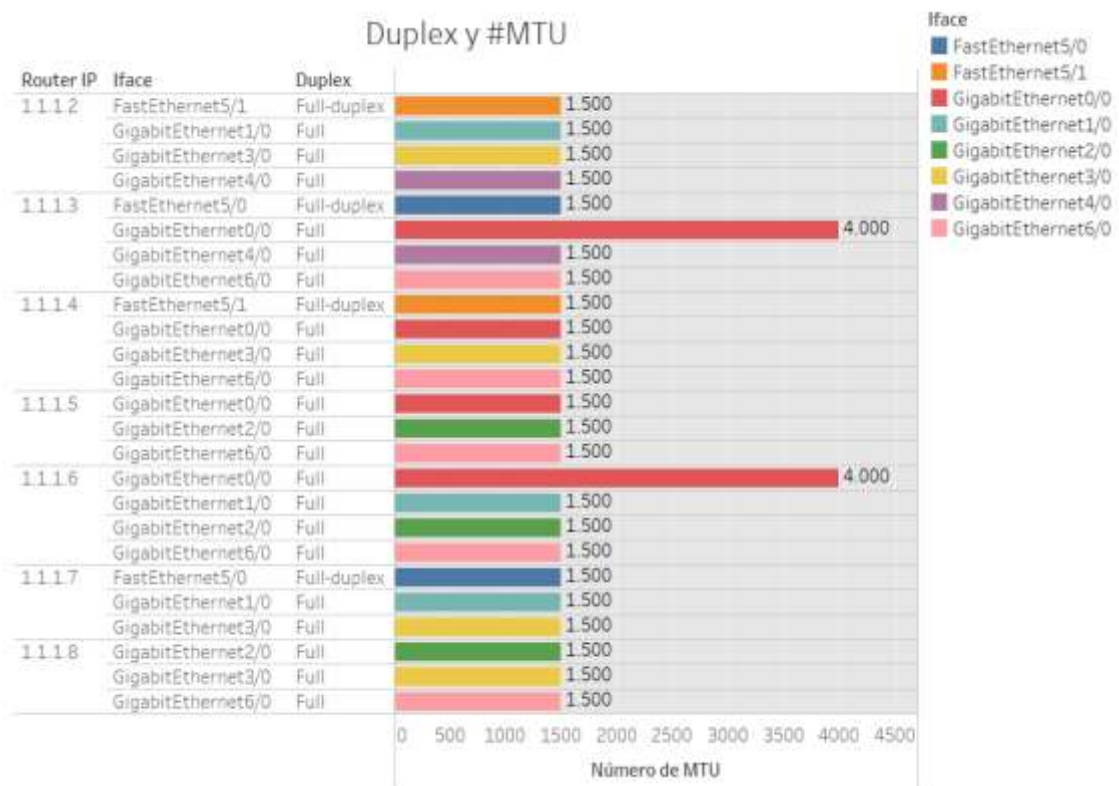
- Al abrir el programa Tableau se da clic en la pestaña archivo y luego nuevo, consecutivamente seleccionar el Excel que se tienen los KPIs.
- Luego dar formato a las variables como número entero y dar clic derecho en MTU y elegir la opción crear grupo calculado, al igual que los demás datos de input/output rate, input/output errors. Luego cargar los datos, dar clic opción Hoja 1 , para mostrar las variables.
- Se considera realizar 4 dashboards referentes a cada medida.
- Para el primero se crea para especificar si la interface es half/full duplex y la cantidad de MTU que tiene la interface por donde pasa el tráfico, en la parte de columnas adjuntar el elemento Suma(Número de MTU) mientras que en la fila se adjunta Router IP, Iface, y Duplex.
- Para el segundo dashboard se crea para mostrar el ancho de banda de las interfaces del router activas, para ello en la parte de columnas se pone la variable Router IP, Iface. En las filas se utiliza las variables Suma(Bw Giga) y la de Status la cual se le debe aplicar el filtro de up.
- Para el tercer dashboard se habla del input y output rate % de cada interface activa del router, en la columna se adjunta las variables

Router IP, Iface y Status la cual debe tener el filtro de up. En las filas se adjunta las variables Suma(%Input rate) y Suma(%Output rate).

- Para el cuarto dashboard se crea para mostrar el input error, por lo cual en la parte de columna se adjunta las variables Suma(Número de Input) y Status el cual debe tener el filtro de up.
- Dar clic en Mostrarme para escoger el tipo de gráfico, se procede a elegir la forma en el cuadro Marcas, dar clic derecho las parte del gráfico para editar fuentes, textos, sombreado, etc.

Al extraer los dashboard se puede realizar su respectivo análisis. En la Figura 32 que es el primer dashboard que es el de Duplex full y MTU, se observa que las interfaces de los routers tienen un MTU de 1500, mientras que la interface que une los routers RTGYE01 y RTUIO01 tienen un MTU de 4000, considerando que ese enlace une 2 redes por ende debe ser mayor.

Figura 32. Dashboard de MTU y Duplex

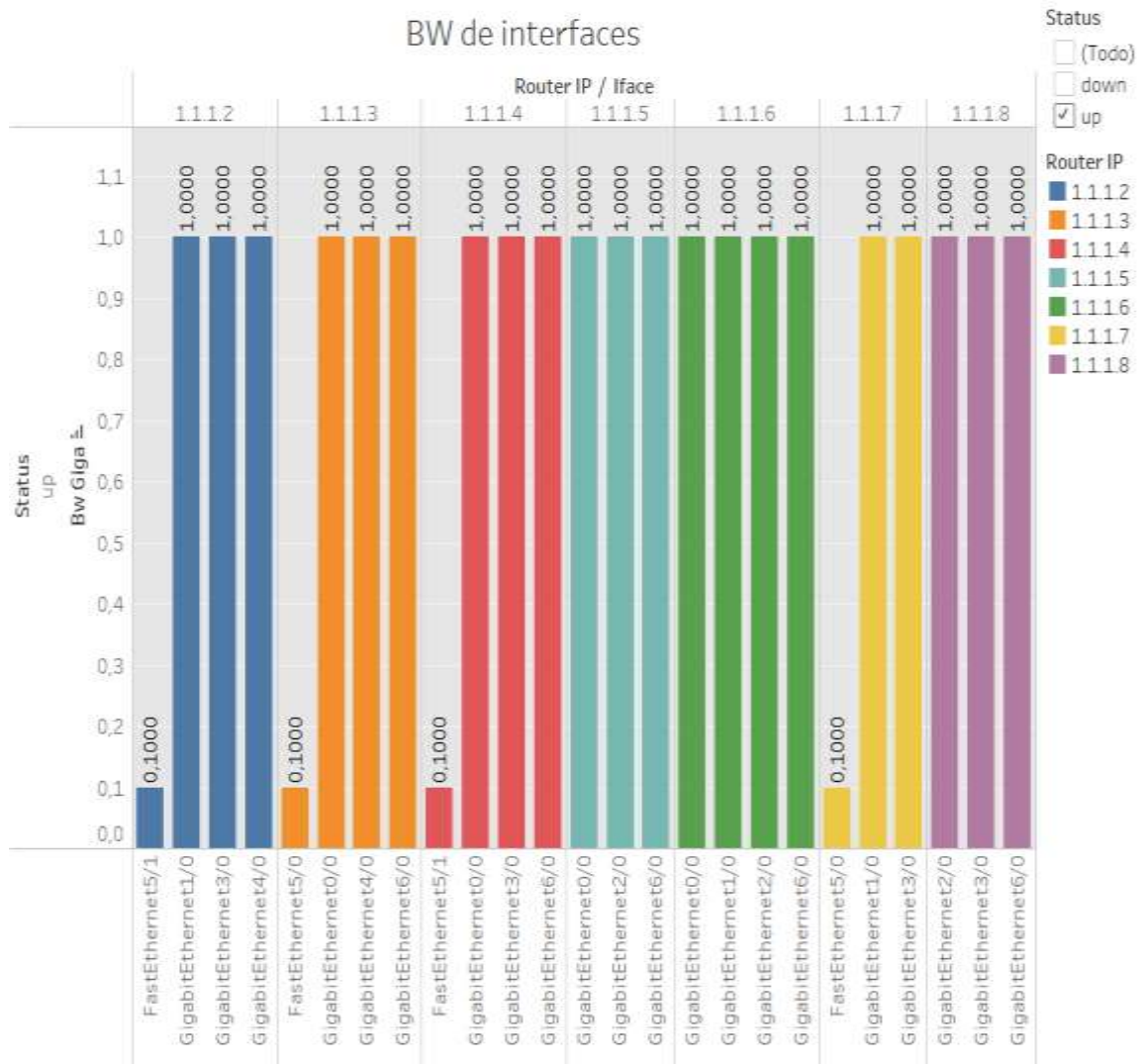


Elaborado por: Autor

En el segundo dashboard que se observa en la Figura 33 presenta la cantidad de ancho de banda de las interfaces activas, se muestra que las interfaces FastEthernet transformadas en Gigabits están configuradas con un

ancho de banda de 0,100 Gigabits/segundo lo que en Megabits sería 100 Megabits/segundo, mientras que las interfaces GigabitEthernet poseen un ancho de banda de 1 Gigabit/segundo, por ende, la interface FastEthernet posee menor ancho de banda que una de GigabitEthernet.

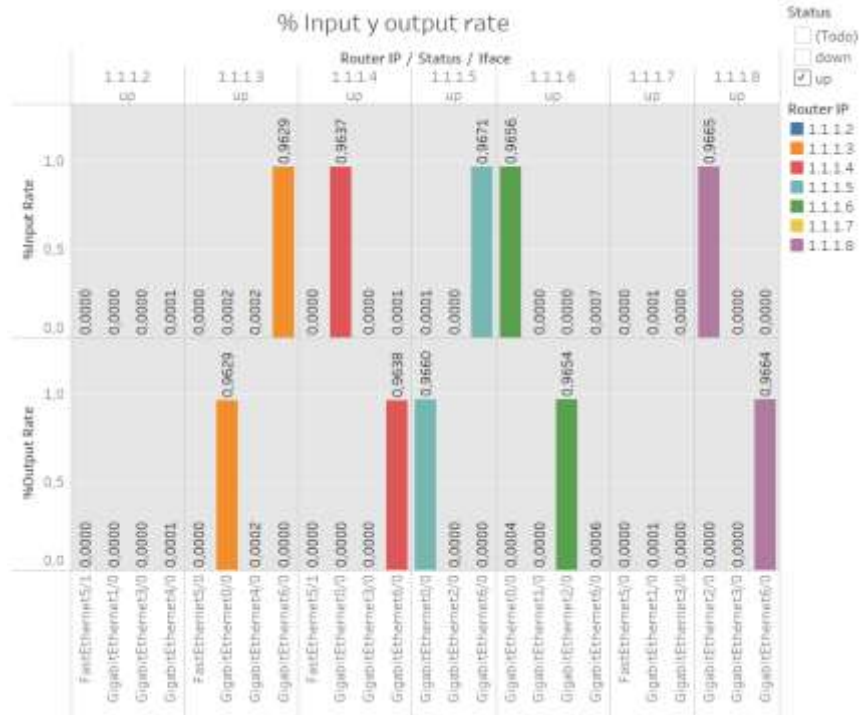
Figura 33. Dashboard de Ancho de Banda



Elaborado por: Autor.

En el tercer dashboard que se observa en la Figura 34 se logra analizar que el input rate indican la cantidad de datos que ingresan en las interfaces en un determinado período, en este caso, cada 5 minutos, mientras que el output rate indica la tasa de datos que sale de las interfaces en un determinado período de 5 minutos, se puede observar como la red genera la ruta para llevar los datos, también se puede detallar que las otras interfaces indican una ligera cantidad, posiblemente al ejecutar el Python ingresen los datos por esas interfaces.

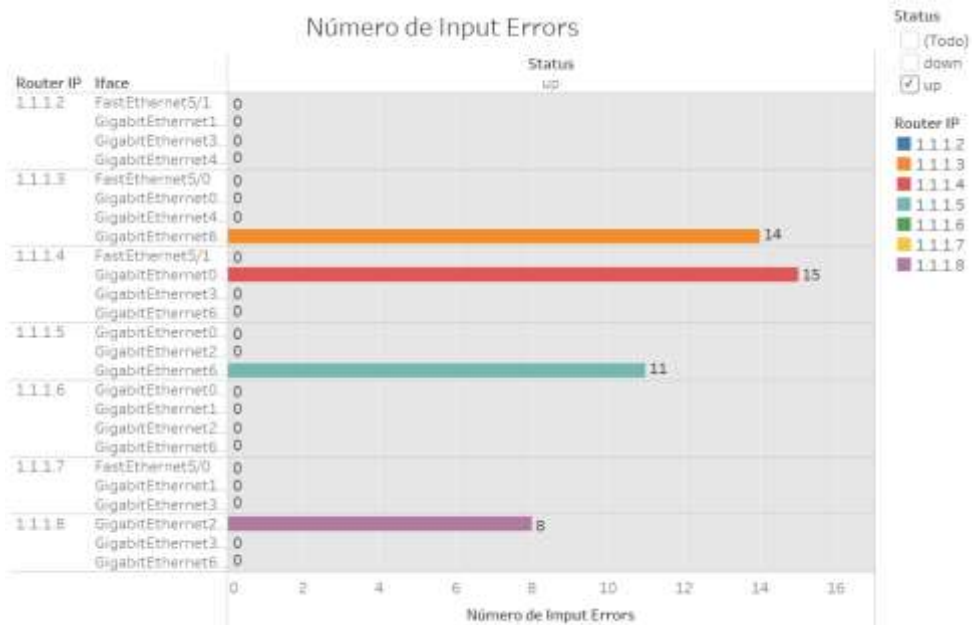
Figura 34. Dashboard de Ruta 1 % Input y Output rate



Elaborado por: Autor.

En el cuarto dashboard que se visualiza en la Figura 35 se analiza el input errors , en este caso no incluye drops ni output errors porque salieron 0 ya que es un simulador, el input error es bajo por ende se concluye que afecta de manera ligera al input y output rate en las interfaces, posiblemente por problemas del enlace, malas conexión, configuración fallida, etc.

Figura 35. Dashboard de Ruta 1 Número de Input Errors



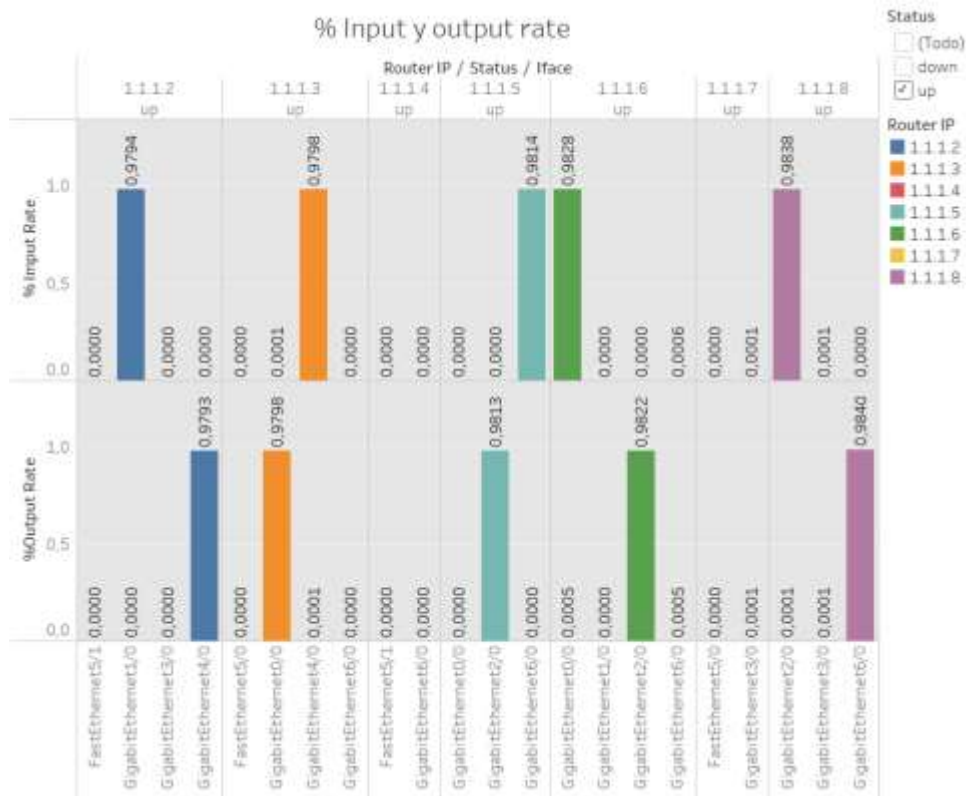
Elaborado por: Autor.

Al concluir el primer escenario se realiza los dashboards del segundo escenario. El segundo escenario se divide en 3 situaciones las cuales se determinan por las 3 rutas que se generó al extraer los KPIs en Python y generar un Excel. Se utiliza el mismo método para crear dashboards usado en el anterior escenario. En este escenario se analiza si realiza el cambio de enrutamiento dinámico por OSPF y también KPIs de input/output error y rate, en todas las situaciones el MTU, ancho de banda serán las misma.

La primera situación fue creada en el escenario anterior por ende se analiza las rutas escogidas por el protocolo OSPF, el tráfico va por el RTYGU01, luego al RTMLG01 y después al RTGYE01, el cual se enlaza con el RTUIO01, que elige el camino del RTRIO01 enviando la data a su destino.

En la segunda situación que se refiere la ruta 2 para verificar que el tráfico se envié correctamente por la interface. En el primer dashboard se puede analizar el input/output rate se ejecuta correctamente, mientras que las otras interfaces activas poseen lo mínimo por lo que se concluye que ahí paso un tráfico mínimo de Python, lo cual se observa en la Figura 3.6.

Figura 36. Dashboard de Ruta 2 % Input y Output rate.

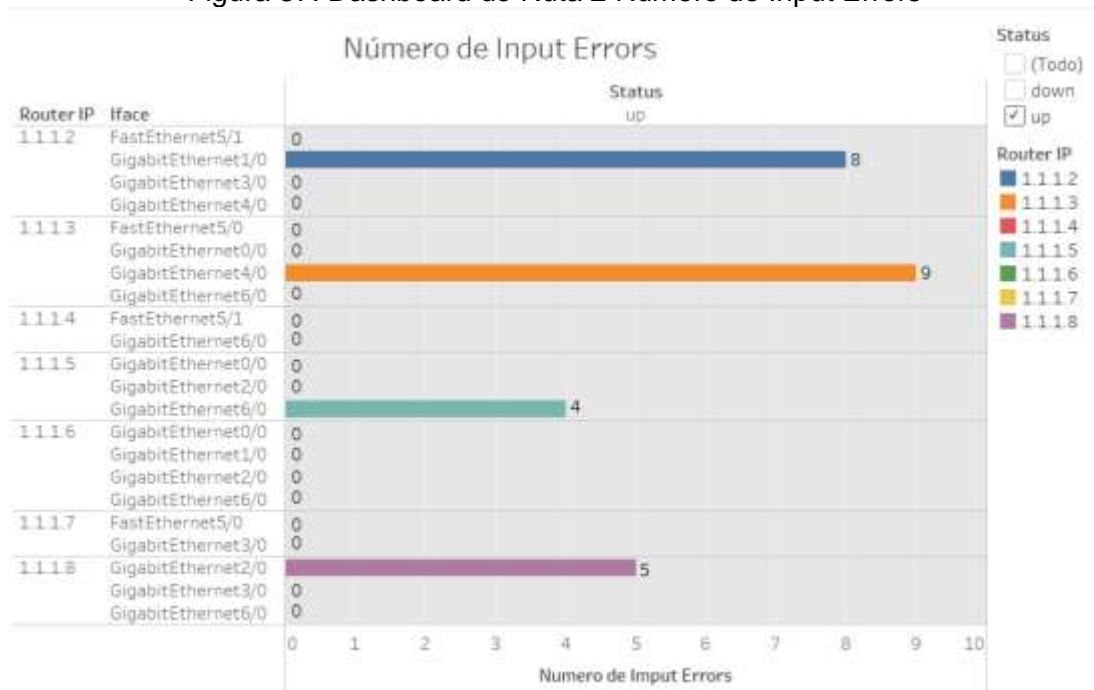


Elaborado por: Autor.

Las rutas escogidas para la ruta 2 es que el tráfico de lpterm salga hacia el RTYGU01 el cual envía la data al RTDUR01 quien se conecta con el RTGYE01, este router conecta esa red con la otra por medio del enlace con RTUIO01, el RTUIO01 se conecta con el RTRIO01 el cual envía el tráfico a su destino es decir el lpterm 2.

En la Figura 37 se analiza los input errors de las interfaces de la ruta 2 las cuales muestran una disminución en los inputs errors de cada interface de los equipos de red es decir los routers. La interface RTGYE01 es donde más ocurren errores y en el RTMLG01 por lo que esto afectará de forma insignificante los input/ output rate.

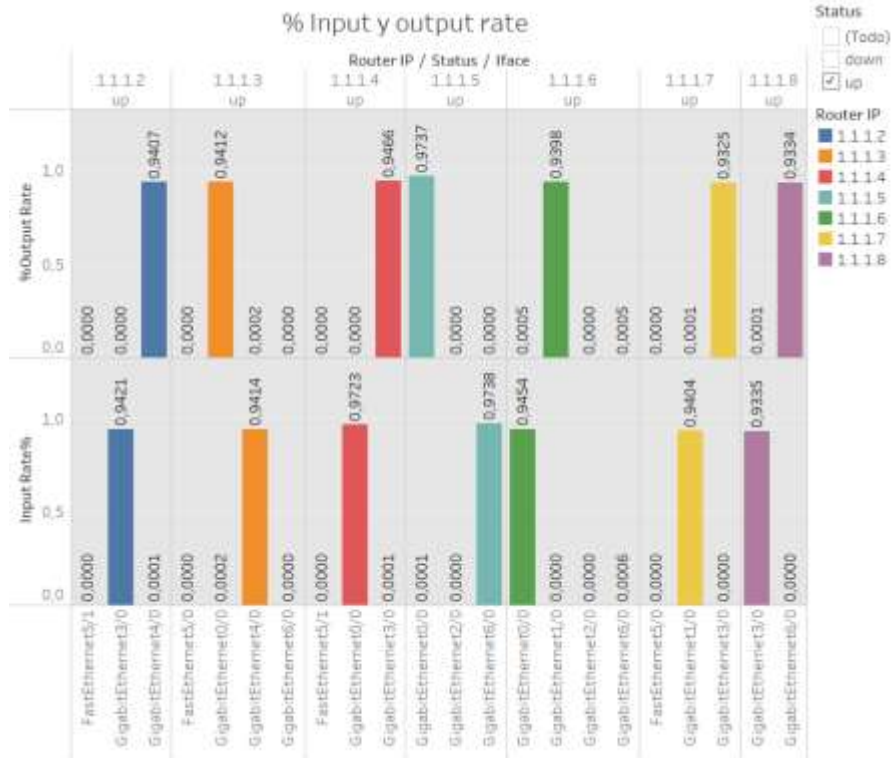
Figura 37. Dashboard de Ruta 2 Número de Input Errors



Elaborado por: Autor.

Luego de realizar la situación 2 del escenario 2 se procede a analizar la situación 3 , donde se observa en la Figura 38, donde se analiza que los caminos tomados, ya que se los ha forzado a que tomen esos saltos, el tráfico del lpterm 1 se dirige al RTYGU01 y luego al RTMLG01 el cual al no tener conexión con el RTGYE01 envía el tráfico al RTDUR01 que envía al RTGYE01, que tiene un enlace con el RTUIO01, el cual no posee conexión con el RTRIO01 entonces debe dar un salto al RTCNC01 el cual lleva la data al RTRIO01, el cual es el router que lleva los paquetes a su destino.

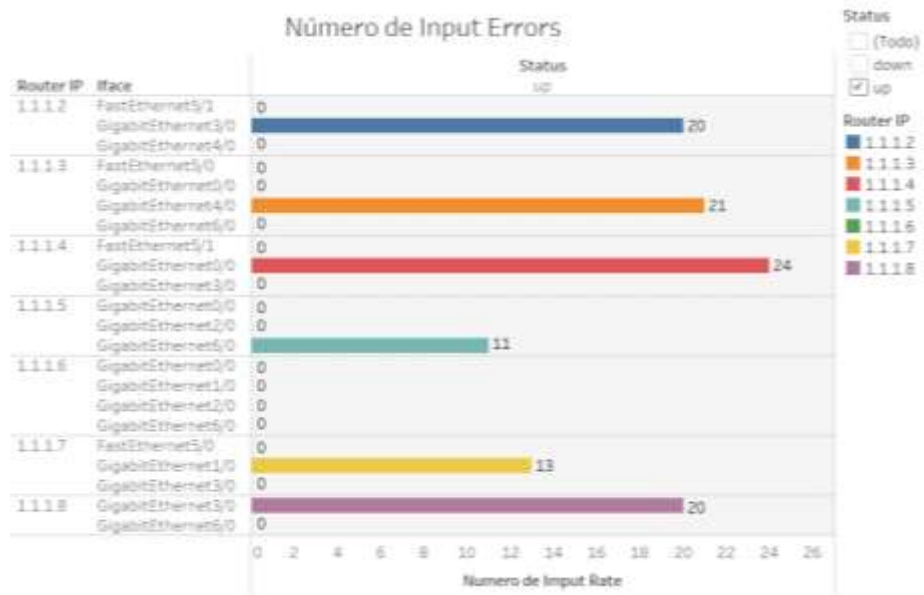
Figura 38. Dashboard de Ruta 3 % Input y Output rate



Elaborado por: Autor.

Se observa las interfaces con la tasa de input y output al tomar esa ruta, asegurando que el trayecto es el correcto. En el segundo dashboard se analiza el input error es decir que existen errores en la entrada de las interfaces al momento de enrutar los datos, en la Figura 39 se logra visualizar el número de input error por interface de cada router.

Figura 39. Dashboard de Ruta 3 Número de Input Errors



Elaborado por: Autor.

### 3.4.3. Dashboards de KPIs de Chasis.

Todo router o equipo de red tiene su chasis correspondiente, lo cual es de gran utilidad al momento de realizar un inventario, ejecutar algún cambio al tener algún defecto o también tener una descripción específica de lo que ofrece el producto que se está utilizando para ello se realiza un dashboard en Tableau para obtener el inventario, para ello es necesario tener los datos de los KPIs extraídos en un Excel como se muestra en la Figura 40.

Figura 40. Excel de los KPIs de las Interfaces

	A	B	C	D	E
	IP	NAME:	DESCR:	PID:	SN:
2	1.1.1.2	"Chassis"	"Cisco 7206VXR"	CISCO7206VXR	4279256517
3	1.1.1.2	"NPE400 0"	"Cisco 7200VXR Network Processing Engine NPE-400"	NPE-400	11111111
4	1.1.1.2	"module 0"	"I/O Ether/GigEther Controller"	C7200-I/O-2FE/E	00000000
5	1.1.1.2	"module 1"	"GigabitEthernet"	PA-1GE=	0
6	1.1.1.2	"module 2"	"GigabitEthernet"	PA-1GE=	0
7	1.1.1.2	"module 3"	"GigabitEthernet"	PA-1GE=	0
8	1.1.1.2	"module 4"	"GigabitEthernet"	PA-1GE=	0
9	1.1.1.2	"module 5"	"Dual Port FastEthernet (RJ45)"	PA-2FE-FX	XXXX00000000
10	1.1.1.2	"module 6"	"GigabitEthernet"	PA-1GE=	0
11	1.1.1.2	"Power Supply 0"	"Cisco 7200 AC Power Supply"	PWR-7200-AC	
12	1.1.1.2	"Power Supply 1"	"Cisco 7200 AC Power Supply"	PWR-7200-AC	
13	1.1.1.3	"Chassis"	"Cisco 7206VXR"	CISCO7206VXR	4279256517
14	1.1.1.3	"NPE400 0"	"Cisco 7200VXR Network Processing Engine NPE-400"	NPE-400	11111111
15	1.1.1.3	"module 0"	"I/O Ether/GigEther Controller"	C7200-I/O-2FE/E	00000000
16	1.1.1.3	"module 1"	"GigabitEthernet"	PA-1GE=	0
17	1.1.1.3	"module 2"	"GigabitEthernet"	PA-1GE=	0
18	1.1.1.3	"module 3"	"GigabitEthernet"	PA-1GE=	0
19	1.1.1.3	"module 4"	"GigabitEthernet"	PA-1GE=	0
20	1.1.1.3	"module 5"	"Dual Port FastEthernet (RJ45)"	PA-2FE-FX	XXXX00000000
21	1.1.1.3	"module 6"	"GigabitEthernet"	PA-1GE=	0
22	1.1.1.3	"Power Supply 0"	"Cisco 7200 AC Power Supply"	PWR-7200-AC	
23	1.1.1.3	"Power Supply 1"	"Cisco 7200 AC Power Supply"	PWR-7200-AC	
24	1.1.1.4	"Chassis"	"Cisco 7206VXR"	CISCO7206VXR	4279256517
25	1.1.1.4	"NPE400 0"	"Cisco 7200VXR Network Processing Engine NPE-400"	NPE-400	11111111
26	1.1.1.4	"module 0"	"I/O Ether/GigEther Controller"	C7200-I/O-2FE/E	00000000
27	1.1.1.4	"module 1"	"GigabitEthernet"	PA-1GE=	0

Elaborado por: Autor

Cuando se obtiene el Excel al correr el script de Python se procede a realizar el gráfico en el entorno de Tableau para ello es necesario que se ejecute los siguientes pasos:

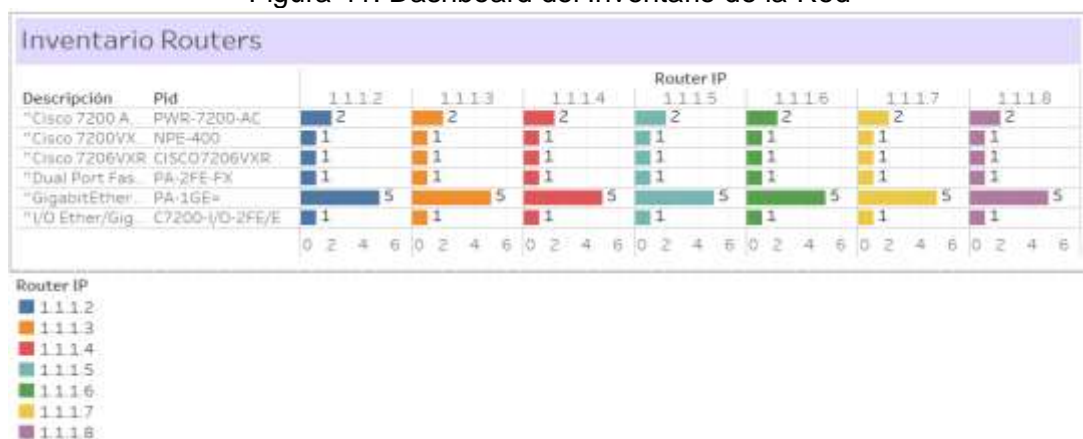
- En el entorno de trabajo de Tableau se presiona en la pestaña archivo, consecutivamente en la opción nuevo.
- Se selecciona el archivo de Excel que se obtuvo del chasis en Python para que carguen los datos necesarios.
- Luego de cargar los datos, se dirige a la opción Hoja 1, la cual muestra las variables cargadas y el entorno para crear dashboard junto a sus diseños de gráficos.
- La información que se utilizará es la del conteo de elementos (Tableau lo realizó por defecto), la IP del router, la descripción y el PID de elemento.



- Luego de determinar las variables las arrastramos de acuerdo como se quiere plasmar estadísticamente, en las columnas se adjuntará el conteo de elementos de cada router (Sheet1 (Recuento)) y su respectiva IP. En las filas se plasmará la descripción y el PID para poder reconocer de forma específica.
- Para la forma de representación estadístico, se debe presionar en el cuadro marcas la opción automático donde se desplegará todas las opciones de gráficas en este caso de barras junto al color a elegir, luego ir al cuadro Mostrarme para elegir el modelo estadístico a presentar en este caso se elige barras invertidas.
- Una vez conseguido el gráfico se presiona clic derecho sobre cada área del gráfico, para obtener la opción de cambiar el color, tipo de letra, negrita, subrayado entre otras.

Una vez obtenido el gráfico final se realiza el siguiente análisis. En la Figura 41 se logra visualizar que cada router tiene 2 Cisco 7200 AC Power Supply la cual es una fuente de corriente alterna para series de Cisco 7200. Se encuentran también 5 módulos de interfaz Gigabitethernet. Se visualiza un módulo de procesamiento de red Cisco 7200VXR Network Processing Engine NPE-400, tiene un módulo Dual Port FastEthernet (RJ45) que tiene 2 puertos correspondientes a uno de RJ45 y uno FastEthernet. También se encuentra el módulo I/O Ether/GigEther Controller el cual es un controlador de Entrada/Salida en dispositivos de red. Finalmente se describe al Chasis como tal el cual tiene la siguiente descripción Cisco 7206VXR.

Figura 41. Dashboard del Inventario de la Red



Elaborado por: Autor

### 3.4.4. Dashboards de KPIs de route source.

Para crear un dashboard en Tableau sobre los KPIs del router source es necesario que extraer en el Excel por medio de Python los KPIs los cuales son overhead, memory , route source, y el número de subnets los cuales se observan en la Figura 42.

Figura 42. Excel de los KPIs de Route Source

	A	B	C	D	E	F
	Router IP	Route Source	Networks	Subnets	Overhead	Memory (bytes)
1						
2	1.1.1.2	connected	0	9	540	1620
3	1.1.1.2	static	0	0	0	0
4	1.1.1.2	ospf	0	18	1200	3312
5	1.1.1.3	connected	0	9	540	1620
6	1.1.1.3	static	0	0	0	0
7	1.1.1.3	ospf	0	18	1260	3312
8	1.1.1.4	connected	0	9	540	1620
9	1.1.1.4	static	0	0	0	0
10	1.1.1.4	ospf	0	18	1200	3312
11	1.1.1.5	connected	0	7	420	1260
12	1.1.1.5	static	0	0	0	0
13	1.1.1.5	ospf	0	19	1920	3496
14	1.1.1.6	connected	0	9	540	1620
15	1.1.1.6	static	0	0	0	0
16	1.1.1.6	ospf	0	18	1140	3312
17	1.1.1.7	connected	0	7	420	1260
18	1.1.1.7	static	0	0	0	0
19	1.1.1.7	ospf	0	19	1200	3496
20	1.1.1.8	connected	0	7	420	1260
21	1.1.1.8	static	0	0	0	0
22	1.1.1.8	ospf	0	19	1200	3496

Elaborado por: Autor.

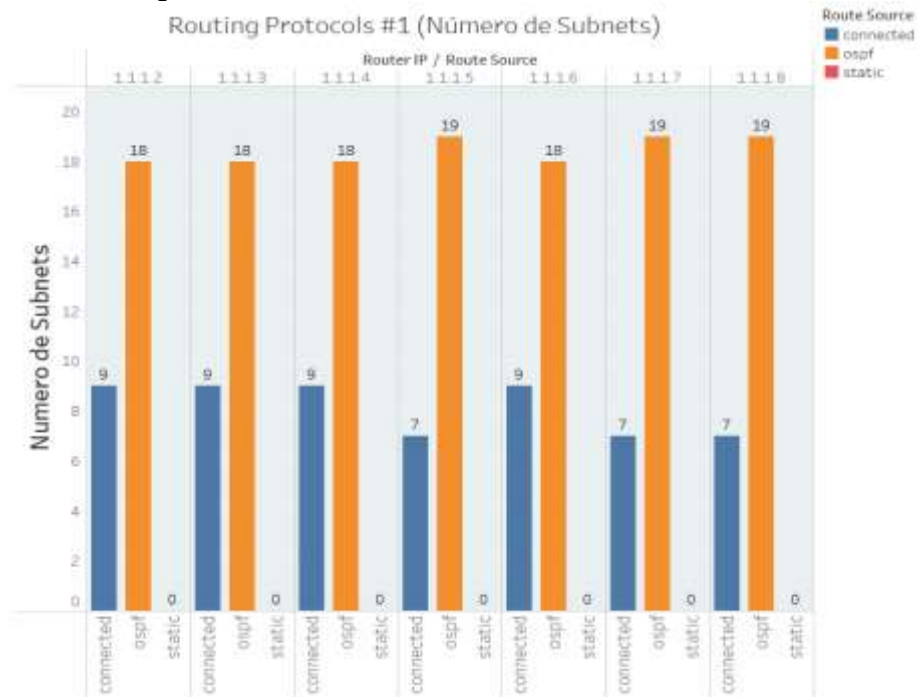
Luego de generar un Excel con los KPIs solicitados se procede a abrirlo en el entorno de Tableau con los siguientes pasos:

- Al abrir Tableau se presiona en la pestaña archivo en la parte superior del entorno, consecutivamente en la opción nuevo.
- Se selecciona el documento de Excel donde se obtuvo los KPIs.
- Luego dar formato a las variables como número entero y dar clic derecho en subnets y elegir la opción crear grupo calculado, al igual que los demás datos de memory, overhead. Luego cargar los datos, dar clic opción Hoja 1 , para mostrar las variables.
- Se considera realizar 3 dashboards referentes a cada medida.
- Para el primero se crea para subnets, se utiliza en las filas la variable Suma(Número de Subnets) mientras que en la columna se ubican las variables Router IP y Router Interface.
- Para el segundo dashboard se considera en las filas la variable Suma(Numero de Overhead) mientras que en las columnas se ubica las variables Router IP y Router Interface.

- Para el tercer dashboard se habla de la memoria para ello se usa en las columnas las variables Router IP y Router Interface, mientras en la fila Suma(Número de Memoria).
- Dar clic en Mostrarme para escoger el tipo de gráfico, se procede a elegir la forma en el cuadro Marcas, dar clic derecho las parte del gráfico para editar fuentes, textos, sombreado, etc.

Una vez obtenido el gráfico se realiza su respectivo análisis de cada variable, correspondiente al gráfico ejecutado. En la Figura 43 se logra visualizar el número de subredes que conoce cada router, es decir el número de redes que conocen sus routers vecinos, también se analiza el tipo de conexión del router, se tiene conectadas, estáticas y ospf. En la gráfica se analiza que todos los routers menos el RTCNC01, RTRIO01 y el RTYGU01 tienen 4 subredes, mientras que los 3 routers mencionados tienen solo 3 subredes cada uno conectadas.

Figura 43. Dashboard del Número de Routers vecinos



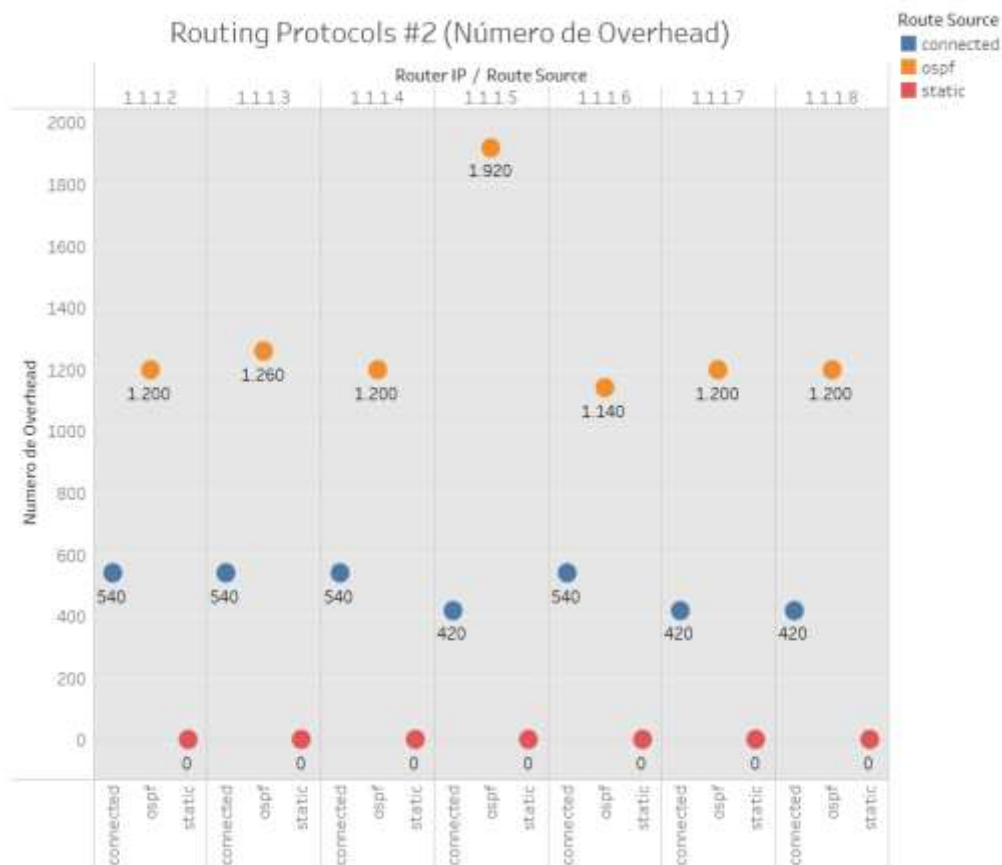
Elaborado por: Autor.

Al analizar sobre las subredes que se aprendieron por OSPF se cuenta todas las que pertenecen al área 1, menos las que están conectadas al router, por ello se conoce que los 3 routers mencionados tendrán una subred aprendida más que los demás, porque ellos no tienen una subred conectada,

se observa que los 3 routers tienen 19 subredes ospf mientras que los demás solo 18 subredes.

En el segundo análisis de la Figura 44 se logra visualizar el overhead de cada router. El overhead son los recursos utilizados para gestionar y controlar la red. Se logra visualizar un incremento significativo en el router RTYGU01 por lo cual se puede determinar que se ejecutó mayor uso de recursos por lo que se puede concluir que se debe a la lejanía con los otros elementos de la red. También se observa que los routers RTYGU01, RTRIO01, RTCNC01 tiene menos recursos en las fuentes que son conectadas debido a que tienen menos conexiones a diferencias de los otros routers mencionados.

Figura 44. Dashboard del Número de Overhead

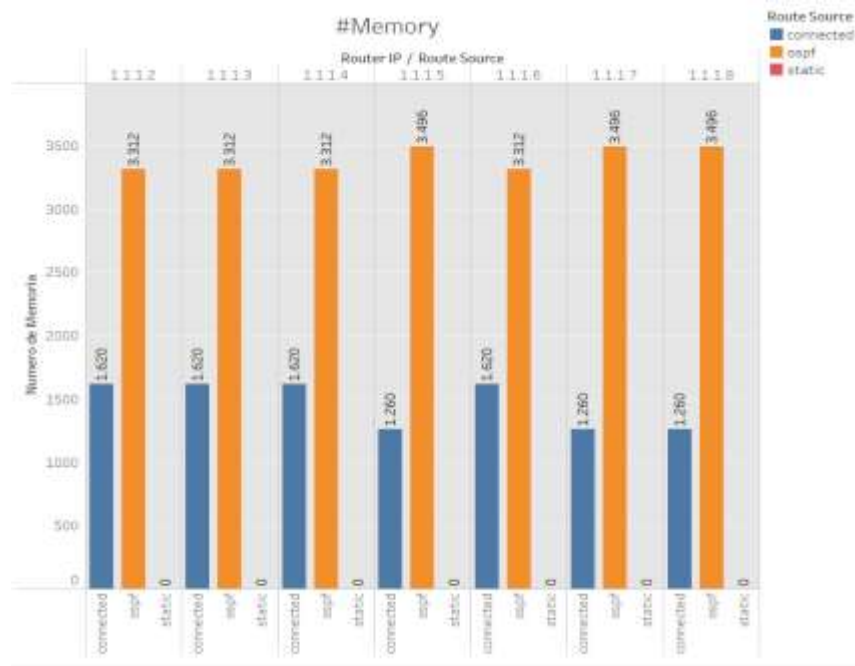


Elaborado por: Autor.

En el tercer análisis se determina la cantidad de bytes que utiliza cada fuente de enrutamiento, para establecerse en la tabla de enrutamiento. En la Figura 45 se observa que en connected hay una disminución en memoria en RTRIO01, RTYGU01, RTCNC01 respecto a los demás porque estos routers tienen una conexión menos, por ende, tienen una memoria menor, al contrario

del caso en el que los 3 routers mencionados incrementan su memoria porque poseen más conexiones por OSPF.

Figura 45. Dashboard del Número de Memoria



Elaborado por: Autor.

### 3.4.5. Dashboards de KPIs de OSPF neighbor.

Después de extraer los KPIs referentes a los router vecinos conocidos por OSPF a través de Python se busca conocer 3 elementos, los cuales son el número de routers vecinos de cada router, indicar si el router vecino es DR o BDR, y por último conocer por que red e interface descubrió el router que tiene routers vecinos. En la Figura 46 se muestra los KPIs que se recolectaron por medio de Python en un archivo de Excel el cual posee variables necesarias para el entorno de Tableau.

Figura 46. Excel de los KPIs de Router Neighbor.

	A	B	C	D	E	F	G
	Router IP	Neighbor ID	Pri	State	Dead Time	Address	Interface
1	1.1.1.2	1.1.1.3	1	FULL/DR	00:00:36	10.10.10.30	GigabitEthernet4/0
2	1.1.1.2	1.1.1.4	1	FULL/DR	00:00:33	10.10.10.21	GigabitEthernet3/0
3	1.1.1.2	1.1.1.5	1	FULL/DR	00:00:37	10.10.10.17	GigabitEthernet1/0
4	1.1.1.3	1.1.1.6	1	FULL/DR	00:00:38	10.10.10.34	GigabitEthernet0/0
5	1.1.1.3	1.1.1.2	1	FULL/BDR	00:00:39	10.10.10.29	GigabitEthernet4/0
6	1.1.1.3	1.1.1.4	1	FULL/DR	00:00:39	10.10.10.25	GigabitEthernet6/0
7	1.1.1.4	1.1.1.3	1	FULL/BDR	00:00:36	10.10.10.26	GigabitEthernet6/0
8	1.1.1.4	1.1.1.2	1	FULL/BDR	00:00:33	10.10.10.22	GigabitEthernet3/0
9	1.1.1.4	1.1.1.5	1	FULL/DR	00:00:35	10.10.10.13	GigabitEthernet0/0
10	1.1.1.5	1.1.1.2	1	FULL/BDR	00:00:39	10.10.10.18	GigabitEthernet2/0
11	1.1.1.5	1.1.1.4	1	FULL/BDR	00:00:39	10.10.10.14	GigabitEthernet0/0
12	1.1.1.6	1.1.1.7	1	FULL/DR	00:00:35	20.20.20.6	GigabitEthernet1/0
13	1.1.1.6	1.1.1.8	1	FULL/DR	00:00:39	20.20.20.2	GigabitEthernet2/0
14	1.1.1.6	1.1.1.3	1	FULL/BDR	00:00:37	10.10.10.33	GigabitEthernet0/0
15	1.1.1.7	1.1.1.8	1	FULL/DR	00:00:34	20.20.20.10	GigabitEthernet3/0
16	1.1.1.7	1.1.1.6	1	FULL/BDR	00:00:32	20.20.20.5	GigabitEthernet1/0
17	1.1.1.8	1.1.1.7	1	FULL/BDR	00:00:31	20.20.20.9	GigabitEthernet3/0
18	1.1.1.8	1.1.1.6	1	FULL/BDR	00:00:38	20.20.20.1	GigabitEthernet2/0
19	1.1.1.8	1.1.1.5	1	FULL/DR	00:00:38	20.20.20.1	GigabitEthernet2/0

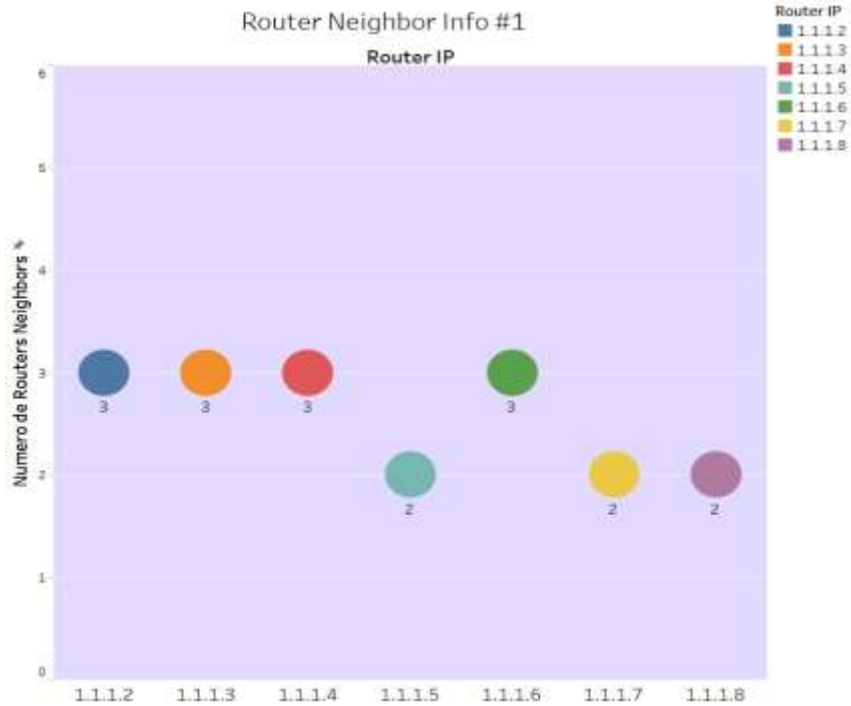
Elaborado por: Autor.

Luego de obtener el archivo de Excel con todos los KPIs recolectados junto con las variables necesarias para la creación de dashboards en Tableau se ejecuta los siguiente pasos:

- Abrir el programa Tableau para dar clic en la pestaña archivo el cual permitirá seleccionar el documento de Excel a cargar.
- Después de cargar los datos se procede a dar clic derecho en State y se selecciona la opción crear un grupo calculado. Consecutivamente se da clic en Hoja 1 para crear un entorno de trabajo donde utilizar las variables que se extrajo.
- Se considera crear 3 dashboards los cuales muestran los KPIs.
- El primero es el número de router neighbor, se utiliza en las filas la variable Suma(Número de neighbors) mientras que en la columna se ubican las variables Router IP y Router Neighbor.
- Para el segundo dashboard se considera en las filas la variable Router IP mientras que en las columnas se muestran las variables Número de BDR/DR y Neighbor ID.
- Para el tercer dashboard se habla de la memoria para ello se usa en las columnas las variables Router IP y Router Interface, mientras en la fila Suma(Numero de Memoria).
- Una vez conseguido los tres dashboard se procede a realizar la edición del tipo de gráfico a utilizar se debe dar clic en la parte de Mostrarme para elegir el gráfico a utilizar luego dar clic en la parte de Marcas para elegir la forma del gráfico. Al elegir el formato dar clic derecho en la parte del entorno de trabajo del dashboard que se requiere editar colores, fuentes, tamaño etc.

Una vez obtenido los dashboards en Tableau se realiza el análisis de cada gráfico creado. Primero se analiza el del número de routers neighbors, en la Figura 47 se observa como todos los routers menos los routers RTCNC01, RTRIO01 y RTYGU01 tienen 3 routers vecinos ya que su ubicación en la topología es más central que los demás y poseen más conexiones, mientras que los que se ubican por los límites de la topología tienen solo 2 routers vecinos al poseer una conexión menos a un router.

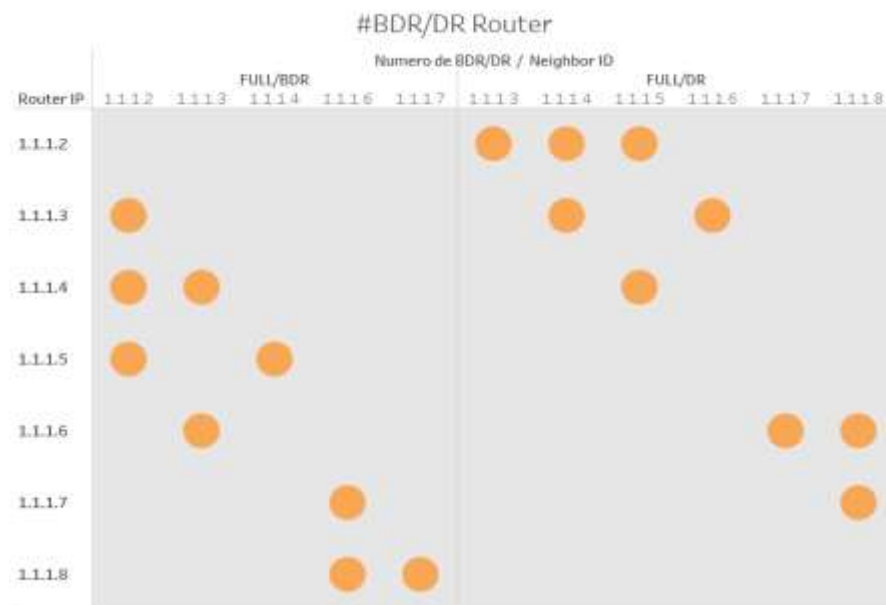
Figura 47. Dashboard de Numero de Router Vecinos



Elaborado por: Autor.

Para el segundo dashboard, se analiza en relación con el caso planteado anteriormente el cual menciona que todos los routers están configurados con prioridad 1 por ende la selección de DR o BDR se deberá a su dirección IP. En la Figura 48 se observa que el router con una IP menor como es el caso del RTDUR01 posee más routers DR mientras que el router de mayor IP el cual es el RTRIO01 posee más router BDR.

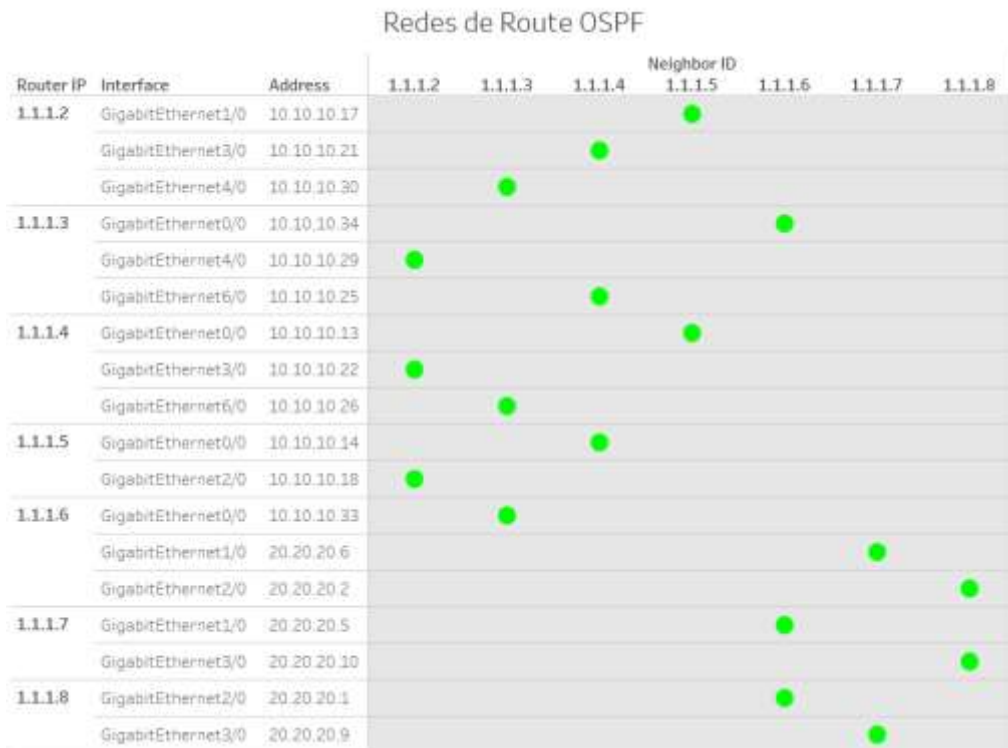
Figura 48. Dashboard de Router DR/BDR.



Elaborado por: Autor.

En el tercer dashboard se analiza la dirección de la red que conecta 2 routers por medio de una interface para identificarse como routers neighbors. En la Figura 49 se muestra porque interface aprendió la red por medio del protocolo OSPF, así realizando una correcta conexión al realizar intercambio de mensajes, conectando así toda la topología realizada obteniendo una red enmallada conectada por medio del protocolo OSPF el cual decidirá y aprenderá de manera dinámica el comportamiento de la red.

Figura 49. Dashboard de Redes aprendidas por OSPF.



Elaborado por: Autor.

Cada router en la red posee un Neighbor ID el cual es el identificador del router, en este caso es la IP loopback que facilita el reconocimiento del equipo en la topología, la cual poseen solo un área el cual responde a los routers ubicados de manera local, para el correcto enrutamiento de paquetes y el análisis del comportamiento de la red.

Este gráfico es indispensable para conocer las IPs que se encuentran en la red y donde se las puede ubicar por medio de sus interfaces, que es de gran utilidad al realizar troubleshooting, el cual es el método para reconocer e identificar donde se origina un problema en la red, para así el gestor o administrador de la red obtenga un análisis más detallado.



## Conclusiones.

- La investigación revela que comprender los fundamentos del networking es crucial para la configuración eficiente de los routers en una red enmallada tal como se menciona en el objetivo específico 1. Además, la identificación adecuada de los indicadores KPIs y el dominio de herramientas de simulación como GNS3 son esenciales para garantizar la precisión y la eficacia de las simulaciones.
- Se evidencia que el diseño cuidadoso de los escenarios de simulación en GNS3 propuestos en el objetivo específico 2 permiten la captura efectiva de métricas KPIs relevantes para evaluar el rendimiento de la red enmallada. La asignación de prioridades y la configuración de protocolos como OSPF son aspectos críticos en este proceso.
- La utilización de scripts de Python para la recopilación de datos en tiempo real durante la simulación demuestra ser una estrategia eficaz. Estos scripts propuestos en el objetivo específico 3 permiten la extracción automatizada de información vital de los routers, incluyendo datos de interfaces, rendimiento de CPU/memoria, etc.
- La transformación de los datos recopilados en formatos accesibles, como documentos de Excel, facilita su análisis y visualización posterior en Tableau. Esta integración de datos brinda una comprensión holística del rendimiento de la red y permite la identificación de áreas de mejora siendo este el resultado del objetivo específico 4 del proyecto.
- El análisis de los resultados revela una comprensión profunda del estado tanto físico como lógico de la red enmallada. Se observa que factores como la distribución de la carga, la disponibilidad de rutas alternativas y el uso de recursos impactan significativamente en el rendimiento global de la red, concluyendo que el objetivo específico 5 se realizó correctamente.

## Recomendaciones.

En este trabajo de titulación se realizan las siguientes recomendaciones que son de gran relevancia tanto para el diseño de la topología, configuración de OSPF, elaboración de los scripts de Python, creación de los dashboards de los KPIs en Tableau.

- Definir las redes y la cantidad de host necesarios para realizar el respectivo subnetting.
- Utilizar la IP loopback de cada router como Router ID para mayor facilidad al momento de reconocer equipos y datos.
- Tener en cuenta la capacidad del computador donde se simulará porque estos programas utilizan demasiada memoria, se recomienda instalarlos en el Disco D.
- Tener en cuenta que las condicionantes que sirven para extraer los KPIs varían de acuerdo con el router asignado, o el tipo de interface que se utilice.
- Utilizar Oracle Workstation para gestionar la GNS3 VM ya que este se encuentra de manera gratuita y es de fácil uso.
- Para las pruebas de conexión y envío de paquetes utilizar la herramienta Ipterm, usando paquetes UDP configurando una cierta cantidad de tiempo para ejecutarse.
- Para la elaboración de una topología de alta capacidad utilizar routers robustos desde la versión 7200 el cual posee muchas interfaces GigabitEthernet.

## Bibliografía

- Aleman, P., Roman, A., Vilalta, R., Pol, A., Bonnet, J., Kapassa, E., Touloupou, M., Kyriazis, D., Panagiotis, K., Panagiotis, T., Martrat, J., Casellas, R., Martinez, R., & Muñoz, R. (2021). A KPI-Enabled NFV MANO Architecture for Network Slicing with QoS. *IEEE Communications Magazine*, 59, 44–50.  
<https://doi.org/10.1109/MCOM.001.2001077>
- Araf, M. (2023). *Performance Analysis of OSPF and EIGRP Routing Protocols*. <https://doi.org/10.13140/RG.2.2.25197.67045>
- Arteaga, J. (2023). Análisis comparativo de IDEs enfocados a Machine Learning. *Esprint Investigación*, 2, 5–13.  
<https://doi.org/10.61347/ei.v2i2.53>
- Basira, Ts. H., Razaleigh, B. S., & Sanjay, C. A. (2015). *IPv4 Subnetting and vlsn calculation edition 1.0 © 2015 (1.0)*.
- Burns, H. (2021). Crowdsourced-Data Normalization with Python and Pandas. *The Programming Historian*, 10.  
<https://doi.org/10.46430/phen0093>
- Bustamante, R., & Hincapié, R. (2005). Análisis, modelamiento y simulación de redes enmalladas basadas en el estándar 802.16-2004. *Sistemas y Telemática*, 3(6), Article 6. <https://doi.org/10.18046/syt.v3i6.1081>
- Cañizares, L., & Alejandro, E. (2020). *Diseño e Implementación de Soluciones Integradas en LAN WAN Fundamentos de Networking y Principios de Enrutamiento*.  
<http://repository.unad.edu.co/handle/10596/36213>
- Castillo-Velazquez, J.-I., & Huerta, M. (2021). *Performance Comparison for File Transfers for Four Advanced Networks in GNS3 under IPv6*.
- Choi, B. (2021). *Python Network Automation Labs: SSH paramiko and netmiko* (pp. 583–628). [https://doi.org/10.1007/978-1-4842-6806-3\\_14](https://doi.org/10.1007/978-1-4842-6806-3_14)

- Clausen, T. H., Jacquet, P., & Baccelli, E. (2004). *OSPF-style Database Exchange and Reliable Synchronization in the OLSR*.
- Co, Ltd. joe. huangxu, & Huawei Technologies Co., Ltd. (2022). IP Addresses and Subnetting. En *Data Communications and network technologies* (pp. 113–153). [https://doi.org/10.1007/978-981-19-3029-4\\_4](https://doi.org/10.1007/978-981-19-3029-4_4)
- Darín, J. R. (2016). *Fundamentos de Redes Informáticas: 2ª Edición*. IT Campus Academy. <https://books.google.es/books?hl=es&lr=&id=gGTkDAAAQBAJ&oi=fnd&pg=>
- Dieudonne, M., & Wang, H. (2023). *Whitepaper Beyond 5G/6G KPI Measurement*. <https://doi.org/10.5281/zenodo.7963247>
- Ding, S., Wei, X., Xu, L., & Wang, H. (2023). *SMTWM: Secure Multiple Types Wildcard Pattern Matching Protocol from Oblivious Transfer* (pp. 471–489). [https://doi.org/10.1007/978-3-031-22677-9\\_25](https://doi.org/10.1007/978-3-031-22677-9_25)
- Elezi, A., & Karras, D. (2022). On Detailed Network Systems Configuration Management Automation using Python. *Wseas transactions on communications*, 22, 1–16. <https://doi.org/10.37394/23204.2023.22.1>
- Escobar León, A. A., & Pineda Suárez, M. N. (2022). *Simulación de redes WAN mediante GNS3 utilizando múltiples fabricantes para el CP - Redes de computadoras*. [bachelorThesis, Quito, 2022]. <http://bibdigital.epn.edu.ec/handle/15000/22096>
- García, S. J., & María, I. (2019). *Diseño e implementación de un módulo Python para representar datos geográficos en Dataframes* [Bachelor thesis]. <https://e-archivo.uc3m.es/handle/10016/31974>
- GNS3. (2023). *GNS3 | The software that empowers network professionals*. <https://www.gns3.com/>
- Golightly, L., Modesti, P., & Chang, V. (2023). Deploying Secure Distributed Systems: Comparative Analysis of GNS3 and SEED Internet

Emulator. *Journal of Cybersecurity and Privacy*, 3, 464–492.  
<https://doi.org/10.3390/jcp3030024>

González Lattuada, A. F., & Maggiolo Rocca, S. C. (2017). *Analizador de calidad en redes IP*.  
<https://dspace.ort.edu.uy/handle/20.500.11968/3390>

Gonzalez-Ramos, A. D., Ibañez-Bautista, J. P., Zamacona-Prado, N., & Peza-Ortiz, E. (2019). Método de cálculo de subredes de las direcciones IP v4 para la asignación y administración dentro de una red corporativa. *Revista de Tecnologías Computacionales*, 21–26.  
<https://doi.org/10.35429/JOCT.2019.12.3.21.26>

Grgurevic, I., Barišić, G., & Stančić, A. (2021). *Analysis of MPLS and SD-WAN Network Performances Using GNS3* (pp. 77–90).  
[https://doi.org/10.1007/978-3-030-78459-1\\_6](https://doi.org/10.1007/978-3-030-78459-1_6)

Guamán, I. P. O. (2017). *Medición del consumo de energía en un nodo sensor inalámbrico en la transmisión de video sobre ipv6 en aplicaciones de domótica*. 113.

Gueant, V. (2023). *iPerf—The TCP, UDP and SCTP network bandwidth measurement tool*. <https://iperf.fr/>

Gupta, S. (2022). *Installing PyCharm IDE*. [https://doi.org/10.1007/978-1-4842-8202-1\\_2](https://doi.org/10.1007/978-1-4842-8202-1_2)

Hernández, E. A., Bautista, J. C., Zenil, A. E. G., Medellín, A. A. H., Hernández, S. H., & Hernández, G. H. (2017). Comparación de los modelos OSI y TCP/IP. *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla*, 5(10), Article 10.  
<https://doi.org/10.29057/esh.v5i10.2461>

Herrera, T., & Nunez, F. (2020). Design and Prototyping of a Thread Border Router Based on a Non Network-Co-Processor Architecture. *IEEE Access, PP*, 1–1. <https://doi.org/10.1109/ACCESS.2020.2983734>

- Jetbrains. (2023). *PyCharm: El IDE de Python para desarrolladores profesionales, por JetBrains*. JetBrains. <https://www.jetbrains.com/es-es/pycharm/>
- Jimenez, E., Caycho, M., Pantigozo, A., & Bailón, F. (2023). Efecto del case virtual de telecomunicaciones WAN sobre la competencia gestiona los recursos de enrutadores de telecomunicaciones. *Revista Científica: Biotech and engineering*, 3. <https://doi.org/10.52248/eb.Vol3Iss1.90>
- Kamble, N., & Kotwal, R. (2023). OSI Model. *International Journal of Advanced Research in Science, Communication and Technology*, 219–225. <https://doi.org/10.48175/IJAR SCT-13032>
- Kara, F., Kaya, H., Ng, B., & Lam, C. T. (2023). *Bit-Interleaved Multiple Access: Improved Fairness, Reliability, and Latency for Massive IoT Networks*. *PP*, 1–1. <https://doi.org/10.1109/JIOT.2023.3267079>
- Kingsley, M. (2023). *Numbering Systems and Subnetting* (pp. 59–89). [https://doi.org/10.1007/978-3-031-33669-0\\_3](https://doi.org/10.1007/978-3-031-33669-0_3)
- Kumar, R., & Shinde, P. R. (2016). *Computer Network—IP Address & Subnetting*. 5(4). [https://d1wqtxts1xzle7.cloudfront.net/63046837/D454904541620200422-20895-11suu1l-libre.pdf?1587540058=&response-content-disposition=inline+filename-Computer\\_Network\\_IP\\_Address\\_and\\_Subnetting.pdf&Expires=1699942826&Signature=W2Ca6~tu6mfclow9gQH-](https://d1wqtxts1xzle7.cloudfront.net/63046837/D454904541620200422-20895-11suu1l-libre.pdf?1587540058=&response-content-disposition=inline+filename-Computer_Network_IP_Address_and_Subnetting.pdf&Expires=1699942826&Signature=W2Ca6~tu6mfclow9gQH-)
- Lau, R. (2021). *Secure Shell (SSH)* (pp. 159–174). [https://doi.org/10.1007/978-1-4842-6960-2\\_7](https://doi.org/10.1007/978-1-4842-6960-2_7)
- Li, X., Xu, J., Nie, J., Guo, C., & Zhao, X. (2012). *Simulation and Analysis of OSPFv2 DR Election Based on GNS*. <https://doi.org/10.2991/citcs.2012.131>
- Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information*, 12(2), Article 2. <https://doi.org/10.3390/info12020087>

- López-Carmona, D., Vargas-Canales, J., Orozco-Cirilo, S., Jimenez-Jimenez, E., Rodríguez-Haros, B., & Agronegocios, L. (2023). Análisis del networking en el sector agroalimentario. *JÓVENES EN LA CIENCIA*, 21, 1–7.
- Lopez-Carreño, J., Calvo-Lavado, C., & Zarate Perez, E. (2023). *University Student Control Detection System Based on Machine Learning and Artificial Intelligence*.
- Manca, V., & Bonnici, V. (2023). *Introduction to Python* (pp. 189–221). [https://doi.org/10.1007/978-3-031-44501-9\\_7](https://doi.org/10.1007/978-3-031-44501-9_7)
- Martinez, D. M. A., & Kelsey, R. J. R. (2007). *Redes inalámbricas enmalladas metropolitanas*. <https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/1479/0040419.pdf?sequence=1&isAllowed=y>{
- Mishra, A., & Sahoo, A. (2020). *S-OSPF: A Traffic Engineering Solution for OSPF based Best Effort Networks*.
- Nedyalkov, I. (2023). Application of GNS3 to Study the Security of Data Exchange between Power Electronic Devices and Control Center. *Computers*, 12(5), Article 5. <https://doi.org/10.3390/computers12050101>
- Nedyalkov, I., & Georgiev, G. (2023). *Using the GNS3 Platform for Characterizing the Traffic in a VoIP Network and Study Its Performance* (pp. 543–553). [https://doi.org/10.1007/978-981-99-3236-8\\_43](https://doi.org/10.1007/978-981-99-3236-8_43)
- Neteris. (2023). *Tableau Software Visualización de Datos*. <https://info.neteris.com/tableau-software-visualizacion-datos/>
- Neumann, J. C. (2015). *The book of GNS3: Build virtual network labs using Cisco, Juniper, and more*. San Francisco : No Starch Press. [http://archive.org/details/thebookofgns3\\_201908](http://archive.org/details/thebookofgns3_201908)

- Nuñez Maturel, L. (2013). Interconexión de las redes mediante enrutadores. *Avanzada Científica*, 16(2), 1–18.
- OpenPyXL. (2023). *Openpyxl—A Python library to read/write Excel 2010 xlsx/xlsm files—Openpyxl 3.1.2 documentation*.  
<https://openpyxl.readthedocs.io/en/stable/>
- Openpyxl / openpyxl · GitLab*. (2023, octubre 25). GitLab.  
<https://foss.heptapod.net/openpyxl/openpyxl>
- Oviedo, B., Litardo, L. S., Mera, E. Z., & Hernández, A. P. R. (2018). Visualizador de tráfico de red de comunicación basadas en la arquitectura TCP/IP. *Revista Científica Ciencia y Tecnología*, 18(20), Article 20. <https://doi.org/10.47189/rcct.v18i20.227>
- Panek, C. (2020). Defining Networks with the OSI Model. En *Networking Fundamentals* (pp. 43–73). Wiley.  
<https://doi.org/10.1002/9781119650768.ch2>
- Papan, J., Segec, P., Yeremenko, O., Bridova, I., & Hodon, M. (2020). Enhanced Multicast Repair Fast Reroute Mechanism for Smart Sensors IoT and Network Infrastructure. *Sensors*, 20(12), Article 12.  
<https://doi.org/10.3390/s20123428>
- Pearson IT Certification. (2012). *OSPF Basic Configuration | OSPF Basic Configuration | Pearson IT Certification*. <https://www.pearsonitcertification.com/articles/article.aspx?p=1868078>
- Putri, D., & Mogi, I. K. (2020). Designing Computer Network Subnetting with School Objects Using Cisco Packet Tracer 6.1. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 8, 445. <https://doi.org/10.24843/JLK.2020.v08.i04.p10>
- Rico-Bautista, D., Sánchez-Espinosa, L., & Portillo-Ballesteros, E. Y. (2014). Redes Mesh, una alternativa a problemas de cobertura de red: Una revisión de literatura. *Revista Ingenio*, 7(1), Article 1.  
<https://doi.org/10.22463/2011642X.2031>



- Ryzhkov, F., Ryzhkova, Y., & Elinson, M. (2023). Python in Chemistry: Physicochemical Tools. *Processes*, 11, 2897. <https://doi.org/10.3390/pr11102897>
- Sachinidis, T., Politis, A., & Hilas, C. (2023). *To Split or not to Split? A Simulation Study on the Network Convergence Duration of Multi-Area OSPF*. <https://doi.org/10.1109/TSP59544.2023>
- Sanchez, E., Figueroa, D., & Gamarra, A. (2020). *Implementación de un Servidor DNS Seguro basado en Pi-Hole utilizando un entorno virtualizado*.
- Shaikh, A., & Greenberg, A. (2004). *OSPF Monitoring: Architecture, Design, and Deployment Experience*. (p. 70).
- Shamim, F. (Ed.). (2002). *Troubleshooting IP routing protocols*. Cisco Press.
- Shang, Z., Wei, Q., & Li, Z. (2023). Machine Vision Algorithm for Identifying Packaging Components of HN-3 Arterial Blood Sample Collector. *Applied Sciences*, 13(14), Article 14. <https://doi.org/10.3390/app13148450>
- Siahaan, A. P. U. (2017). *A Review of IP and MAC Address Filtering in Wireless Network Security* [Preprint]. INA-Rxiv. <https://doi.org/10.31227/osf.io/g6emr>
- Sisat, S., Bhopale, P., & theVishwajit, B. (2013). IP Subnetting. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 2, 5–10.
- Song, L., Di, B., Zhang, H., & Han, Z. (2023). *Basics of Aerial Access Networks* (pp. 7–8). <https://doi.org/10.1017/9781108936538.003>
- T, C., G, K., Sharma, S., & S, S. (2023). A Review on Fast Convergence Scheme in OSPF Network. *International Journal of Advanced Research in Science, Communication and Technology*, 672–676. <https://doi.org/10.48175/IJARSCT-8024>

- Tamrin, T., Muhaidi, N., Arifin, A., & Ariyanto. (2023). Implementasi metode vlsm (variable length subnet mask) pada pemetaan ip address lan (local area network) di lab fakultas saint dan teknologi (FST) UNISNU JEPARA. *Jurnal Publikasi Teknik Informatika*, 2, 6–11. <https://doi.org/10.55606/jupti.v1i1.963>
- Teca, G., & Natkaniec, M. (2023). A Novel Covert Channel for IEEE 802.11 Networks Utilizing MAC Address Randomization. *Applied Sciences*, 13(14), Article 14. <https://doi.org/10.3390/app13148000>
- Tiwari, A., & Sahoo, A. (2007). Providing QoS in OSPF based best effort network using load sensitive routing. *Simulation Modelling Practice and Theory*, 15, 426–448. <https://doi.org/10.1016/j.simpat.2006.11.008>
- Tyagi, A. (2020). TCP/IP Protocol Suite. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 59–71. <https://doi.org/10.32628/CSEIT206420>
- Ylonen, T., Turner, P., Scarfone, K., & Souppaya, M. (2015). *Security of Interactive and Automated Access Management Using Secure Shell (SSH)* (NIST IR 7966; p. NIST IR 7966). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.IR.7966>
- Zadka, M. (2019). *Paramiko* (pp. 111–119). [https://doi.org/10.1007/978-1-4842-4433-3\\_9](https://doi.org/10.1007/978-1-4842-4433-3_9)



Presidencia  
de la República  
del Ecuador



Plan Nacional  
de Ciencia, Tecnología,  
Innovación y Saberes



SENESCYT  
Secretaría Nacional de Educación Superior,  
Ciencia, Tecnología e Innovación

## DECLARACIÓN Y AUTORIZACIÓN

Yo, **Alvarez Granda, David Andrés** con C.C: # 094152112-2 autor del Trabajo de Titulación: **Elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3 por medio de herramientas de Excel, Python y Tableau**, previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 16 de febrero del 2024

f. \_\_\_\_\_

Nombre: Alvarez Granda, David Andrés

C.C: 094152112-2



Presidencia  
de la República  
del Ecuador



Plan Nacional  
de Ciencia, Tecnología,  
Innovación y Saberes



SENESCYT  
Secretaría Nacional de Educación Superior,  
Ciencia, Tecnología e Innovación

## **REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA**

### **FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN**

<b>TÍTULO Y SUBTÍTULO:</b>	Elaboración, recopilación, análisis y presentación de los KPIs de una red enmallada simulada en GNS3 por medio de herramientas de Excel, Python y Tableau.		
<b>AUTOR(ES)</b>	David Andrés Alvarez Granda		
<b>REVISOR(ES)/TUTOR(ES)</b>	M. Sc. Palacios Meléndez, Edwin Fernando		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Educación Técnica para el Desarrollo		
<b>CARRERA:</b>	Ingeniería en Telecomunicaciones		
<b>TÍTULO OBTENIDO:</b>	Ingeniero en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	16 de Febrero del 2024	<b>No. DE PÁGINAS:</b>	99
<b>ÁREAS TEMÁTICAS:</b>	Networking, Programación		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	GNS3, OSPF, KPIs, Python, Tableau, Scripts, Dashboards, Enrutamiento.		
<b>RESUMEN:</b>	<p>El presente trabajo de titulación propone la elaboración de una red enmallada en GNS3, la cual está configurada con el protocolo de enrutamiento OSPF (protocolo de enrutamiento dinámico). Esta red se compone de 7 routers los cuales poseen el router ID de acuerdo a la IP de su interface Loopback, luego se extraen los KPIs del chasis, route source, ospf neighbors, interfaces, cpu y memoria entre otros, los cuales deben variar de acuerdo con los tres escenarios propuestos y por medio de scripts de Python se genera un Excel, con el archivo de Excel se cargan los datos de las variables a Tableau donde se cambia formatos, calcula y se gestiona de acuerdo a su ubicación en los ejes correspondientes para que se elaboren los dashboards referentes a los KPIs mostrando gráficos de forma interactiva para el usuario consecutivamente se realiza el respectivo análisis de la red para obtener los resultados de los escenarios propuestos, generando nuevas conclusiones.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR:</b>	<b>Teléfono:</b> +593991105161	<b>E-mail:</b> david.alvarez03@cu.ucsg.edu.ec	
<b>CONTACTO CON LA INSTITUCIÓN:COORDINADOR DEL PROCESO DE UTE</b>	<b>Nombre:</b> Ubilla Gonzalez, Ricardo Xavier		
	<b>Teléfono:</b> +593 99 952 8515		
	<b>E-mail:</b> ricardo.ubilla@cu.ucsg.edu.ec		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			