

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE *ANSIBLE* PARA LA CONFIGURACIÓN DE
EQUIPOS DE *NETWORKING***

**IMPLEMENTAR *ANSIBLE* PARA CONFIGURAR EQUIPOS DE
NETWORKING DE FORMA DINÁMICA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

FRANK JOHN VILLAGRÁN ANGULO

DIRECTOR: ING. GABRIELA KATHERINE CEVALLOS SALAZAR, MSC.

DMQ, septiembre 2022

CERTIFICACIONES

Yo, Frank John Villagrán Angulo declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Frank John Villagrán Angulo

frank.villagran@epn.edu.ec

frankjvilla95@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Frank John Villagrán Angulo, bajo mi supervisión.



ING. Gabriela Katherine Cevallos Salazar, MSC.

DIRECTOR

gabriela.cevalloss@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Frank John Villagrán Angulo

DEDICATORIA

Dedico con toda mi alma, vida y corazón, esta tesis a mi familia, mis padres Johnny y Jenny, por brindarme todo su ser para que esta meta se cumpla y por nunca negar su apoyo durante este largo trayecto, su cariño, sus valores y toda su voluntad junto a cada paso que hacía, por las noches y madrugadas de desvelo que tuvieron junto a mí.

A mis hermanos Ronny y Ricky que estimularon mis días y mis noches con sus graciosas ocurrencias y a pesar de todo me brindaron sus consejos, su apoyo, su lealtad y sentido común para aprender y crecer juntos cada día.

A mis tíos, en especial a Eugenia y Diego por ese sentimiento de apoyo que día a día me extendían, así como también sus consejos y su tiempo.

A mis profesores que a lo largo de mi vida deportiva me dejaron grandes enseñanzas, gracias, Carlos, Daysi y Jorge, por esas grandes anécdotas, vivencias y tiempo juntos.

A mis profesores de la ESFOT que han hecho de mi vida estudiantil y laboral una grata experiencia, gracias por sus enseñanzas.

¡Cada uno ha sido para mí una razón más para alcanzar esta meta, gracias!

Frank J. Villagrán A.

AGRADECIMIENTO

Agradecido con Dios por darme la voluntad y valentía para llegar a culminar esta meta con felicidad y salud en conjunto con quienes más aprecio.

A mi familia, por toda la tolerancia, comprensión, y dedicación para ayudarme durante todo este tiempo en cada peldaño para lograr cumplir un objetivo más.

A la ESFOT, por las grandes victorias que festejamos juntos y todo lo que logramos y conseguimos en conjunto. Esfotdianos!!!

Agradezco a la EPN por la oportunidad ofrecida para adquirir los conocimientos que hoy por hoy, me vuelven el profesional que soy.

En especial a mi tutora de trabajo y diseño de titulación Ing. Gabriela Cevallos Msc., por la dedicación, el respeto, la humildad y ante todo el profesionalismo que logró inspirar mi admiración por su persona.

A quienes hicieron de este momento en mi vida, disfrutarlo, sonreír, ameno, amigable y positivo, por ser una parte de mi forma de ser y estar en mis locuras mientras cumplía uno de mis sueños.

¡Siempre feliz y antes muerto que dejar de forjar sueños!

Frank J. Villagrán A.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORÍA	¡Error! Marcador no definido.
DEDICATORIA.....	¡Error! Marcador no definido.
AGRADECIMIENTO.....	¡Error! Marcador no definido.
ÍNDICE DE CONTENIDO.....	¡Error! Marcador no definido.
RESUMEN.....	¡Error! Marcador no definido.
ABSTRACT.....	¡Error! Marcador no definido.
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	1
1.3 Alcance	1
1.4 Marco Teórico	2
Particularidades de Ansible	2
Sistema Operativo Hospedador.....	2
Software de Simulación de Redes.....	3
2 METODOLOGÍA.....	4
3 RESULTADOS	5
3.1 Instalación de Ansible en el nodo controlador	5
Creación de la Máquina Virtual con Ubuntu Server	5
Instalación de la Herramienta Ansible	6
3.2 Implementación de la topología en el simulador de red	7
3.3 Implementación de las configuraciones mediante Ansible	12
Diseño del <i>playbook</i>	13
3.4 Pruebas de funcionamiento.....	25
Datos ingresados por el usuario	25
Respuesta de <i>playbook</i> a su ejecución	26
Consolas de los equipos de <i>networking</i>	28

	Ping desde dispositivos finales a diversos puntos de la red	34
4	CONCLUSIONES.....	40
5	RECOMENDACIONES	41
6	Referencias	43
7	ANEXOS.....	45
	ANEXO I: Certificado de Originalidad	i
	ANEXO II: Video de implementación y pruebas de verificación	ii
	ANEXO III: Código de los playbooks	iii

RESUMEN

Cada día la automatización de sistemas se desarrolla aún más, con la finalidad de satisfacer las necesidades de uso cotidiano para las personas, entre estos desarrollos se encuentra la automatización de redes y para ello se utiliza la herramienta Ansible. Ansible permite ejecutar tareas de configuración de equipos de *networking*, por medio de un nodo controlador, que pueda manejar el lenguaje Python como único requisito para el sistema y que cuente con Ansible instalado.

Al establecer comunicación SSH, esta herramienta ejecuta un archivo con extensión *.yaml* denominado *playbook*, el cual contiene las configuraciones a ser realizadas en los equipos con los que se tiene comunicación.

El primer capítulo es concerniente a la descripción del componente, se presentan los objetivos, el alcance y el marco teórico que permite el desarrollo de este proyecto de titulación.

El segundo capítulo corresponde a la metodología, donde se describen los procedimientos realizados para cada uno de los objetivos planteados, hasta el punto donde se implementa Ansible para la configuración de equipos de *networking*.

El tercer capítulo corresponde a los resultados, en el que se encuentra el desarrollo de cada objetivo planteado, la verificación de las configuraciones realizadas y las pruebas de verificación con los resultados conseguidos.

El capítulo cuarto expone las conclusiones, las cuales son obtenidas de cada uno de los objetivos que fueron planteados.

Finalmente, el quinto capítulo corresponde a las recomendaciones, que se obtuvieron en base a los desafíos y las soluciones que fueron alcanzadas durante el desarrollo de este proyecto de titulación.

PALABRAS CLAVE: Ansible, SSH, EIGRP, DHCP, banner y contraseñas.

ABSTRACT

Every day the automation of systems is developed even more, to meet the needs of daily use for people, among these developments is the automation of networks and for this the Ansible tool is used. Ansible allows you to execute configuration tasks of networking equipment, through a controller node, which can handle the Python language as the only requirement for the system and that has Ansible installed.

When establishing SSH communication, this tool executes a file with a .yml extension called playbook, which contains the configurations to be made on the computers with which you have communication.

The first chapter concerns to description of the component, the objectives, scope, and theoretical framework that allows the development of this degree project are presented.

The second chapter corresponds to the methodology, where the procedures carried out for each of the objectives are described, to the point where Ansible is implemented for the configuration of networking teams.

The third chapter corresponds to the results, which includes the development of each proposed objective, the verification of the configurations carried out and the verification tests with the results achieved.

The fourth chapter presents the conclusions, which are obtained from each of the objectives that were proposed.

Finally, the fifth chapter corresponds to the recommendations, which were obtained based on the challenges and solutions that were reached during the development of this degree project.

KEYWORDS: *Ansible, SSH, EIGRP, DHCP, banner, and passwords.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Por la gran cantidad de equipos que puede contener una red, la configuración de todos estos se vuelve trabajosa, consumiendo tiempo y recurso humano. Por lo que es necesaria la automatización de varios de los procesos básicos para que la red funcione correctamente y se minimicen los errores de configuración realizada por los administradores de la red.

El presente proyecto pretende desplegar una topología de red, automatizando la configuración de los equipos de *networking* mediante el uso de la herramienta de DevOps Ansible. Ansible permite automatizar tareas de administración de los dispositivos de red de manera remota; garantizando eficiencia, rapidez y fiabilidad; con menos esfuerzo y riesgo de errores humanos.

1.1 Objetivo general

Implementar Ansible para la configuración de equipos de *networking*.

1.2 Objetivos específicos

- Instalar la herramienta de Ansible en el nodo controlador
- Montar los diferentes dispositivos de la red en un simulador de redes
- Implementar las configuraciones mediante la herramienta Ansible
- Realizar pruebas de funcionamiento y verificación de los resultados obtenidos

1.3 Alcance

El presente proyecto conlleva la implementación de una topología de red, donde los dispositivos logren tener conectividad configurándolos de forma automatizada mediante la herramienta de Ansible. Para esto se debe establecer una comunicación SSH entre el nodo controlador, máquina principal donde se instala Ansible, y los diferentes dispositivos de red. Con esto se despliega de manera remota las configuraciones necesarias para la implementación de la topología de red establecida.

1.4 Marco Teórico

Particularidades de Ansible

La herramienta Ansible de *DevOps* que es *open source*, permite la automatización de los procesos de administración, despliegue, actualización y configuración de red. Se encuentra desarrollado en lenguaje Python que se encuentra preinstalado o puede correr sobre toda distribución de la familia Linux, además de contar con lenguaje YAML para el uso de sus archivos, esto hace que la herramienta sea extensible y fácil de usar [1] [2] [3] [4] [5].

Una de las características de funcionamiento remoto es que no necesita de la instalación de un agente, para operarlo únicamente se requiere conocimientos en programación y haber establecido comunicación SSH con el *host*. Es una herramienta que no posee un entorno gráfico, pero cuenta con las herramientas para validar y depurar. Para ser ejecutado se requiere de un nodo de control y mínimo un *host* [6] [7].

Sistema Operativo Hospedador

Para determinar el mejor sistema operativo para el nodo controlador se toma en cuenta las características de las versiones del sistema operativo Ubuntu *Server*, el cual cuenta con soporte de largo plazo LTS *Long Term Support*, así como un entorno sin interfaz gráfica usada en ambientes comerciales y la documentación existente para la versión.

Ubuntu realiza varias correcciones para estabilizar sus distintas versiones LTS entre estos cambios se tiene:

- Ubuntu *server* 20.04.4: Es considerada una versión anterior que cuenta con soporte hasta abril de 2025. Cuenta con mantenimiento de seguridad extendida, por sus siglas en inglés ESM - *Extended Security Maintenance*. Solo cuenta con sistema para 64 bits. Se pueden configurar o deshabilitar las notificaciones del sistema. Admite aplicaciones instantáneas [8] [9].
- Ubuntu *server* 18.04.6: Es considerada una versión anterior que cuenta con soporte hasta abril de 2023. Cuenta con ESM. Se puede realizar la instalación ligera de Ubuntu, elimina paquetes de *software*; además, se puede tener como ambiente de escritorio y navegador web. Es compatible con Netplan. Tiene una versión de Python preinstalada en sus paquetes. Cuenta con sistema tanto para 32 como para 64 bits. No silencia las notificaciones de sistema. Admite aplicaciones instantáneas [8] [10].

- Ubuntu server 16.04.7: Ya es considerada una versión antigua y que contó con soporte hasta abril de 2021, cuenta con ESM [11] [12].

Al estudiar cada distribución se considera a Ubuntu server 18.04.6 LTS, por su compatibilidad de actualización con versiones LTS, además por la gran cantidad de documentación que existe para esta versión. Se lo elige por su estabilidad y la corrección de errores frente a sus versiones anteriores, además de permitir interactuar como nodo controlador y contar con un tamaño aceptable para la realización de este proyecto [13].

Software de Simulación de Redes

Se presenta a continuación una comparativa entre varios sistemas de simulación de redes; estos permiten de forma gráfica, crear ambientes de análisis de redes con distintas tecnologías, además de entender el funcionamiento de distintos fabricantes de equipos de redes, poner a prueba distintas topologías. También permiten emular y simular ambientes académicos de estudio, para el aprendizaje y el reconocimiento de los comandos utilizados en las distintas marcas de los equipos. Algunos *softwares* permiten la emulación de los equipos, añadiendo al *software* las imágenes del sistema operativo de los equipos reales [14] [15] [16] [17] [18]. Los más representativos se los compara en la Tabla 1.1.

Tabla 1.1 Comparación entre *softwares* de simulación de redes

Software	GNS3	Eve-ng	Cisco Packet Tracer
Sistema Operativo	Windows posterior a 7. Windows server posterior a 2012. Mac OS posteriores a 10.9. Linux Esxi	Windows Windows server Mac OS, inferior a Big Sur. Linux Esxi posterior a 6.0.	Todos, dependiendo de la versión a instalar.
Necesita Hardware real	No	No	No
Emula Equipos Reales	Si	Si	No
Imágenes de equipos para el diseño	Posee predeterminadas y el usuario proporciona	Requiere de agentes y aplicaciones de terceros	Predeterminados
Cliente HTML5	Si	Si	No

A nivel profesional GNS3 es un gran motor de emulación y simulación con prestaciones que permiten desarrollar proyectos grandes con capacidades simuladas con lo cual su

ejecución al momento de crear ambientes controlados de forma intuitiva y sin procesos engorrosos hacen de esta herramienta la más útil y eficaz para realizar una práctica con tecnologías con lo es Ansible.

2 METODOLOGÍA

En este proyecto de titulación se planteó un tipo de investigación exploratoria aplicada, que permitió desarrollar estudios en ambientes poco investigados en su ejecución, permitió predecir los sucesos a modificaciones de configuraciones iniciales. También permitió resolver el problema de errores humanos y la distancia entre el equipo con el administrador, generando aportes en la parte de ejecución de su documentación actual como técnica investigativa.

Para continuar esta línea de investigación, la metodología de este proyecto está comprendida de los pasos que se describen a continuación:

Para instalar el nodo controlador, se creó una máquina virtual en el *software* VirtualBox *release* 6.1.34 Qt 5.6.2, en el cual se instaló la distribución Ubuntu 18.04 de la familia Linux, que hospeda la herramienta Ansible versión 2.9.27.

Para la configuración del *software* GNS3, se agregaron las imágenes correspondientes de los equipos de *networking* y los dispositivos finales que se utilizan. Para el análisis se estableció comunicación segura SSH entre el nodo controlador y los equipos de *networking*.

Se diseñó y se puso en ejecución el *playbook*, el cual fue creado para ser interactivo, en otras palabras, solicita el ingreso de datos al administrador al momento de ser ejecutado; estos datos fueron solicitados por medio del *prompt*, las tareas fueron ejecutadas de acuerdo con los datos ingresados. En primera instancia se configuró uno a uno los equipos de *networking* que forman parte de una topología; en estos se configuró interfaces, enrutamiento EIGRP, nombre, *banner*, contraseñas y *pool* de direcciones IPv4 para DHCP que va hacia las LAN.

Para finalizar se realizó las pruebas de funcionamiento de la configuración de los equipos de *networking*, haciendo uso de la herramienta Ansible, así como también, se verificó la comunicación entre una red LAN hacia otra mediante el protocolo de enrutamiento EIGRP.

3 RESULTADOS

El presente proyecto se enmarca en la implementación de Ansible para la configuración de equipos de *networking* de forma dinámica; la topología desarrollada cuenta con dos *routers*, dos equipos finales y su enrutamiento se lo realizó con el protocolo EIGRP. Ansible es instalado en una máquina virtual que cuenta con un sistema operativo Ubuntu, esta posee SSH para la comunicación con los equipos de *networking* y aloja los *playbooks* que se ejecutan de forma interactiva solicitando datos al usuario, las tareas de configuración de interfaces, *hostname*, *banner*, contraseñas de modo privilegiado, VTY y línea de consola. Además, una tarea del *playbook* es cargar un archivo al equipo donde se tiene un *pool* de direcciones para DHCP hacia la LAN de cada *router* en los dispositivos finales.

3.1 Instalación de Ansible en el nodo controlador

Creación de la Máquina Virtual con Ubuntu Server

La creación de la máquina virtual para el sistema operativo Ubuntu 18.04.6 LTS, se configuró con las siguientes características en VirtualBox, como se aprecia en la Figura 3.1.

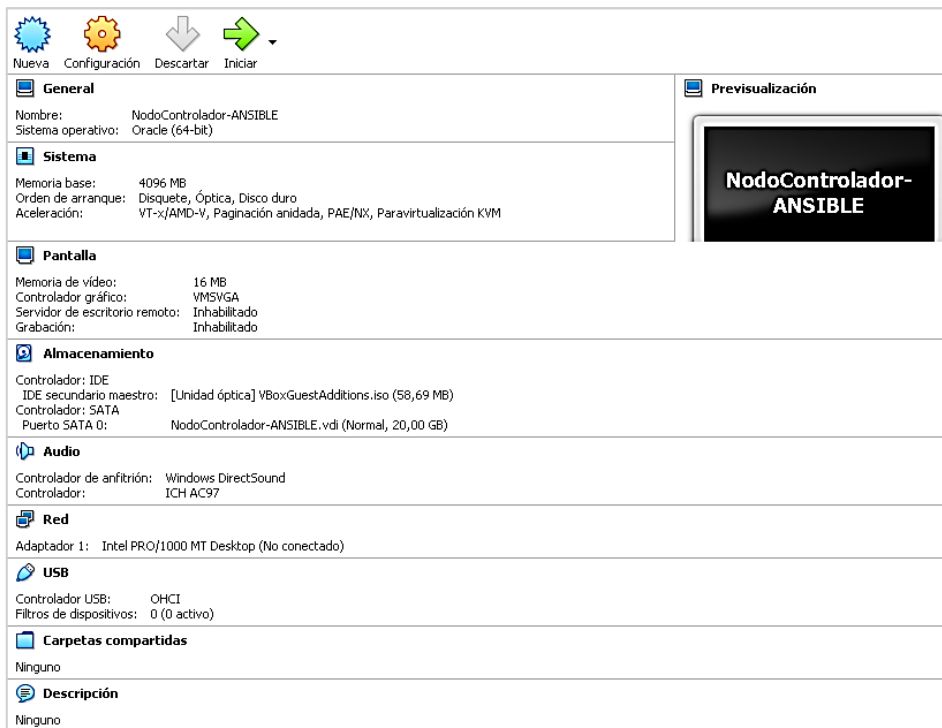


Figura 3.1 Características de la máquina virtual

En esta máquina virtual se instaló el sistema operativo Ubuntu 18.04.6 LTS, se lo puede apreciar en la Figura 3.2 al instante que termina su instalación.

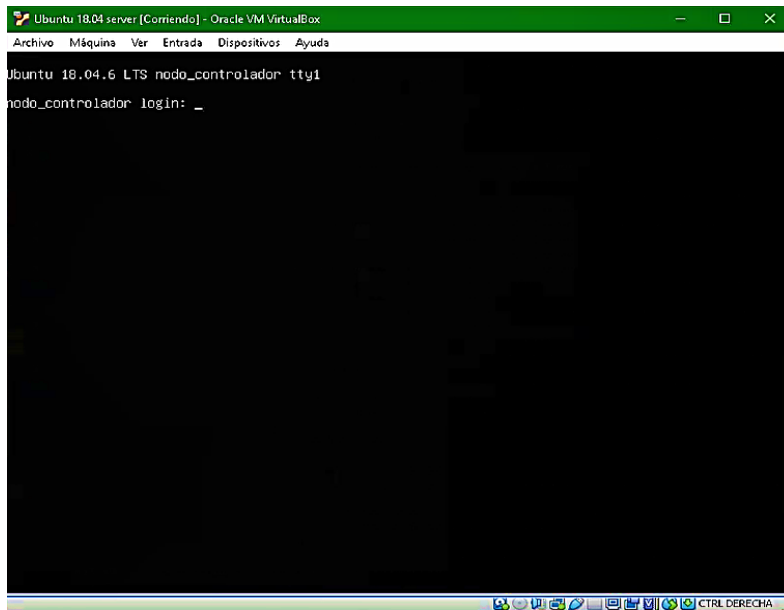


Figura 3.2 Ubuntu server 18.04.6 LTS, al arranque

Instalación de la Herramienta Ansible

Para iniciar con la instalación de la herramienta primero se debe verificar que la máquina virtual tenga conexión a Internet, esto se puede alcanzar añadiendo en las configuraciones de la máquina virtual un adaptador de red tipo NAT.

Se empezó instalando los paquetes que requiere Ansible, para realizar estas configuraciones, obsérvese la Figura 3.3, es necesario estar en modo super usuario por lo que se ejecutó el comando 1. En la misma figura se ejecutó el comando 2 el cual instala los paquetes de propiedades y repositorios del *software* para Ansible.

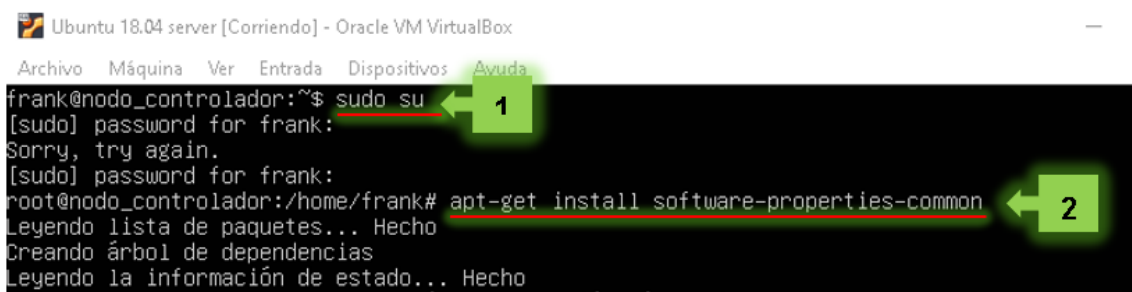
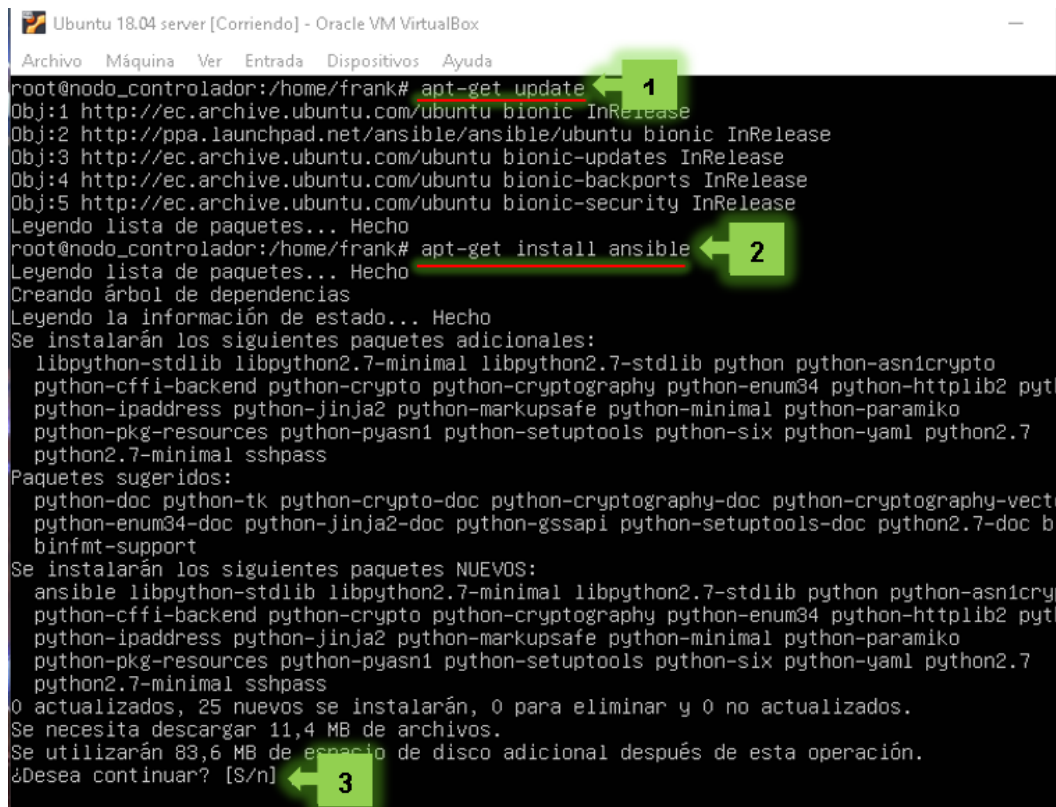


Figura 3.3 Ingreso a super usuario para instalar paquetes de Ansible

Para evitar conflictos de algún tipo, en este punto se realiza la descarga de actualización de paquetes con el comando 1 de la Figura 3.4. En la misma figura se ejecutó el comando 2, que es la instalación del paquete Ansible, el cual solicitará confirmar la

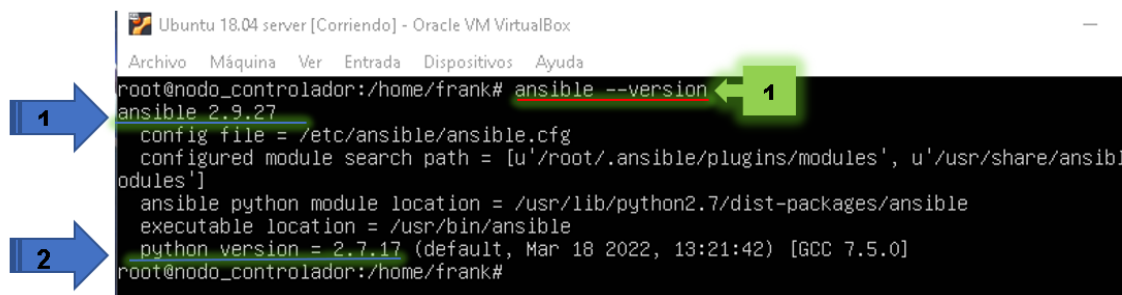
descarga de varios paquetes ingresando la letra S en español o Y en inglés que se vería como punto 3.



```
Ubuntu 18.04 server [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@nodo_controlador:/home/frank# apt-get update ← 1
Obj:1 http://ec.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu bionic-backports InRelease
Obj:5 http://ec.archive.ubuntu.com/ubuntu bionic-security InRelease
Leyendo lista de paquetes... Hecho
root@nodo_controlador:/home/frank# apt-get install ansible ← 2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-asn1crypto
 python-cffi-backend python-crypto python-cryptography python-enum34 python-httplib2 pyt
 python-ipaddress python-jinja2 python-markupsafe python-minimal python-paramiko
 python-pkg-resources python-pyasn1 python-setuptools python-six python-yaml python2.7
 python2.7-minimal sshpass
Paquetes sugeridos:
 python-doc python-tk python-crypto-doc python-cryptography-doc python-cryptography-vect
 python-enum34-doc python-jinja2-doc python-gssapi python-setuptools-doc python2.7-doc b
 binfmt-support
Se instalarán los siguientes paquetes NUEVOS:
 ansible libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-asn1cry
 python-cffi-backend python-crypto python-cryptography python-enum34 python-httplib2 pyt
 python-ipaddress python-jinja2 python-markupsafe python-minimal python-paramiko
 python-pkg-resources python-pyasn1 python-setuptools python-six python-yaml python2.7
 python2.7-minimal sshpass
0 actualizados, 25 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 11,4 MB de archivos.
Se utilizarán 83,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] ← 3
```

Figura 3.4 Actualización de paquetes e instalación de Ansible

En la Figura 3.5, se ejecutó el comando 1 para mostrar la correcta instalación y la versión de Ansible. La flecha azul 1 indica la versión de Ansible instalada, que es la 2.9.27; de igual forma se observa la versión de Python que se encuentra utilizando, que es la 2.7.17, flecha azul 2.



```
Ubuntu 18.04 server [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
root@nodo_controlador:/home/frank# ansible --version ← 1
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible
  modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Mar 18 2022, 13:21:42) [GCC 7.5.0]
root@nodo_controlador:/home/frank#
```

Figura 3.5 Versión de Ansible y de Python

3.2 Implementación de la topología en el simulador de red

Para que el *software* GNS3, reconozca la máquina virtual del nodo controlador, no solo es necesario colocarla como parte de los *templates* del *software*, sino también en sus

configuraciones de red debe estar el adaptador de red como Controlador genérico, esto se muestra en la Figura 3.6.

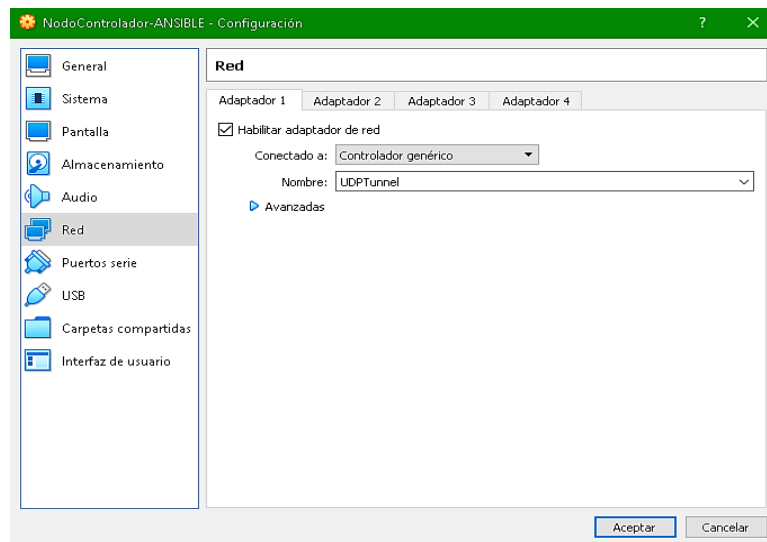


Figura 3.6 Cambios en adaptador de red virtual

Para armar la topología se realiza la instalación de las imágenes del *router* en GNS3 y de la computadora para las redes LAN; además, de la máquina virtual del nodo controlador. Estos equipos se los puede encontrar en el lado derecho de la interfaz de GNS3, refiriéndose a la Figura 3.7. El *router* se muestra con el número uno, la computadora con el número dos y el número tres es el nodo controlador.

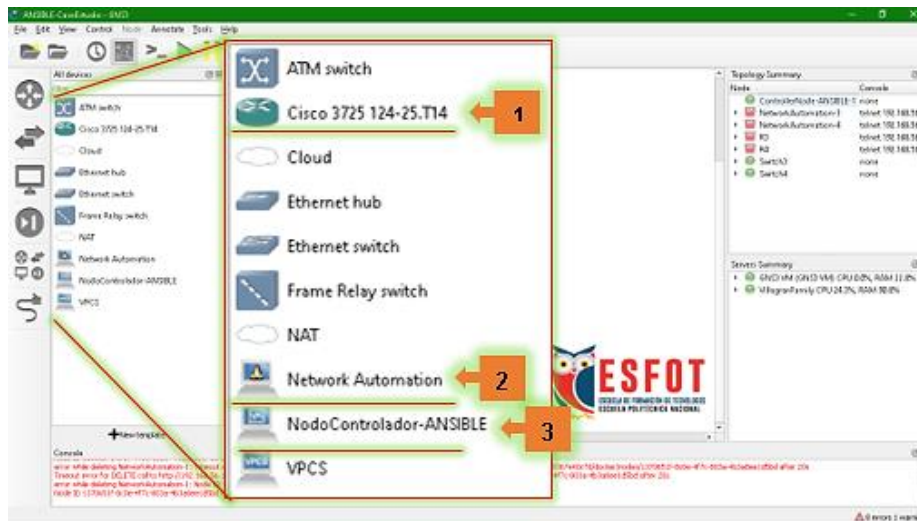


Figura 3.7 Equipos finales y de *networking* instalados en GNS3

Con estos equipos se presenta a continuación en la Figura 3.8, la topología desarrollada en este proyecto, donde se realizarán las configuraciones desde el nodo controlador hacia los *routers* del fabricante Cisco.

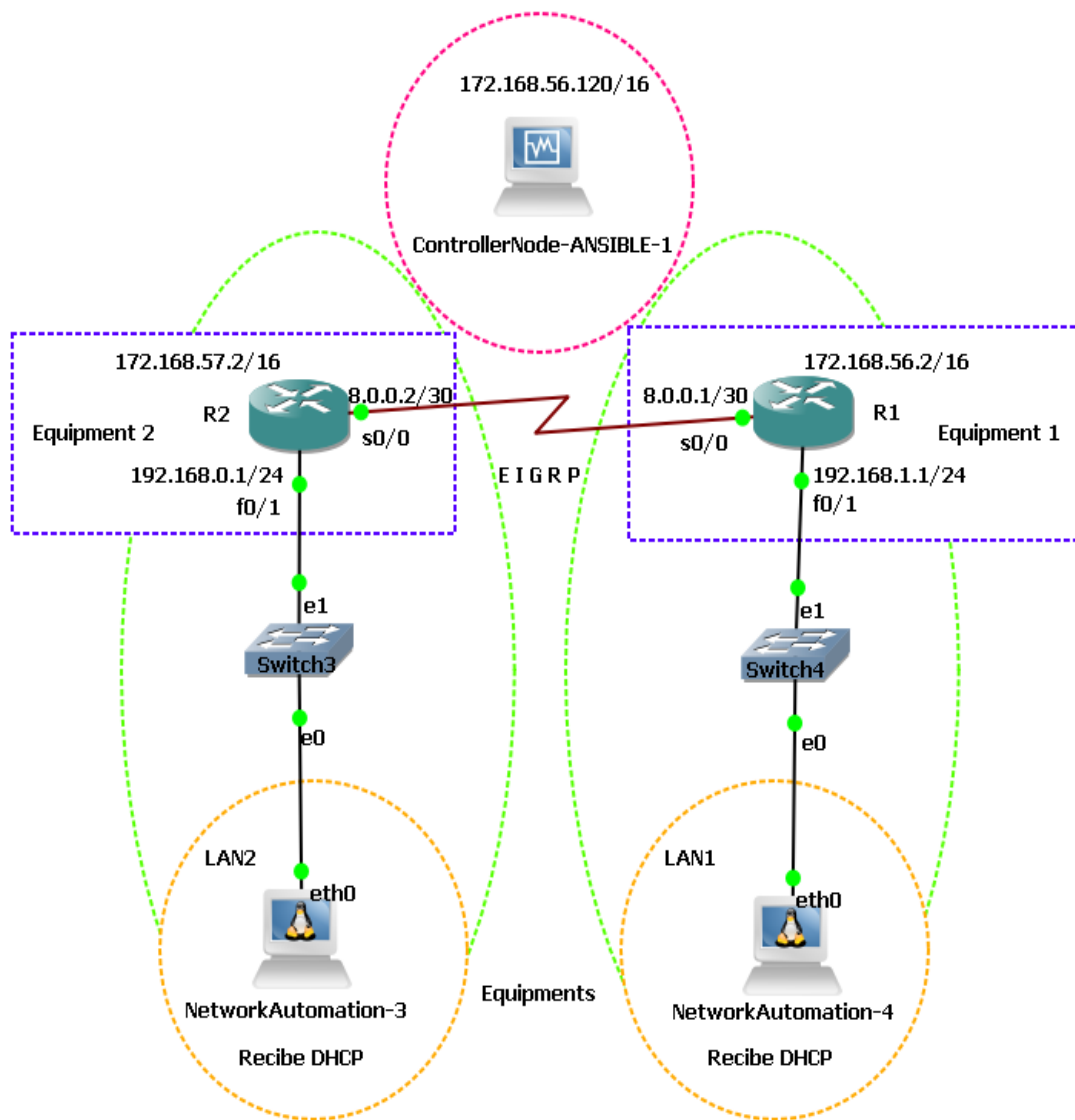


Figura 3.8 Topología de la red

Para que el nodo controlador tenga comunicación con los equipos primero se genera las llaves RSA con el comando 1 de la Figura 3.9, al ejecutarlo se pregunta si desea sobre escribirlo, se escribe la letra “y” para proceder.

```

frank@frank:/etc/ansible$ ssh-keygen ← 1
Generating public/private rsa key pair.
Enter file in which to save the key (/home/frank/.ssh/id_rsa):
/home/frank/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/frank/.ssh/id_rsa.
Your public key has been saved in /home/frank/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:GD02ENSFJ4utbDCb0BGRD0o9zQqFb8qkYy+uENegvy4 frank@frank
The key's randomart image is:
+---[RSA 2048]-----+
|  *+0 0.0.0          |
| +0+ 0 B =          |
| ..=0= + 0         |
|.0 0.* = 0         |
| * * + = S         |
| oB . .            |
| 0.0               |
| E. 0              |
|+++               |
+----[SHA256]-----+

```

Figura 3.9 Generación de llaves

Se requiere de configuraciones básicas en los *routers* para que la comunicación SSH se la pueda realizar, a continuación, en la Tabla 3.1 se detalla el comando y la acción que se ejecuta en el equipo.

Tabla 3.1 Comandos ejecutados en equipos

Número	Comando	Acción
1	<i>configure terminal</i>	Ingresar a modo de configuración privilegiada
2	<i>interface f0/0</i>	Abrir configuración de interfaz Fast Ethernet 0/0
3	<i>ip address 172.168.56.2 255.255.0.0</i>	Asignar la dirección a la interfaz
4	<i>no shutdown</i>	Levantar la interfaz
5	<i>exit</i>	Salir del modo configuración de la interfaz
6	<i>ip domain-name zona.com</i>	Asignar un nombre de dominio
7	<i>username zona1 privilege 15 secret zona1</i>	Crear un usuario y asignar una contraseña secreta
8	<i>crypto key generate rsa</i>	Generar contraseña para SSH
9	<i>ip ssh version 2</i>	Utilizar SSH en su versión 2
10	<i>enable secret EPN</i>	Asignar contraseña de enable
11	<i>line vty 0 4</i>	Abrir configuración de VTY
12	<i>transport input ssh</i>	Indicar el transporte de SSH
13	<i>login local</i>	Solicitar Ingreso para comunicación
14	<i>exit</i>	Salir de configuración VTY

Estas configuraciones se ven ejecutadas en los equipos de *networking* como se aprecia en la Figura 3.10. Al final se guardó las configuraciones con el comando *write & read* o *wr*.

```
R1#
R1#configure terminal ← 1
Enter configuration commands one per line. End with CNTL/Z.
R1(config)#interface f0/0 ← 2
R1(config-if)#ip address 172.168.56.2 255.255.0.0 ← 3
R1(config-if)#no shutdown ← 4
R1(config-if)#exit ← 5
R1(config)#ip domain-name zona.com ← 6
R1(config)#username zonal privilege 15 secret zonal ← 7
R1(config)#crypto key generate rsa ← 8
*Mar 1 00:47:44.579: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to
up
*Mar 1 00:47:45.579: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet
0/0, changed state to up
R1(config)#crypto key generate rsa
The name for the keys will be: R1.zona.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

StoDgo(config)#
*Mar 1 03:48:28.279: %SSH-5-ENABLED: SSH 1.99 has been enabled
StoDgo(config)#ip ssh ve
StoDgo(config)#ip ssh version 2 ← 9
StoDgo(config)#enable se
StoDgo(config)#enable secret EPN ← 10
StoDgo(config)#line vty 0 4 ← 11
StoDgo(config-line)#tra
StoDgo(config-line)#transport in
StoDgo(config-line)#transport input ss
StoDgo(config-line)#transport input ssh ← 12
StoDgo(config-line)#login local ← 13
StoDgo(config-line)#exit ← 14
```

Figura 3.10 Configuraciones para SSH en los *routers*

Para la comunicación SSH entre el nodo controlador y los *routers*, se copia la llave privada del equipo de *networking* en el nodo controlador, con lo cual se van a comunicar entre sí, esto se realiza por medio del comando 1 de la Figura 3.11.

```

frank@frank:/etc/ansible$ ssh-copy-id zona1@172.168.56.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: /home/frank/.ssh/id_rsa.pub
The authenticity of host '172.168.56.2 (172.168.56.2)' can't be established.
RSA key fingerprint is SHA256:1FLvdvrf01qDQ2qukriniJ+e1rbM+K3A5R8s9mx9AMI.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
Password:
Line has invalid autocommand "exec sh -c 'cd ; umask 077 ; mkdir -p .ssh && { [-z `tail -1c .ssh/au
thorized_keys 2>/dev/null` ] || echo >> .ssh/authorized_keys ; } && cat >> .ssh/authorized_keys || e
xit 1 ; if type restorecon >/dev/null 2>&1 ; then restorecon -F .ssh .ssh/authorized_"
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'zona1@172.168.56.2'"
and check to make sure that only the key(s) you wanted were added.

```

Figura 3.11 Envío de copia de la llave generada

Al final se podrá acceder al equipo, por medio de comunicación SSH, como se observa en la Figura 3.12, con esto se verifica que hay comunicación desde el nodo controlador con el equipo de *networking* mediante este protocolo.

```

frank@frank:/etc/ansible$ ssh zona1@172.168.56.2
Password:
R1#

```

Figura 3.12 Acceso al equipo por SSH

3.3 Implementación de las configuraciones mediante

Ansible

Primero es necesario ubicar el archivo de inventario en la dirección */etc/ansible/Inventory*, donde se va a colocar el registro de los equipos que van a ser configurados con Ansible, esto se muestra en la Figura 3.13. Este archivo almacena un nombre para el equipo entre corchetes **[equipment1]**, una etiqueta con el nombre asignado al *router R1*, así como también un grupo de variables **[equipment1:vars]** que corresponde a este equipo, donde se coloca el usuario y la contraseña, credenciales que usará Ansible para establecer la comunicación SSH con el equipo **ansible_host=172.168.56.2**. Lo mismo se tiene para el *router R2* con sus respectivas credenciales. En las líneas finales de este archivo inventario se indica que todos los equipos pertenecen al mismo grupo denominado **[equipments]** este es muy útil si se requiere configurar a todos estos equipos declarados al mismo tiempo.

```
NodoControlador-ANSIBLE [Corriendo] - Oracle VM VirtualBox
GNU nano 2.9.3

[equipment1]
R1 ansible_host=172.168.56.2

[equipment1:vars]
ansible_user=zona1
ansible_password=zona1

[equipment2]
R2 ansible_host=172.168.57.2

[equipment2:vars]
ansible_user=zona2
ansible_password=zona2

[equipments:vars]
gather_facts=False
ansible_python_interpreter=/usr/bin/python
ansible_connection=network_cli
ansible_network_os=ios

[equipments:children]
equipment1
equipment2
```

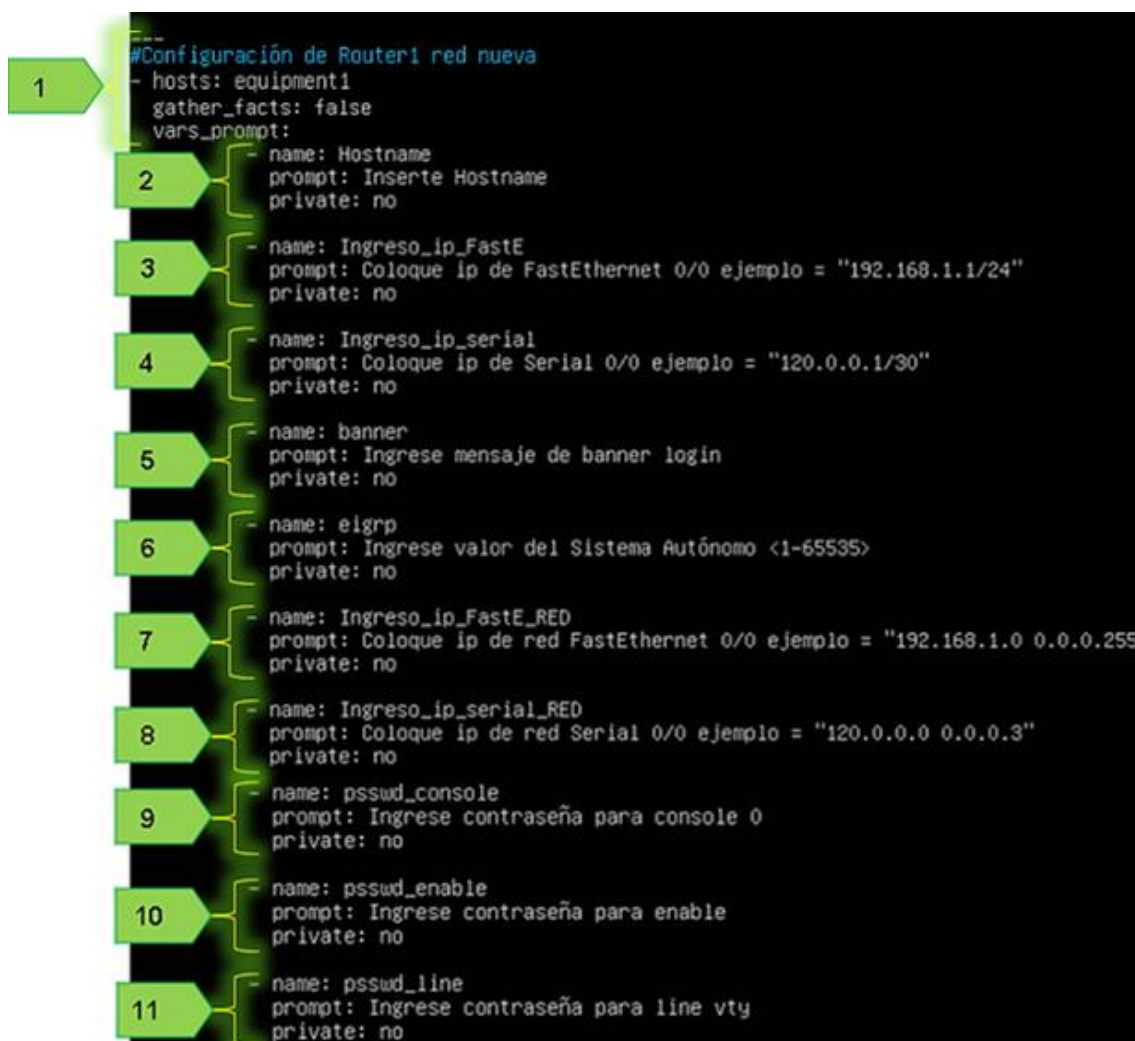
Figura 3.13 Declaración de equipos en Inventario

Diseño del *playbook*

Para la creación del *playbook* que es un archivo en lenguaje YAML, se toman varias consideraciones, como son:

- Siempre se inician los archivos con tres guiones medios seguidos.
- Los comentarios deben iniciar con el carácter #.
- Las tabulaciones no pueden ser usadas.
- Para la estructura de la escritura se utilizan los espacios.
- El guion medio determina la subdivisión entre objetos.
- Después del uso de los dos puntos siempre va un espacio.
- Después del uso de un igual no va con espacio.
- Las variables no pueden iniciar con números.
- Los equipos para configurar deben estar registrados en el archivo *Inventory*.
- Las subdivisiones pueden también contener otras subdivisiones.

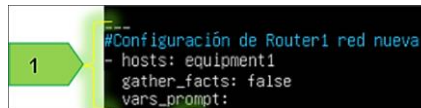
En la Figura 3.14, se encuentra la primera subdivisión del *playbook*, denominada **vars_prompt**: de variables volátiles para la solicitud de datos. Esta subdivisión está numerada y separada por secciones.



```
---
#Configuración de Router1 red nueva
- hosts: equipment1
  gather_facts: false
  vars_prompt:
    1 - name: Hostname
      2 - prompt: Inserte Hostname
      private: no
    3 - name: Ingreso_ip_FastE
      4 - prompt: Coloque ip de FastEthernet 0/0 ejemplo = "192.168.1.1/24"
      private: no
    5 - name: Ingreso_ip_serial
      6 - prompt: Coloque ip de Serial 0/0 ejemplo = "120.0.0.1/30"
      private: no
    7 - name: banner
      8 - prompt: Ingrese mensaje de banner login
      private: no
    9 - name: elgrp
      10 - prompt: Ingrese valor del Sistema Autónomo <1-65535>
      private: no
    11 - name: Ingreso_ip_FastE_RED
      12 - prompt: Coloque ip de red FastEthernet 0/0 ejemplo = "192.168.1.0 0.0.0.255"
      private: no
    13 - name: Ingreso_ip_serial_RED
      14 - prompt: Coloque ip de red Serial 0/0 ejemplo = "120.0.0.0 0.0.0.3"
      private: no
    15 - name: pssud_console
      16 - prompt: Ingrese contraseña para console 0
      private: no
    17 - name: pssud_enable
      18 - prompt: Ingrese contraseña para enable
      private: no
    19 - name: pssud_line
      20 - prompt: Ingrese contraseña para line vty
      private: no
```

Figura 3.14 Declaración de variables del *playbook*

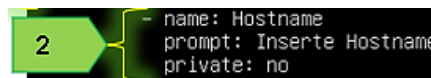
En la sección 1, se encuentra la cabecera del archivo `.yml`, inicia con tres guiones medios. En la siguiente línea se encuentra un comentario que en este caso se coloca la acción y el equipo a configurar, en la siguiente línea se utiliza el guion para declarar un primer objeto o **host** con el nombre `equipment1` del *router* R1. En la siguiente línea se utiliza el módulo de hechos de configuración, **gather_facts=false**, el cual recopila datos de todos los módulos y los fusiona, en este caso no lo requiere por lo que se colocó en falso. En la línea final de esta sección se indica que se van a declarar las variables volátiles **vars_prompt**, que las ingresa un usuario por medio del *prompt* y que van a ser utilizadas en la ejecución del libro de jugadas o *playbook*. Esta sección se la observa en la Figura 3.15.

A screenshot of an Ansible playbook section. On the left, a green arrow with the number '1' points to the first line of code. The code is as follows:

```
---  
#Configuración de Router1 red nueva  
- hosts: equipment1  
  gather_facts: false  
  vars_prompt:
```

Figura 3.15 Sección 1 de variables del *playbook*

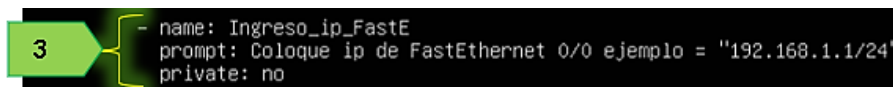
En la sección 2, se realiza una subdivisión con un guion donde se declara el nombre de la variable **Hostname**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese Hostname** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se la aprecia en la Figura 3.16.

A screenshot of an Ansible playbook section. On the left, a green arrow with the number '2' points to the first line of code. The code is as follows:

```
- name: Hostname  
  prompt: Inserte Hostname  
  private: no
```

Figura 3.16 Sección 2 de variables del *playbook*

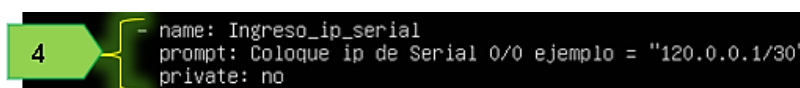
En la sección 3, se realiza una subdivisión con un guion donde se declara el nombre de la variable **Ingreso_ip_FastE**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Coloque ip de FastEthernet 0/0 ejemplo = "192.168.1.1/24"** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se muestra en la Figura 3.17.

A screenshot of an Ansible playbook section. On the left, a green arrow with the number '3' points to the first line of code. The code is as follows:

```
- name: Ingreso_ip_FastE  
  prompt: Coloque ip de FastEthernet 0/0 ejemplo = "192.168.1.1/24"  
  private: no
```

Figura 3.17 Sección 3 de variables del *playbook*

En la sección 4, se realiza una subdivisión con un guion donde se declara el nombre de la variable **Ingreso_ip_serial**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Coloque ip de Serial 0/0 ejemplo = "120.0.0.1/30"** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se la puede analizar en la Figura 3.18.

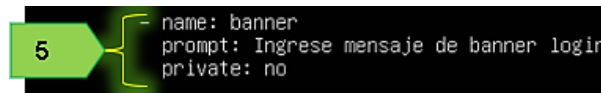
A screenshot of an Ansible playbook section. On the left, a green arrow with the number '4' points to the first line of code. The code is as follows:

```
- name: Ingreso_ip_serial  
  prompt: Coloque ip de Serial 0/0 ejemplo = "120.0.0.1/30"  
  private: no
```

Figura 3.18 Sección 4 de variables del *playbook*

En la sección 5, se realiza una subdivisión con un guion donde se declara el nombre de la variable **banner**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese mensaje de banner login** al ejecutar el

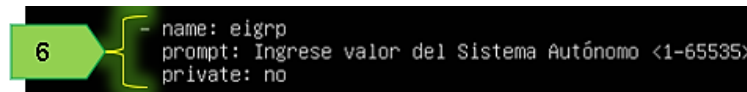
playbook, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. La estructura de esta sección se la puede ver en la Figura 3.19.

A screenshot of a configuration block for section 5 of a playbook. It features a green arrow-shaped label with the number '5' on the left. To its right, a black box contains the following text: '- name: banner', 'prompt: Ingrese mensaje de banner login', and 'private: no'.

```
5 { - name: banner
    prompt: Ingrese mensaje de banner login
    private: no
```

Figura 3.19 Sección 5 de variables del *playbook*

En la sección 6, se realiza una subdivisión con un guion donde se declara el nombre de la variable **eigrp**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese valor del Sistema Autónomo <1-65535>** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección es apreciable en la Figura 3.20.

A screenshot of a configuration block for section 6 of a playbook. It features a green arrow-shaped label with the number '6' on the left. To its right, a black box contains the following text: '- name: eigrp', 'prompt: Ingrese valor del Sistema Autónomo <1-65535>', and 'private: no'.

```
6 { - name: eigrp
    prompt: Ingrese valor del Sistema Autónomo <1-65535>
    private: no
```

Figura 3.20 Sección 6 de variables del *playbook*

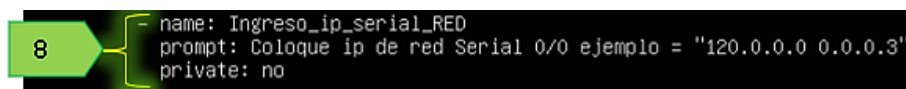
En la sección 7, se realiza una subdivisión con un guion donde se declara el nombre de la variable **Ingreso_ip_FastE_RED**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Coloque ip de red FastEthernet 0/0 ejemplo = "192.168.1.0 0.0.0.255"** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se la evidencia en la Figura 3.21.

A screenshot of a configuration block for section 7 of a playbook. It features a green arrow-shaped label with the number '7' on the left. To its right, a black box contains the following text: '- name: Ingreso_ip_FastE_RED', 'prompt: Coloque ip de red FastEthernet 0/0 ejemplo = "192.168.1.0 0.0.0.255"', and 'private: no'.

```
7 { - name: Ingreso_ip_FastE_RED
    prompt: Coloque ip de red FastEthernet 0/0 ejemplo = "192.168.1.0 0.0.0.255"
    private: no
```

Figura 3.21 Sección 7 de variables del *playbook*

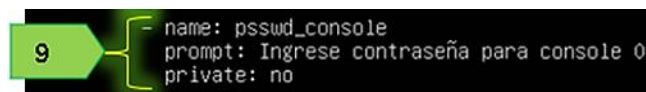
En la sección 8, se realiza una subdivisión con un guion donde se declara el nombre de la variable **Ingreso_ip_serial_RED**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Coloque ip de red Serial 0/0 ejemplo = "120.0.0.0 0.0.0.3"** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se la puede verificar en la Figura 3.22.

A screenshot of a configuration block for section 8 of a playbook. It features a green arrow-shaped label with the number '8' on the left. To its right, a black box contains the following text: '- name: Ingreso_ip_serial_RED', 'prompt: Coloque ip de red Serial 0/0 ejemplo = "120.0.0.0 0.0.0.3"', and 'private: no'.

```
8 { - name: Ingreso_ip_serial_RED
    prompt: Coloque ip de red Serial 0/0 ejemplo = "120.0.0.0 0.0.0.3"
    private: no
```

Figura 3.22 Sección 8 de variables del *playbook*

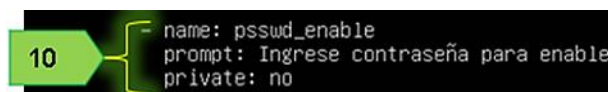
En la sección 9, se realiza una subdivisión con un guion donde se declara el nombre de la variable **psswd_console**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese contraseña para console 0** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección se observa en conjunto con la Figura 3.23.



```
9 - name: psswd_console
  prompt: Ingrese contraseña para console 0
  private: no
```

Figura 3.23 Sección 9 de variables del *playbook*

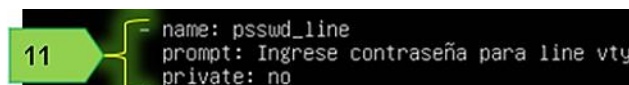
En la sección 10, se realiza una subdivisión con un guion donde se declara el nombre de la variable **psswd_enable**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese contraseña para enable** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección es apreciable en la Figura 3.24.



```
10 - name: psswd_enable
  prompt: Ingrese contraseña para enable
  private: no
```

Figura 3.24 Sección 10 de variables del *playbook*

En la sección 11, se realiza una subdivisión con un guion donde se declara el nombre de la variable **psswd_line**, en la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Ingrese contraseña para line vty** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta **private:no**. Esta sección está ligada a la Figura 3.25.



```
11 - name: psswd_line
  prompt: Ingrese contraseña para line vty
  private: no
```

Figura 3.25 Sección 11 de variables del *playbook*

Luego de haber configurado la subdivisión de **vars_prompt**, se procede a crear las tareas donde se utilizan cada una de las variables. Para iniciar con esta subdivisión se inicia con la línea **tasks** seguido de dos puntos, a continuación, se detallan las secciones de cada tarea del *playbook* que se encuentran marcadas en la Figura 3.26.

```

tasks:
  - name: Asignación de hostname (AnsibleControlado#)
    cisco.ios.ios_hostname:
      config:
        hostname: "{{ Hostname }}"
        state: merged
  - name: Asignación de ip a cada interfaz
    cisco.ios.ios_l3_interfaces:
      config:
        - name: FastEthernet0/0
          ipv4:
            - address: "{{ Ingreso_ip_FastE }}"
        - name: Serial0/0
          ipv4:
            - address: "{{ Ingreso_ip_serial }}"
        state: merged
  - name: Habilita Interfaces
    cisco.ios.ios_interfaces:
      config:
        - name: FastEthernet0/0
          enabled: true
          description: Configured with Ansible "Designed by Frank J. Villagrán A."
        - name: Serial0/0
          enabled: true
          description: Configured with Ansible "Designed by Frank J. Villagrán A."
  - name: Mensaje para acceso
    cisco.ios.ios_banner:
      banner: login
      text: |
        {{ banner }}
      state: present
  - name: Corrección de errores
    cisco.ios.ios_config:
      lines:
        - no router eigrp {{ eigrp }}
        - exit
      save_when: always
  - name: Configuración de Protocolo EIGRP (60000)
    cisco.ios.ios_config:
      lines:
        - router eigrp {{ eigrp }}
        - network {{ Ingreso_ip_FastE_RED }}
        - network {{ Ingreso_ip_serial_RED }}
      save_when: always
  - name: Configuración contraseña Line console 0
    cisco.ios.ios_config:
      lines:
        - line console 0
        - password {{ psswd_console }}
        - login
        - do wr
      save_when: always
  - name: Configuración contraseña enable
    cisco.ios.ios_config:
      lines:
        - enable password {{ psswd_enable }}
        - do wr
      save_when: always
  - name: Configuración contraseña line vty
    cisco.ios.ios_config:
      lines:
        - line vty 1 4
        - password {{ psswd_line }}
        - login local
        - do wr
      save_when: always

```

Figura 3.26 Tareas del *playbook*

En la sección 1 se declara el nombre de la tarea **Asignación de *hostname*** (**AnsibleControlado#**), en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_hostname* que va a ejecutarse en esta tarea. Luego la siguiente línea establece un nombre de host para IOS, el cual reconoce la variable **Hostname** ingresada por el usuario previamente. En la última línea se agrega el estado en el cual se van a quedar las configuraciones, en este caso combinada o *merged* para que se cambien o se guarden los valores que tengan similitud. Nótese esta sección en la Figura 3.27.



```
tasks:
- name: Asignación de hostname (AnsibleControlado#)
  cisco.ios.ios_hostname:
  config:
    hostname: "{{ Hostname }}"
  state: merged
```

Figura 3.27 Sección 1 de tareas del *playbook*

En la sección 2 se declara el nombre de la tarea **Asignación de ip a cada interfaz**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_l3_interfaces* que va a ejecutarse en esta tarea. La siguiente línea indica que se van a ingresar configuraciones para interfaces IOS, la siguiente línea se especifica qué interfaz se va a configurar, para ello se añade una nueva subdivisión para las configuraciones de *FastEthernet 0/0*. Luego en la siguiente línea se indica el formato de direcciones que es IPv4. Se abre otra subdivisión que configura la dirección de la interfaz, el cual reconoce la variable **Ingreso_ip_FastE** ingresada por el usuario previamente.

Para la siguiente línea se considera el nivel jerárquico de las subdivisiones donde se añade una subdivisión para las configuraciones de la interfaz Serial 0/0, la siguiente línea indica que el formato de direcciones es IPv4, para la siguiente línea se abre una subdivisión que configura la dirección de la interfaz, el cual reconoce la variable **Ingreso_ip_serial** ingresada por el usuario previamente. En la última línea se agrega el estado en el cual se van a quedar las configuraciones en este caso combinada o *merged* para que se cambien o se guarden los valores que tengan similitud, con esto se cierra la subdivisión de esta tarea, véase la Figura 3.28.

```

- name: Asignación de ip a cada interfaz
  cisco.ios.ios_l3_interfaces:
    config:
      - name: FastEthernet0/0
        ipv4:
          - address: "{{ Ingreso_ip_FastE }}"
      - name: Serial0/0
        ipv4:
          - address: "{{ Ingreso_ip_serial }}"
    state: merged

```

Figura 3.28 Sección 2 de tareas del *playbook*

En la sección 3 se inicia con una subdivisión donde se declara el nombre de la tarea **Habilita Interfaces**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_interfaces* que va a ejecutarse en esta tarea. Luego la siguiente línea especifica a qué interfaz se va a configurar, se añade una subdivisión para la configuración de la interfaz *FastEthernet 0/0*. En la siguiente línea se habilita el enrutamiento, para la siguiente se añade una descripción a la interfaz **Configured with Ansible “Designed by Frank J. Villagran A.”**, estas acciones se repiten para la interfaz Serial 0/0, como se detalla en la Figura 3.29.

```

- name: Habilita Interfaces
  cisco.ios.ios_interfaces:
    config:
      - name: FastEthernet0/0
        enabled: true
        description: Configured with Ansible "Designed by Frank J. Villagrán A."
      - name: Serial0/0
        enabled: true
        description: Configured with Ansible "Designed by Frank J. Villagrán A."

```

Figura 3.29 Sección 3 de tareas del *playbook*

En la sección 4 se inicia con una subdivisión donde se declara el nombre de la tarea **Mensaje para acceso**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_banner* que va a ejecutarse en esta tarea. La siguiente línea permite establecer el tipo de **banner**, en este caso de **login**, la línea siguiente **text** tiene el carácter “|” que indica que el mensaje inicia en la siguiente línea reconociendo la variable **banner**, ingresada por el usuario previamente. En la última línea se agrega el estado en el cual se van a quedar las configuraciones en este caso se requiere que el estado sea presente, según la documentación del módulo usado. A continuación, se aprecia esta sección en la Figura 3.30.

```

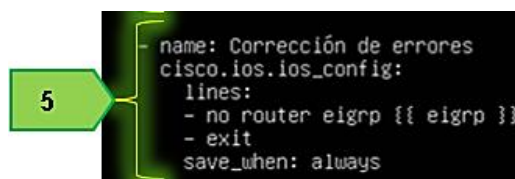
- name: Mensaje para acceso
  cisco.ios.ios_banner:
    banner: login
    text: |
      {{ banner }}
    state: present

```

Figura 3.30 Sección 4 de tareas del *playbook*

En la sección 5 se inicia con una subdivisión donde se declara el nombre de la tarea **Corrección de errores**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea. Luego la siguiente línea establece líneas que serán iguales a las de la configuración del dispositivo que va a ser configurado, en la siguiente línea se utiliza el comando **no router eigrp** y se reconoce la variable **eigrp** previamente ingresada por el usuario.

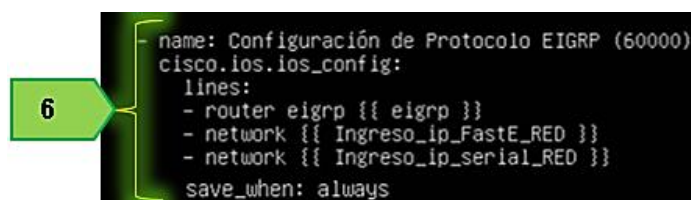
En este contexto, se agrega una línea para salir de las configuraciones de modo privilegiado, en la última línea se agrega guardar-cuando en el cual se van a guardar las configuraciones. Obsérvese esta sección en la Figura 3.31.

A screenshot of an Ansible playbook section. On the left, a green arrow with the number '5' points to the start of a task. The task is defined as follows:

```
- name: Corrección de errores
  cisco.ios.ios_config:
    lines:
      - no router eigrp {{ eigrp }}
      - exit
    save_when: always
```

Figura 3.31 Sección 5 de tareas del *playbook*

En la sección 6 se inicia con una subdivisión donde se declara el nombre de la tarea **Configuración de Protocolo EIGRP (60000)**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea. La siguiente línea establece líneas que serán iguales a las de la configuración del dispositivo que va a ser configurado, en la siguiente línea se utiliza el comando **router eigrp** y se reconoce la variable **eigrp** ingresada previamente por el usuario. La siguiente línea tiene el comando **network** con la variable **Ingreso_ip_FastE_RED** donde se establece la red directamente conectada a la FastEthernet 0/0; de igual manera se coloca el comando **network** con la variable **Ingreso_ip_serial_RED** donde se establece la red directamente conectada a la interfaz Serial 0/0. En la última línea se agrega el guardar-cuando el cual permite guardar las configuraciones. Dicha sección se la puede analizar en la Figura 3.32.

A screenshot of an Ansible playbook section. On the left, a green arrow with the number '6' points to the start of a task. The task is defined as follows:

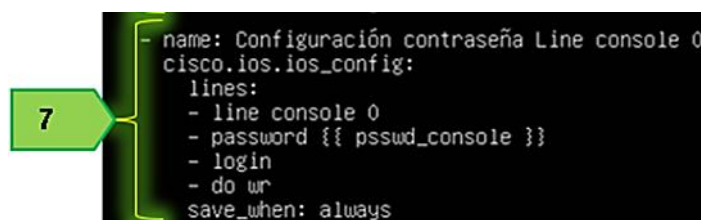
```
- name: Configuración de Protocolo EIGRP (60000)
  cisco.ios.ios_config:
    lines:
      - router eigrp {{ eigrp }}
      - network {{ Ingreso_ip_FastE_RED }}
      - network {{ Ingreso_ip_serial_RED }}
    save_when: always
```

Figura 3.32 Sección 6 de tareas del *playbook*

En la sección 7 se inicia con una subdivisión donde se declara el nombre de la tarea **Configuración contraseña Line Console 0**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta

tarea. La siguiente línea establece líneas que serán iguales a las de la configuración del dispositivo que va a ser configurado, en la siguiente línea se utiliza el comando **line console 0** para ingresar al modo de configuración de línea de consola.

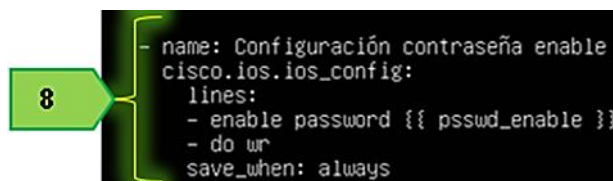
En la siguiente línea se configura la contraseña con la variable **psswd_console** previamente ingresada por el usuario, la siguiente línea establece el **login**, en la siguiente línea se coloca el comando **do wr** para guardar los cambios y en la última se agrega el guardar-cuando el cual permite guardar las configuraciones siempre. Se puede denotar esta sección en la Figura 3.33.



```
- name: Configuración contraseña Line console 0
  cisco.ios.ios_config:
    lines:
      - line console 0
      - password {{ psswd_console }}
      - login
      - do wr
    save_when: always
```

Figura 3.33 Sección 7 de tareas del *playbook*

En la sección 8 se inicia con una subdivisión donde se declara el nombre de la tarea **Configuración contraseña enable**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea. La siguiente línea establece líneas que serán iguales a las de la configuración del dispositivo que va a ser configurado, en la siguiente línea se utiliza el comando **enable password**, con la variable **psswd_enable** previamente ingresada por el usuario. Se coloca el comando **do wr** para guardar los cambios y en la última línea se agrega el guardar-cuando el cual permite guardar las configuraciones. Esta sección se encuentra ilustrada en la Figura 3.34.



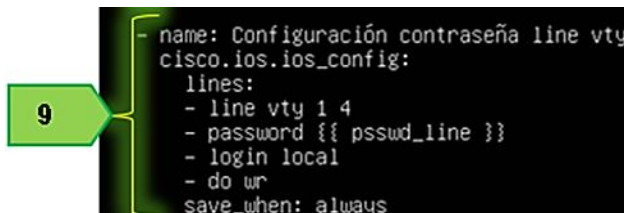
```
- name: Configuración contraseña enable
  cisco.ios.ios_config:
    lines:
      - enable password {{ psswd_enable }}
      - do wr
    save_when: always
```

Figura 3.34 Sección 8 de tareas del *playbook*

En la sección 9 se inicia con una subdivisión donde se declara el nombre de la tarea **Configuración contraseña line vty**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea. Luego en la siguiente línea se establece líneas que serán iguales a las de la configuración del dispositivo que va a ser configurado, en la siguiente línea se utiliza el

comando **line vty 1 4**, se ingresa a configuraciones de line vty para de allí configurar la contraseña con la variable **psswd_line** previamente ingresada por el usuario.

La siguiente línea establece el **login local**, en la siguiente línea se coloca el comando **do wr** para guardar los cambios y en la última línea se agrega el guardar cuando, el cual permite guardar las configuraciones siempre. Esto se muestra en la Figura 3.35.



```
- name: Configuración contraseña line vty
  cisco.ios.ios_config:
    lines:
      - line vty 1 4
      - password {{ psswd_line }}
      - login local
      - do wr
    save_when: always
```

Figura 3.35 Sección 9 de tareas del *playbook*

En el segundo equipo, el *playbook* difiere del primero porque en este se añade la configuración del reloj del serial.

Dentro de la subdivisión **vars_prompt**, donde se declararon las variables volátiles, se añade la variable que solicita la velocidad del **clock rate** para el segundo equipo. En estas líneas agregadas se realiza una subdivisión con un guion donde se declara el nombre de la variable **cl_rate**. En la siguiente línea se indica el conjunto de caracteres que se mostrarán como solicitud de datos **Coloque velocidad de clock rate ejemplo = "64000"** al ejecutar el *playbook*, y en su tercera línea se indica que la entrada del usuario no va a estar oculta, esto se aprecia en la Figura 3.36.



```
- name: cl_rate
  prompt: Coloque velocidad de clock rate ejemplo = "64000"
  private: no
```

Figura 3.36 Variable para *clock rate*

De igual forma en las tareas se añaden las líneas de la Figura 3.37, se inicia con una subdivisión donde se declara el nombre de la tarea **Reloj de interfaz serial**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea. Luego **lines** donde se ingresa a la **interface serial 0/0**, en la siguiente línea se configuró el **clock rate** con la variable **cl_rate** previamente ingresada por el usuario. En la siguiente línea se coloca el comando **do wr** para guardar los cambios y en la última línea se agrega la línea guardar-cuando que permite guardar siempre las configuraciones.


```

- name: Reloj de interfaz serial
  cisco.ios.ios_config:
    lines:
      - interface serial 0/0
      - clock rate {{ cl_rate }}
      - do wr
    save_when: always

```

Figura 3.37 Tarea para *clock rate*

En los dos *routers* se añade una tarea más a los *playbooks* diseñados para que mediante DHCP proporcionen direcciones IP a sus dispositivos finales. En la Figura 3.38 se inicia con una subdivisión donde se declara el nombre de la tarea **Asignación de pool de direcciones para DHCP**, en la siguiente línea se utiliza la colección de Ansible *ansible-galaxy collection cisco.ios.ios_config* que va a ejecutarse en esta tarea, luego se establece una ruta del archivo de configuraciones a cargar. Al final se requiere una tarea extra la cual simplemente ejecuta el guardado de las configuraciones realizadas con comandos anteriormente mencionados.

```

- name: Asignación de pool de direcciones para DHCP
  cisco.ios.ios_config:
    src: /etc/ansible/config_DHCP1.txt

- name: Guardar cambios
  cisco.ios.ios_config:
    lines:
      - do wr
    save_when: always

```

Figura 3.38. Pool de direcciones para DHCP

El archivo de configuraciones para aplicar DHCP, se redacta de la misma forma como se suele apreciar en el establecimiento del *running-config*. Este archivo se lo observa en la Figura 3.39 con las configuraciones del *router* R1; además, se configura una interfaz de *loopback* para el *router*. En la Figura 3.40 se aprecian las configuraciones DHCP para el *router* R2.



```

GNU nano 2.9.3 config_DHCP1.txt
|
| ip dhcp excluded-address 192.168.1.1
|
| ip dhcp pool LANR1
| network 192.168.1.0 255.255.255.0
| default-router 192.168.1.1
|
| interface FastEthernet0/1
| ip address 192.168.1.1 255.255.255.0
| no shutdown
|
| interface lo 1
| ip address 1.1.1.1 255.255.255.255
|

```

Figura 3.39 Configuración DHCP para el *router* R1



```
GNU nano 2.9.3 config_DHCP2.txt
ip dhcp excluded-address 192.168.0.1

ip dhcp pool LANR2
network 192.168.0.0 255.255.255.0
default-router 192.168.0.1

interface FastEthernet0/1
ip address 192.168.0.1 255.255.255.0
no shutdown

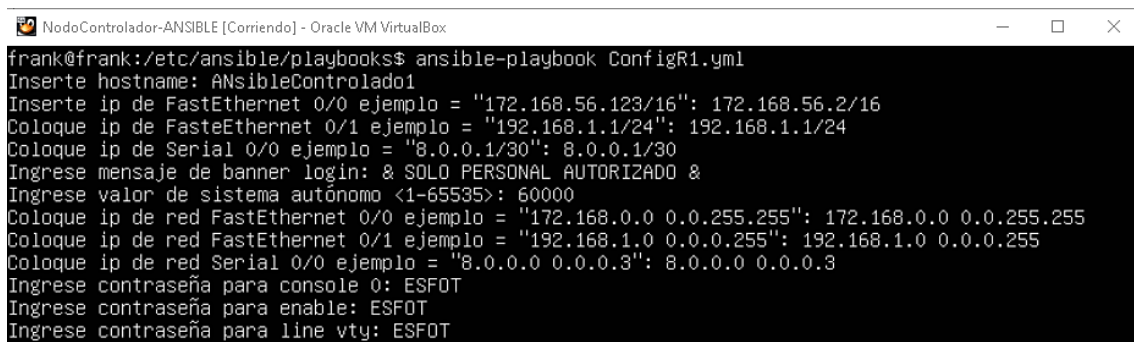
interface lo 1
ip address 2.2.2.2 255.255.255.255
```

Figura 3.40 Configuración DHCP para *router* R2

3.4 Pruebas de funcionamiento

Datos ingresados por el usuario

Como se ha indicado las configuraciones del equipo son realizadas por medio de Ansible, en primera instancia permite ingresar varios de los datos de configuración de forma interactiva, por lo tanto, la primera verificación de funcionamiento del *playbook* corresponde al ingreso de datos del usuario. Los datos ingresados para el *router* R1 se pueden apreciar en la Figura 3.41.



```
frank@frank:/etc/ansible/playbooks$ ansible-playbook ConfigR1.yml
Inserte hostname: ANSibleControlado1
Inserte ip de FastEthernet 0/0 ejemplo = "172.168.56.123/16": 172.168.56.2/16
Coloque ip de FastEthernet 0/1 ejemplo = "192.168.1.1/24": 192.168.1.1/24
Coloque ip de Serial 0/0 ejemplo = "8.0.0.1/30": 8.0.0.1/30
Ingrese mensaje de banner login: & SOLO PERSONAL AUTORIZADO &
Ingrese valor de sistema autónomo <1-65535>: 60000
Coloque ip de red FastEthernet 0/0 ejemplo = "172.168.0.0 0.0.255.255": 172.168.0.0 0.0.255.255
Coloque ip de red FastEthernet 0/1 ejemplo = "192.168.1.0 0.0.0.255": 192.168.1.0 0.0.0.255
Coloque ip de red Serial 0/0 ejemplo = "8.0.0.0 0.0.0.3": 8.0.0.0 0.0.0.3
Ingrese contraseña para console 0: ESF0T
Ingrese contraseña para enable: ESF0T
Ingrese contraseña para line vty: ESF0T
```

Figura 3.41 Datos ingresados por usuario, *router* R1

Los datos ingresados por el usuario para el *router* R2 se pueden apreciar en la Figura 3.42.

```

frank@frank:/etc/ansible/playbooks$ ansible-playbook ConfigR2.yml
Inserte hostname: AnsibleControlado2
Coloque ip de FasteEthernet 0/1 ejemplo = "192.168.0.1/24": 192.168.0.1/24
Coloque ip de Serial 0/0 ejemplo = "8.0.0.2/30": 8.0.0.2/30
Coloque velocidad de clock rate ejemplo = "64000": 64000
Ingrese mensaje de banner login: & SOLD PERSONAL AUTORIZADO AC2 &
Ingrese valor de sistema autónomo <1-65535>: 60000
Coloque ip de red FastEthernet 0/1 ejemplo = "192.168.0.0 0.0.0.255": 192.168.0.0 0.0.0.255
Coloque ip de red Serial 0/0 ejemplo = "8.0.0.0 0.0.0.3": 8.0.0.0 0.0.0.3
Ingrese contraseña para console 0: ESF0T
Ingrese contraseña para enable: ESF0T
Ingrese contraseña para line vty: ESF0T

```

Figura 3.42 Datos ingresados por usuario, *router* R2

Respuesta de *playbook* a su ejecución

Luego del ingreso de los datos por parte del usuario, comienza la ejecución de las tareas en el router R1. El primer equipo se encuentra registrado en el inventario de *host* de Ansible con la etiqueta R1, esta es la que se muestra en la confirmación de cambio de configuraciones al momento de ejecutar el *playbook*. La primera tarea que se observa en la Figura 3.26, se verifica en la ejecución con el rótulo *changed: [R1]*, de color amarillo y muestra el nombre de la tarea realizada: Asignación de *Hostname*, esto se aprecia en la Figura 3.43.

```

PLAY [equipment1] *****
TASK [Asignación de Hostname] *****
changed: [R1]

```

Figura 3.43 Tarea Asignación de *Hostname*

Para las tareas siguientes se replica el mismo formato de la primera tarea antes mencionada, además se muestran alertas en dos de las tareas que son atributos propios del módulo *ansible-galaxy collections* utilizado para estas configuraciones. La documentación de este módulo establece como recomendación no deshabilitarlas indicando que el formato de ingreso de datos al equipo debe tener el mismo formato que tienen las configuraciones en el *running-config*. En la Figura 3.44, se aprecia la ejecución de las tareas y su verificación en el equipo donde se están realizando los cambios, en este caso para el *router* R1. Al terminar de ejecutar el *playbook*, se muestra una recapitulación de tareas, que indica las tareas que no tuvieron conflicto con el rótulo verde *ok=11* y además el rótulo amarillo *changed=11* indica las tareas que ejecutaron cambios en el equipo.

```

TASK [Asignación de Hostname] *****
changed: [R1]

TASK [Asignación de ip a cada interfaz] *****
changed: [R1]

TASK [Habilita interfaces] *****
changed: [R1]

TASK [Mensaje de inicio] *****
changed: [R1]

TASK [Corrección de errores] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar
to how they appear if present in the running configuration on device
changed: [R1]

TASK [Configuración de protocolo eigrp] *****
changed: [R1]

TASK [Configuración contraseña console] *****
changed: [R1]

TASK [Configuración contraseña enable] *****
changed: [R1]

TASK [Configuración contraseña line vty] *****
changed: [R1]

TASK [Asignación de pool de direcciones para DHCP] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar
to how they appear if present in the running configuration on device including the indentation
changed: [R1]

TASK [Guardar cambios] *****
changed: [R1]

PLAY RECAP *****
R1      : ok=11  changed=11  unreachable=0  failed=0  skipped=0  rescued=
0      ignored=0

frank@frank:/etc/ansible/playbooks$ _

```

Figura 3.44 Confirmación de ejecución de tareas *router* R1

El *router* R2 se encuentra registrado en el inventario de host de Ansible, con la etiqueta R2, está es la que se muestra en la confirmación de cambio de configuraciones al momento de ejecutar el *playbook*, como se ilustra en la Figura 3.45. Los cambios se aprecian con el rótulo *changed: [12]* de color amarillo y muestra en dos tareas la alerta de ingreso de datos al equipo, el cual debe tener el mismo formato del *running-config*. Al terminar de ejecutar el *playbook*, se muestra una recapitulación de tareas, que indica las tareas que no tuvieron conflicto con el rótulo verde *ok=12* y además el rótulo amarillo indica las tareas que realizaron cambios en el equipo, *changed=12*.

```
NodoControlador-ANSIBLE [Corriendo] - Oracle VM VirtualBox
PLAY [equipment2] *****
TASK [Asignación de Hostname] *****
changed: [R2]
TASK [Asignación de ip a cada interfaz] *****
changed: [R2]
TASK [Asignación velocidad de clock rate] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar
to how they appear if present in the running configuration on device
changed: [R2]
TASK [Habilita interfaces] *****
changed: [R2]
TASK [Mensaje de inicio] *****
changed: [R2]
TASK [Corrección de errores] *****
changed: [R2]
TASK [Configuración de protocolo eigrp] *****
changed: [R2]
TASK [Configuración contraseña console] *****
changed: [R2]
TASK [Configuración contraseña enable] *****
changed: [R2]
TASK [Configuración contraseña line vty] *****
changed: [R2]
TASK [Asignación de pool de direcciones para DHCP] *****
[WARNING]: To ensure idempotency and correct diff the input configuration lines should be similar
to how they appear if present in the running configuration on device including the indentation
changed: [R2]
TASK [Guardar cambios] *****
changed: [R2]
PLAY RECAP *****
R2      ok=12  changed=12  unreachable=0  failed=0  skipped=0  rescued=0
0      ignored=0
frank@frank:/etc/ansible/playbooks$
```

Figura 3.45 Confirmación de ejecución de tareas *router* R2

Consolas de los equipos de *networking*

En la Figura 3.46, se aprecia la consola del *router* R1, donde se registran los cambios en las configuraciones que fueron realizadas remotamente, el usuario que los realizó y el tipo de conexión que se usó para tal hecho, indicando incluso la dirección IP del nodo controlador. Los datos ingresados modificaron las configuraciones del equipo, se evidencia el *banner login*, la solicitud de la contraseña y el *hostname*; datos ingresados por el usuario según la Figura 3.41.

```

*Mar 1 06:06:15.506: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:06:26.446: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:06:45.510: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:06:50.970: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 60000: Neighbor 8.0.0.2 (Serial0/0) is down: interface down
*Mar 1 06:06:53.598: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:07:05.526: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 60000: Neighbor 8.0.0.2 (Serial0/0) is up: new adjacency
*Mar 1 06:07:08.882: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:07:22.558: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:07:33.482: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:07:44.946: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:07:55.018: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
*Mar 1 06:08:02.446: %SYS-5-CONFIG_I: Configured from console by zona1 on vty0 (172.168.56.120)
& SOLO PERSONAL AUTORIZADO AC1 &

User Access Verification

Password:
AnsibleControlado1#

```

Figura 3.46 Registro de configuraciones router R1

Al ejecutar el comando **show running-config**, en el router R1 se muestran los cambios según los datos ingresados por el usuario, como es la contraseña de modo privilegiado y las configuraciones del pool de direcciones para la LANR1. Esto se aprecia en la Figura 3.47.

```

!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$ayRh$XpIRwsPmg4q5as.4In2L40
enable password ESFOT
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.1.1
!
ip dhcp pool LANR1
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
!
!
no ip domain lookup
ip domain name zona.com
!
multilink bundle-name authenticated
!

```

Figura 3.47 show running-config router R1, parte 1


```

R2 NetworkAutomation
*Mar 1 05:38:41.414: %DUAL-5-NBRCHANGE: IP-EIGRP
rial0/0) is down: Interface Goodbye received
*Mar 1 05:38:48.278: %DUAL-5-NBRCHANGE: IP-EIGRP
rial0/0) is up: new adjacency
*Mar 1 05:54:42.214: %DUAL-5-NBRCHANGE: IP-EIGRP
rial0/0) is down: Interface Goodbye received
*Mar 1 05:54:49.650: %DUAL-5-NBRCHANGE: IP-EIGRP
rial0/0) is up: new adjacency
SOLO PERSONAL AUTORIZADO AC2

User Access Verification

Password:
% Password: timeout expired!
Password:
% Password: timeout expired!
Password:
AnsibleControlado2#
AnsibleControlado2#sh run
Building configuration...

Current configuration : 1796 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname AnsibleControlado2
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$1.Z9$iEu2nyWCYcrhQqtoN/eDQ0
enable password ESPOT
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
no ip dhcp use vrf connected
--More-- █
solarwinds | Solar-PuTTY free tool

```

Figura 3.50 show running-config router R2, parte 1

A continuación, en la Figura 3.51, se muestra la configuración del *pool* de direcciones configurado con el archivo de la Figura 3.40.

```

R2 NetworkAutomation-3
!
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.0.1
!
ip dhcp pool LANR2
 network 192.168.0.0 255.255.255.0
 default-router 192.168.0.1

```

Figura 3.51 show running-config router R2, parte 2

A continuación, en la Figura 3.52, se muestra la configuración de la interfaz de *loopback* (como se detalla en el archivo de configuración del *pool* de direcciones DHCP observado en la Figura 3.40). También se puede apreciar cada interfaz del *router* R2 con su dirección IP y descripción, además de las configuraciones del protocolo de enrutamiento EIGRP del sistema autónomo 60000. Todos estos cambios según las tareas del

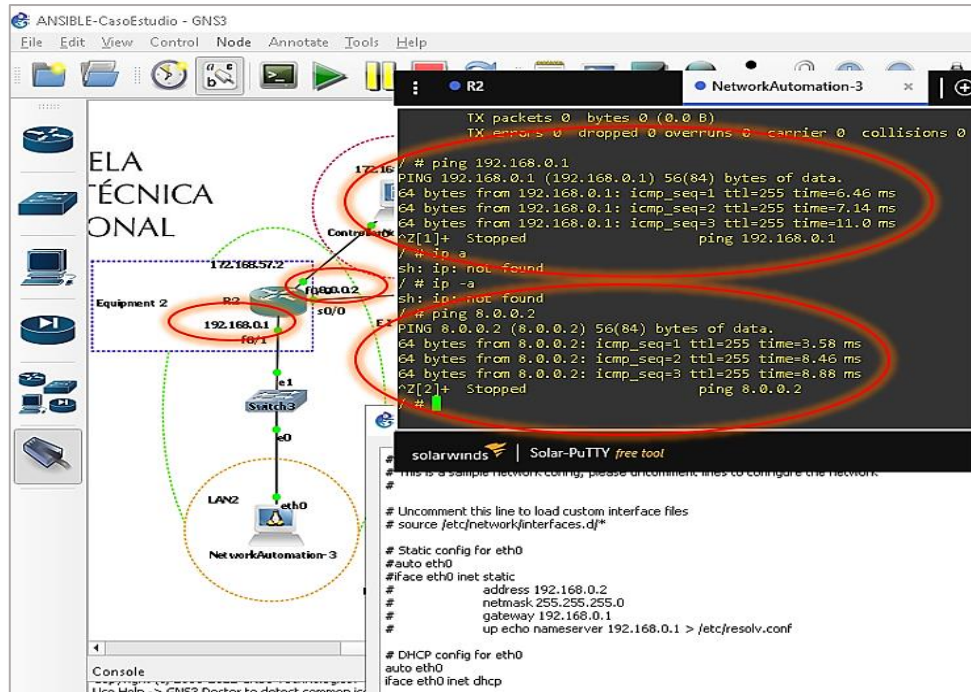


Figura 3.56 Ping desde la PC de la LAN2 a la puerta de enlace y a la interfaz serial

Una vez verificado el ping desde la PC de la LAN2 a las interfaces del *router* R2, se continúa con la verificación a la interfaz del *router* R1, específicamente hacia la interfaz serial 0/0 del mismo; esto se observa en la Figura 3.57.

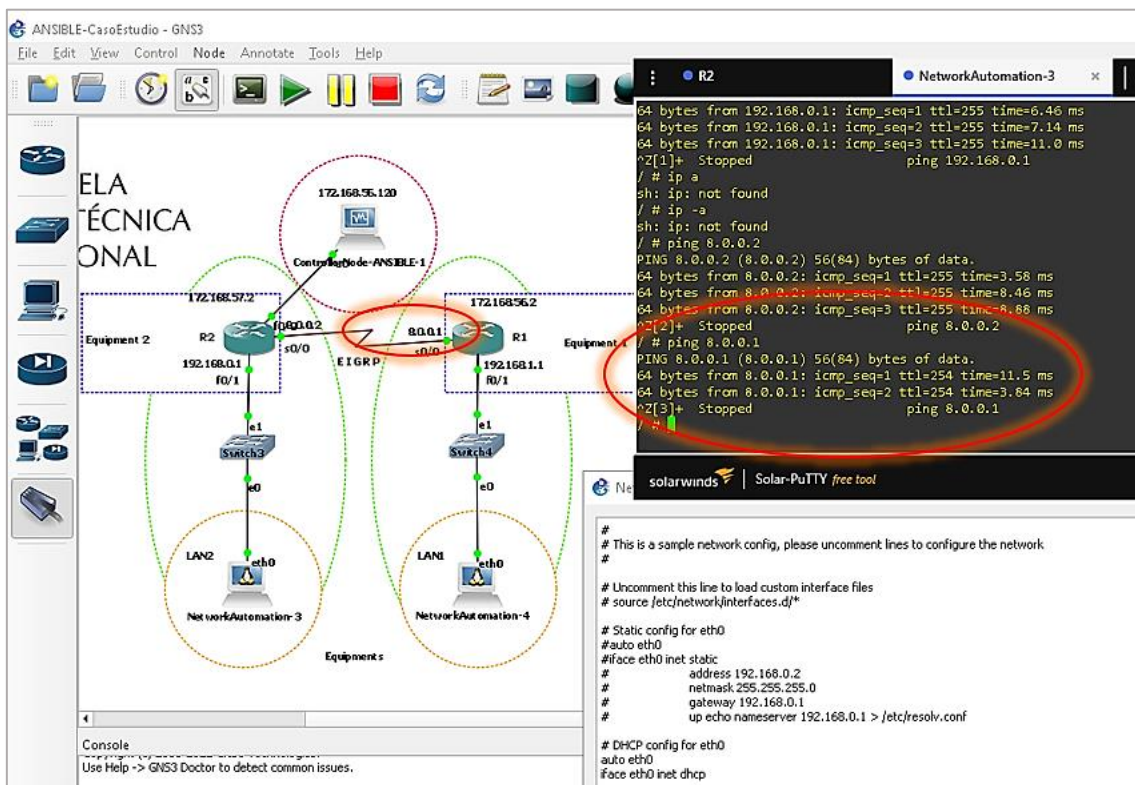


Figura 3.57 Ping desde la PC de la LAN2 hacia la interfaz serial del *router* R1

Se continúa con el ping desde la PC de la LAN2 hacia la interfaz FastEthernet del *router* R1 o puerta de enlace configurada de la LAN1, finalmente se realiza el ping a la PC de la LAN1 como se aprecia en la Figura 3.58.

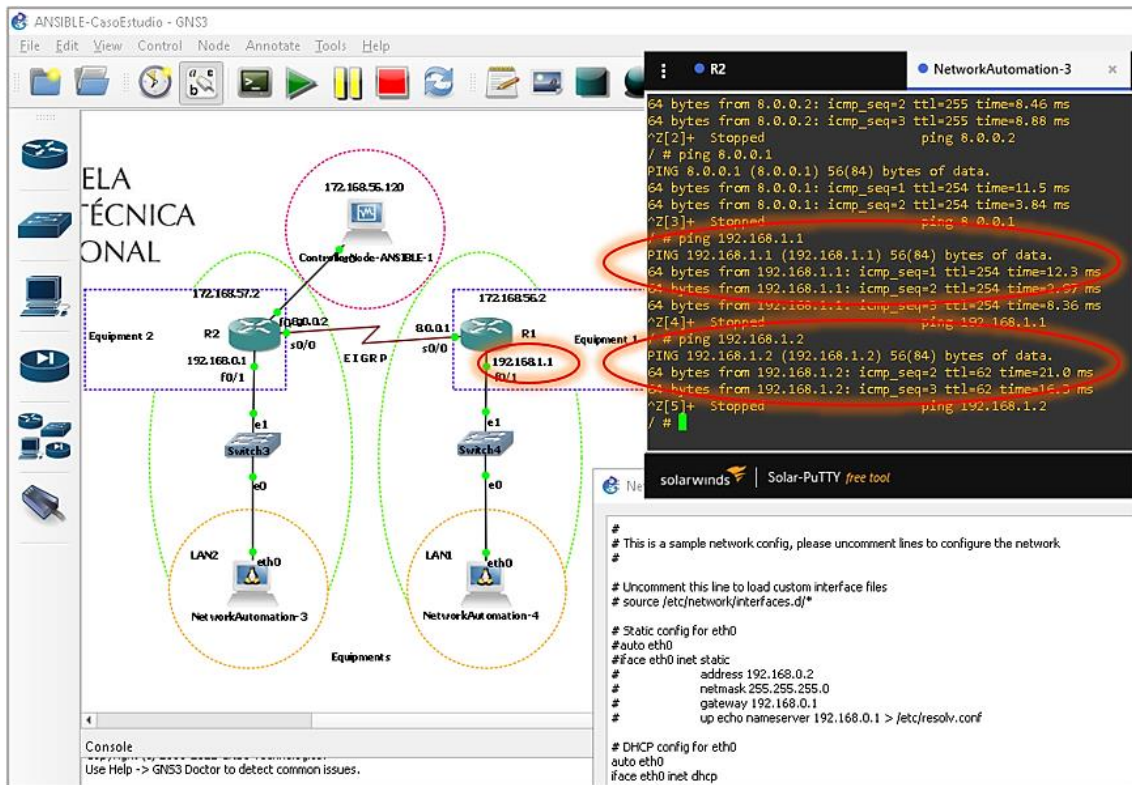


Figura 3.58 Ping desde la PC de la LAN2 hacia la puerta de enlace de la LAN1 y a la PC de la LAN1

En la Figura 3.58, se demostró el ping desde un equipo final a otro, entre LANs. Al haber verificado que el ping llega desde la LAN2 hasta la LAN1 se demuestra que el enrutamiento mediante EIGRP es correcto.

Ahora para iniciar la verificación del ping de regreso desde la LAN1 hacia la LAN2, primero se realiza el ping a la puerta de enlace de la LAN1, desde la PC de la LAN1, al verificarlo se procede a realizar el ping hacia la interfaz serial 0/0 del *router* R1, como se muestra en la Figura 3.59.

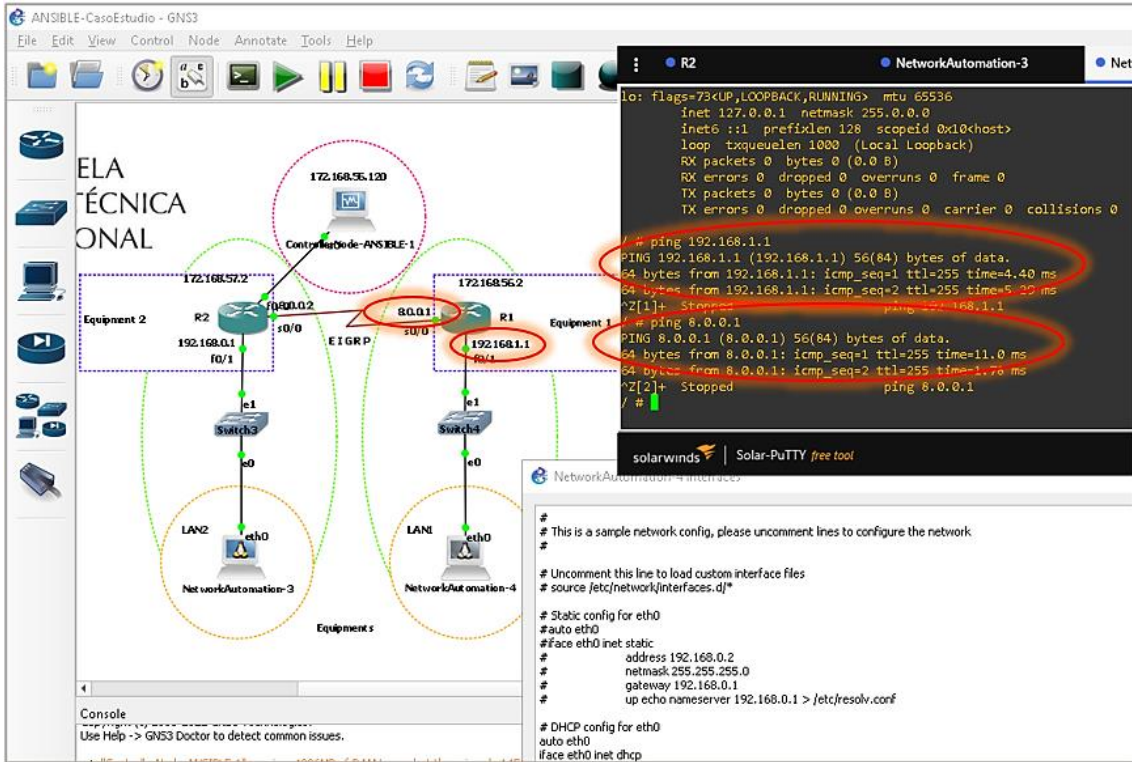


Figura 3.59 Ping desde la PC de la LAN1 hacia las interfaces del *router* R1

Al haber verificado este ping, se realiza el ping hacia la interfaz serial 0/0 del *router* R2, esta ejecución se observa en la Figura 3.60.

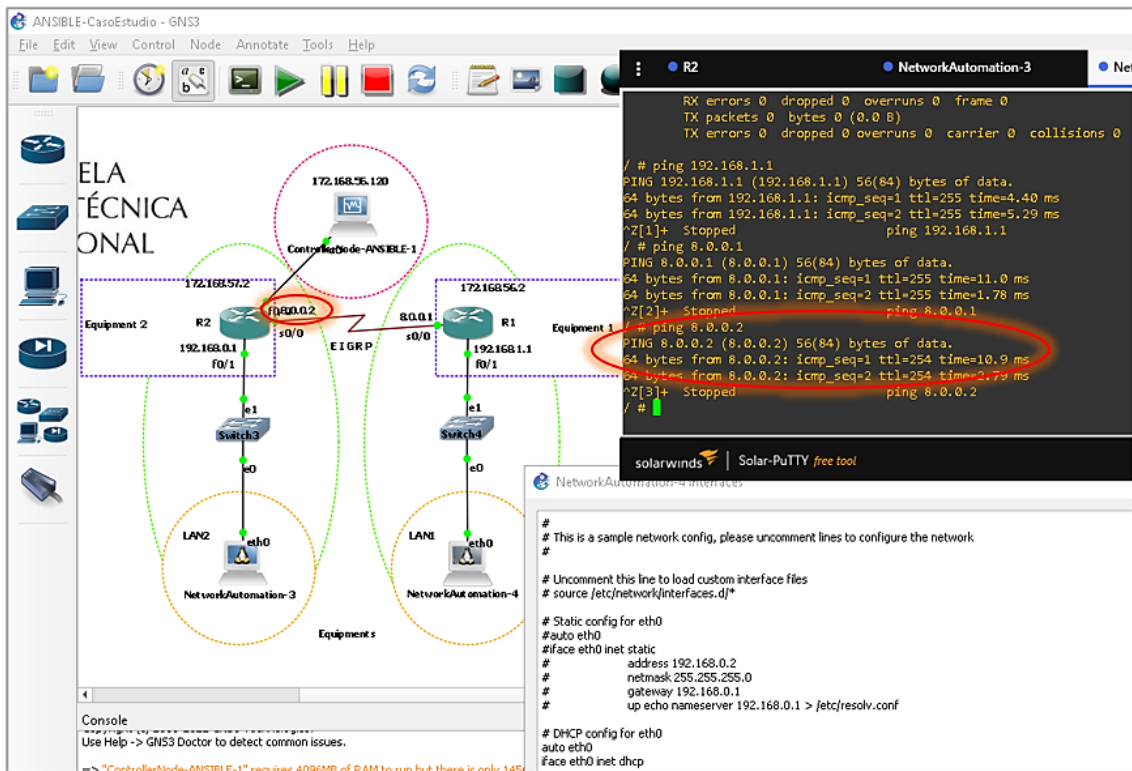


Figura 3.60 Ping desde la PC de la LAN1 hacia la interfaz serial del *router* R2

Finalmente se realiza el ping desde la PC de la LAN1 a la puerta de enlace de la LAN2 y por último a la PC de la LAN2, como se demuestra en la Figura 3.61. Al haber obtenido satisfactoriamente el ping entre LANs ahora sí se verifica el enrutamiento EIGRP en total funcionamiento.

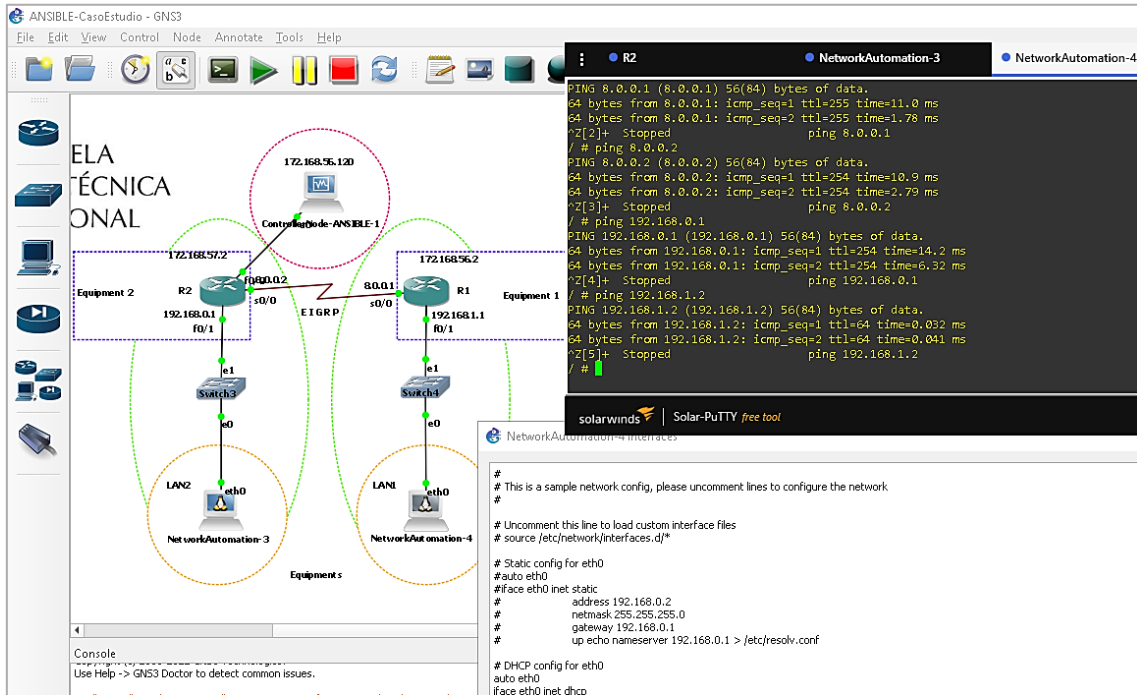


Figura 3.61 Ping entre equipos finales de LAN1 hacia LAN2

Con estas pruebas de verificación se afirma que la ejecución de este proyecto ha sido realizada de forma satisfactoria cumpliendo cada objetivo en su totalidad, así como se verifica las funciones de una red totalmente configurada mediante la herramienta Ansible.

4 CONCLUSIONES

- La revisión técnica de las características del *software* y los elementos necesarios del simulador de red GNS3, previo a la instalación de Ansible, permitió reconocer los principales requerimientos para un adecuado trabajo con la herramienta de DevOps.
- Para establecer la topología de red del proyecto, fue necesario un *software* de simulación de red como lo es GNS3, ya que permite definir equipos reales que se apegan más a la realidad de un sistema operativo implementado en un puesto de trabajo. Gracias al *software* GNS3 se utilizó el equipo *Network Automation* que simula las características de un sistema operativo Ubuntu, además permitió configurar DHCP automático por medio del archivo de configuraciones propio del mismo. En este contexto se permite el estudio de diversas áreas, trabajar en ellas y desarrollar proyecciones de implementación futuras.
- Para la comunicación SSH, es necesario que el archivo ubicado en la dirección */etc/ssh/sshd_config*, se encuentre la línea **KexAlgoritms +diffie-hellman-group1-sha1**, ya que esta permite evitar el error al encontrar métodos de intercambio de claves que coincidan. Este error se muestra al momento de establecer comunicación SSH con los equipos de *networking*, además luego de añadir esta línea al archivo de configuraciones se requiere aplicar el comando ***systemctl restart sshd***.
- La documentación de Ansible indica que para que los módulos sean reconocidos por la herramienta, así como los archivos de inventario, es necesario contar con las colecciones para configuraciones o despliegues incluidas. En este caso se requirió la descarga del paquete de *ansible-galaxy collections*, para que se reconozca el módulo de configuraciones para equipos Cisco.
- Al ejecutar el *playbook*, la documentación indica que por defecto las alertas están activadas, estas no indican un mal funcionamiento, pero dan en advertencia características útiles para el administrador de red, así mismo su documentación indica que es recomendable no desactivarlas.
- Al realizar la estructura del *playbook*, se consideró la ubicación de variables de uso general, para reconocimiento, conexión y optimización de las tareas. Se concluye que existen dos posibilidades, una en la cual se añaden las líneas *ansible_python_interpreter: /usr/bin/python* para establecer una ruta de conexión

con el host de destino, *ansible_connection: network_cli* para indicar a Ansible que la conexión se la realiza por la CLI de los equipos y *ansible_network_os: ios* para que reconozca el sistema operativo de los equipos en este caso Cisco. La segunda opción es que en el archivo de inventario se añadan las variables que todos los equipos usarán de forma común. A diferencia de la primera opción las líneas en lugar de dos puntos deben tener un igual y sin espacios por ejemplo *ansible_network_os=ios*.

- Ansible para la configuración de equipos de *networking* dentro del proyecto, potencia la eficiencia dentro del proceso de configuración de las redes, además también de ingresar datos según se requiera de forma interactiva. También ofrece la característica de cargar las configuraciones desde un archivo .txt, como se lo realizó para las configuraciones de DHCP; es decir, se puede usar casi toda la ejecución del comando ***show running-config***, y cargarlo con la herramienta Ansible.
- La configuración de equipos de *networking* mediante Ansible para la configuración de redes ayuda a las empresas a mejorar su producción, lo que genera un incremento del trabajo, en consecuencia, se podrá configurar en mayor cantidad, con mejores condiciones y reducir errores humanos.
- Se verificó la configuración automática de dos *routers* con los datos ingresados por parte del administrador de red de forma interactiva por medio de Ansible. La ejecución de este proyecto ha sido realizada de forma satisfactoria ya que se demostró mediante pruebas de conectividad, enviando pings, la comunicación entre PCs de diferentes LANs.

5 RECOMENDACIONES

- Se sugiere cambiar los elementos por defecto de GNS3 por elementos que cuenten con mejores características para que no existan fallos o inconsistencias que puedan generar errores.
- Considerando que este proceso de automatización en la configuración de las redes, puede ser parte de un proceso mucho más grande, se sugiere integrar a estos más procesos de configuración, según sea pertinente.
- Se debe tomar en cuenta que esta herramienta está en constante desarrollo y cada día se suman nuevas colecciones y documentación acerca de las

configuraciones de los equipos, existen varios enrutamientos que ya cuentan con una colección propia dentro de la herramienta Ansible, para el caso de enrutamientos que aún no cuentan con un módulo, se utiliza el módulo *lines* porque permite ingresar los comandos directamente como se lo ingresa en el equipo, aunque también es recomendable realizarlo desde un archivo `.txt`.

- Se recomienda utilizar medidas de seguridad del acceso a los archivos de configuraciones, por lo que una solución a este hueco de seguridad sería la creación de un usuario Ansible al cual tenga acceso exclusivo el administrador de red.
- Es recomendable no guardar las configuraciones del *playbook* con una tarea, sino realizarlo desde la *prompt* propia del equipo, con esto el administrador de red se asegura que las configuraciones fueron realizadas de manera satisfactoria y como respaldo a estas correctas configuraciones se ejecuta el comando ***wr***.
- La configuración de equipos de *networking* mediante Ansible abre el camino al desarrollo de más proyectos apoyados en la automatización, que sean capaces de adaptarse a modelos de administración de red, es recomendable utilizar los recursos brindados por las actualizaciones de Ansible para que esta adaptación se convierta en una de las herramientas más utilizadas por los administradores.

6 REFERENCIAS

- [1] Red Hat Enterprise Linux, «Conceptos básicos de Ansible,» Redhat.com, 2022. [En línea]. Available: <https://www.redhat.com/es/topics/automation/learning-ansible-tutorial>. [Último acceso: 23 Junio 2022].
- [2] L. Enciso y C. E. Morales-Iñiguez, «Ansible una estrategia de administración y configuración automatizada sobre GNS3 con OSPF para redes empresariales medianas,» *Revista Ibérica de Sistemas e Tecnologias de Informação*, pp. 239-252, 2021.
- [3] M. Zadca, Ansible. In: DevOps in Python, Berkeley, CA: Apress, 2019.
- [4] A. V. Yunga Toaquiza, «Escuela Superior Politécnica de Chimborazo,» Diciembre 2018. [En línea]. Available: <http://dspace.esPOCH.edu.ec/handle/123456789/10935>. [Último acceso: 10 Julio 2022].
- [5] L. a. R. M. Hochstein, Ansible: Up and Running: Automating configuration management and deployment the easy way, O'Reilly Media, Inc, 2017.
- [6] <https://www.raulprietofernandez.net/blog/gnu-linux/como-instalar-y-configurar-ansible-en-gnu-linux>, «BIBDIGITAL Escuela Politécnica Nacional,» Septiembre 2021. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/22224/1/CD%2011728.pdf>. [Último acceso: 14 Junio 2022].
- [7] R. P. Fernández, «raulprietofernandez.net,» Raúl Prieto Fernández, [En línea]. Available: <https://www.raulprietofernandez.net/blog/gnu-linux/como-instalar-y-configurar-ansible-en-gnu-linux>. [Último acceso: 1 Junio 2022].
- [8] K. John, «Computingforgeeks.com,» 26 Abril 2020. [En línea]. Available: <https://computingforgeeks.com/ubuntu-20-04-vs-ubuntu-18-04-comparison-table/>. [Último acceso: 21 Julio 2022].
- [9] P. Richard, Ubuntu 20.04 LTS Server: Administration and Reference. Surfing Turtle Press, 2020.

- [10] L. Jay, *Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04.*, Packt Publishing Ltd, 2018.
- [11] F. J. M. Bernal, «TRABAJO FINAL DE GRADO INFRAESTRUCTURA GNU/LINUX,» 2020. [En línea]. Available: <https://openaccess.uoc.edu/webapps/o2/bitstream/10609/73965/6/fmarquezbETFG0118memoria.pdf>. [Último acceso: 18 Julio 2022].
- [12] G. G. Z. V. D. K. V. V. L. A. Grynkevych G., «A METHOD OF PROACTIVE RESOURCE MANAGEMENT OF IOT DEVICES BASED ON A SERVERLESS ARCHITECTURE,» *Danish Scientific Journal*, vol. 1, nº 41, pp. 49-56, 2020.
- [13] «Enterprise Open Source and Linux | Ubuntu,» ubuntu, 2020. [En línea]. Available: <https://ubuntu.com/>. [Último acceso: 20 Junio 2022].
- [14] J. C. Neumann, *The book of GNS3: build virtual network labs using Cisco, Juniper, and more*, No Starch Press, 2015.
- [15] J. D. T. Y. J. C.-P. J. C. C. Manuel A. Calle, «Comparison of Parameters for an Appropriate Selection of Network Simulation Tools,» *Información tecnológica*, vol. 29, nº 6, 2018.
- [16] GNS3, «Getting Started with GNS3 | GNS3 Documentation,» Gns3.com, 2020. [En línea]. Available: <https://docs.gns3.com/docs/>. [Último acceso: 29 Julio 2022].
- [17] eve-ng, «Features Key Features,» Eve-ng.net, 2021. [En línea]. Available: <https://www.eve-ng.net/index.php/features/>. [Último acceso: 29 Julio 2022].
- [18] Cisco Packet Tracer, «Packet Tracer FAQs,» Networking Academy, 25 Marzo 2020. [En línea]. Available: <https://www.netacad.com/es/courses/packet-tracer/faq>. [Último acceso: 29 Julio 2022].

7 ANEXOS

ANEXO I: CERTIFICADO DE ORIGINALIDAD

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 2 de septiembre de 2022

De mi consideración:

Yo, GABRIELA KATHERINE CEVALLOS SALAZAR, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE ANSIBLE PARA CONFIGURAR EQUIPOS DE NETWORKING DE FORMA DINÁMICA elaborado por el estudiante FRANK JOHN VILLAGRÁN ANGULO de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

https://epnecuador-my.sharepoint.com/:f/g/personal/gabriela_cevalloss_epn_edu_ec/Eq0CYxFDlpVImI9MxCfEuasBSDd4ZaS2PpJVr1yCKf46EA?e=3eGcz3

Atentamente,



Gabriela Katherine Cevallos Salazar

Docente

Escuela de Formación de Tecnólogos

ANEXO III: VIDEO DE IMPLEMENTACIÓN Y PRUEBAS DE VERIFICACIÓN

El video demostrativo se lo puede localizar en <https://youtu.be/49cs4OyZJEw> o al escanear el siguiente código QR.



ANEXO III.I Código QR de enlace al video

ANEXO IIVI: CÓDIGO DE LOS PLAYBOOKS

Playbook 1

#Configuración de Router1 red nueva

- hosts: equipment1

vars_prompt:

- name: Hostname

prompt: Inserte hostname

private: no

- name: Ingreso_ipFastE

prompt: Inserte ip de FastEthernet 0/0 ejemplo = "172.168.56.123/16"

private: no

- name: Ingreso_ip_FastE

prompt: Coloque ip de FasteEthernet 0/1 ejemplo = "192.168.1.1/24"

private: no

- name: Ingreso_ip_serial

prompt: Coloque ip de Serial 0/0 ejemplo = "8.0.0.1/30"

private: no

- name: banner

prompt: Ingrese mensaje de banner login

private: no

- name: eigrp

prompt: Ingrese valor de sistema autónomo <1-65535>

private: no

- name: Ingreso_ipFastE_RED

prompt: Coloque ip de red FastEthernet 0/0 ejemplo = "172.168.0.0 0.0.255.255"

private: no

- name: Ingreso_ip_FastE_RED
prompt: Coloque ip de red FastEthernet 0/1 ejemplo = "192.168.1.0 0.0.0.255"
private: no

- name: Ingreso_ip_serial_RED
prompt: Coloque ip de red Serial 0/0 ejemplo = "8.0.0.0 0.0.0.3"
private: no

- name: psswd_console
prompt: Ingrese contraseña para console 0
private: no

- name: psswd_enable
prompt: Ingrese contraseña para enable
private: no

- name: psswd_line
prompt: Ingrese contraseña para line vty
private: no

tasks:

- name: Asignación de Hostname
cisco.ios.ios_hostname:
config:
 hostname: "{{ Hostname }}"
state: merged

- name: Asignación de ip a cada interfaz
cisco.ios.ios_l3_interfaces:
config:
 - name: FastEthernet0/0
 ipv4:
 - address: "{{ Ingreso_ipFastE }}"

```

- name: FastEthernet0/1
  ipv4:
    - address: "{{ Ingreso_ip_FastE }}"
- name: Serial0/0
  ipv4:
    - address: "{{ Ingreso_ip_serial }}"
state: merged

- name: Habilita interfaces
  cisco.ios.ios_interfaces:
    config:
      - name: FastEthernet0/0
        enabled: true
        description: Configured with Ansible "Designed by Frank J. Villagran A."
      - name: FastEthernet0/1
        enabled: true
        description: Configured with Ansible "Designed by Frank J. Villagran A."
      - name: Serial0/0
        enabled: true
        description: Configured with Ansible "Designed by Frank J. Villagran A."

- name: Mensaje de inicio
  cisco.ios.ios_banner:
    banner: login
    text: |
      {{ banner }}
    state: present

- name: Corrección de errores
  cisco.ios.ios_config:
    lines:
      - no router eigrp {{ eigrp }}
      - exit
    save_when: always

```

- name: Configuración de protocolo eigrp

cisco.ios.ios_config:

lines:

- router eigrp {{ eigrp }}
- no auto-summary
- network {{ Ingreso_ipFastE_RED }}
- network {{ Ingreso_ip_FastE_RED }}
- network {{ Ingreso_ip_serial_RED }}
- do wr

save_when: always

- name: Configuración contraseña console

cisco.ios.ios_config:

lines:

- line console 0
- password {{ pswd_console }}
- login
- do wr

save_when: always

- name: Configuración contraseña enable

cisco.ios.ios_config:

lines:

- enable password {{ pswd_enable }}
- do wr

save_when: always

- name: Configuración contraseña line vty

cisco.ios.ios_config:

lines:

- line vty 1 4
- password {{ pswd_line }}
- login local
- do wr

save_when: always

- name: Asignación de pool de direcciones para DHCP

cisco.ios.ios_config:

src: /etc/ansible/config_DHCP1.txt

- name: Guardar cambios

cisco.ios.ios_config:

lines:

- do wr

save_when: always

Playbook 2

#Configuración de Router1 red nueva

- hosts: equipment2

vars_prompt:

- name: Hostname

prompt: Inserte hostname

private: no

- name: Ingreso_ip_FastE

prompt: Coloque ip de FasteEthernet 0/1 ejemplo = "192.168.0.1/24"

private: no

- name: Ingreso_ip_serial

prompt: Coloque ip de Serial 0/0 ejemplo = "8.0.0.2/30"

private: no

- name: cl_rate

prompt: Coloque velocidad de clock rate ejemplo = "64000"

private: no

- name: banner

prompt: Ingrese mensaje de banner login

private: no

- name: eigrp

prompt: Ingrese valor de sistema autónomo <1-65535>

private: no

- name: Ingreso_ip_FastE_RED

prompt: Coloque ip de red FastEthernet 0/1 ejemplo = "192.168.0.0 0.0.0.255"

private: no

- name: Ingreso_ip_serial_RED

prompt: Coloque ip de red Serial 0/0 ejemplo = "8.0.0.0 0.0.0.3"

private: no

- name: psswd_console

prompt: Ingrese contraseña para console 0

private: no

- name: psswd_enable

prompt: Ingrese contraseña para enable

private: no

- name: psswd_line

prompt: Ingrese contraseña para line vty

private: no

tasks:

- name: Asignación de Hostname

cisco.ios.ios_hostname:

config:

hostname: "{{ Hostname }}"

state: merged

- name: Asignación de ip a cada interfaz

cisco.ios.ios_l3_interfaces:

config:

- name: FastEthernet0/1

ipv4:

- address: "{{ Ingreso_ip_FastE }}"

- name: Serial0/0

ipv4:

- address: "{{ Ingreso_ip_serial }}"

state: merged

- name: Asignación velocidad de clock rate

cisco.ios.ios_config:

lines:

- interface serial 0/0

- clock rate {{ cl_rate }}

- do wr

save_when: always

- name: Habilita interfaces

cisco.ios.ios_interfaces:

config:

- name: FastEthernet0/1

enabled: true

description: Configured with Ansible "Designed by Frank J. Villagrán A."

- name: Serial0/0

enabled: true

description: Configured with Ansible "Designed by Frank J. Villagrán A."

- name: Mensaje de inicio

cisco.ios.ios_banner:

banner: login

text: |

{{ banner }}

state: present

- name: Corrección de errores

cisco.ios.ios_config:

lines:

- no router eigrp {{ eigrp }}

- exit

save_when: always

- name: Configuración de protocolo eigrp

cisco.ios.ios_config:

lines:

- router eigrp {{ eigrp }}

- network {{ Ingreso_ip_FastE_RED }}

- network {{ Ingreso_ip_serial_RED }}

- do wr

save_when: always

- name: Configuración contraseña console

cisco.ios.ios_config:

lines:

- line console 0

- password {{ pswd_console }}

- login

- do wr

save_when: always

- name: Configuración contraseña enable

cisco.ios.ios_config:

lines:

- enable password {{ pswd_enable }}

- do wr

save_when: always

- name: Configuración contraseña line vty

cisco.ios.ios_config:

lines:

- line vty 1 4
- password {{ psswd_line }}
- login local
- do wr

save_when: always

- name: Asignación de pool de direcciones para DHCP

cisco.ios.ios_config:

src: /etc/ansible/config_DHCP2.txt

- name: Guardar cambios

cisco.ios.ios_config:

lines:

- do wr

save_when: always