



Instituto Politécnico Nacional

Escuela Superior de Computo.

Adolfo Armando Villegas Jiménez

4CM14 Ing. en sistemas computacionales

UA: Administración de servicios en red

Profeso: Fabián Gaspar Medina

Proyecto Tercer Parcial

Reporte

Reporte de práctica

Objetivo

Realizar una interconexión de redes mediante enrutamiento estático y dinámico.

Requerimientos

- Una computadora personal
- Software para simulación de redes (GNS3).
- Python
- Librerías para el desarrollo del programa:
 - Gi
 - Pexpect
 - Paramiko
- IDE Glade

Introducción

Durante el curso se vio sobre los elementos para implementar una topología de red, así sobre el cómo realizar la configuración y buen funcionamiento de la topología. Entre lo que llamo la atención respecto a lo visto en el curso es el realizar consultar a cada uno de los servidores y extraer la información requerida por el usuario sin tener que interactuar directamente con los servidores.

Para poner en práctica los conocimientos vistos en el curso, se realizó un proyecto en el que desde una máquina virtual, y con ayuda del lenguaje de programación python y algunas librerías, entables la comunicación entre los servidores con el fin de efectuar consultas sin interactuar con los servidores. De los temas vistos en clase se realizó la implementación de una topología que contara con enrutamiento dinámico, listas de control de acceso, NAT en algunos servidores, así como el mínimo mencionado en la rúbrica sobre los router y VPC que debe contar la topología. Uno de los elementos principales con los que debe contar la topología es la máquina virtual, debido a que desde la maquina se ejecutará el programa y realizará las consultas requeridas en la rúbrica.

Telnet

El protocolo telnet proporciona un método estándar para que los dispositivos de terminal y los procesos orientados a terminal intercambien información. La desventaja que se encuentra en el protocolo es la falta de seguridad, por ejemplo, al obtener las configuraciones que realizamos y las contraseñas que introducimos al dispositivo.

SSH

Es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor, que a diferencia de telnet, los datos que se intercambian entre los dispositivos, se encuentran encriptados. Eso quiere decir que nosotros no podemos leer los datos encapsulados en las tramas como lo hacíamos con Telnet.

pexpect

Es un módulo de Python para la creación, control y respuesta de aplicaciones secundarias para el manejo de patrones. Pexpect permite crear guiones para interactuar como si un humano estuviera escribiendo.

paramiko

Es una implementación de Python para SSHv2, por lo que no tiene soporte para Telnet y se enfoca a operaciones cliente servidor.

Desarrollo

Topología

Sobre la topología a armar, es el siguiente:

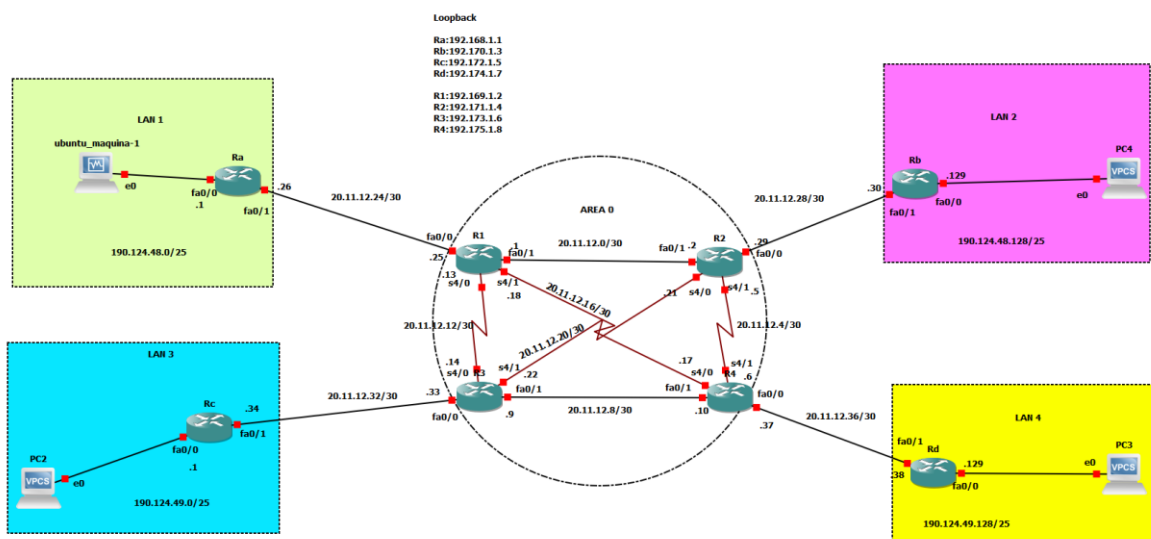


Imagen 1. Topología de la red.

Como se puede visualizar en la imagen, en la LAN 1 estará posicionado la máquina virtual que es donde ejecutaremos el programa.

Configuración

En base a la topología, se configurarán las interfaces de los routers, aparte de que se implementarán loopback en cada router para las consultas que vaya a realizar el programa.

A continuación se muestran las configuraciones de interfaces de los routers:

```
Ra#enable
Ra#config terminal
Ra(config)# interface Loopback0
Ra(config-if)# description loopback
Ra(config-if)# ip address 192.168.1.1 255.255.255.255
Ra(config-if)#no shutdown
```

```
Ra(config-if)#exit
```

```
Ra#enable
```

```
Ra#config terminal
```

```
Ra(config)#interface FastEthernet 0/0
```

```
Ra(config-if)# ip address 190.124.48.1 255.255.255.128
```

```
Ra(config-if)#no shutdown
```

```
Ra(config-if)#exit
```

```
Ra#enable
```

```
Ra#config terminal
```

```
Ra(config)#interface FastEthernet 0/1
```

```
Ra(config-if)# ip address 20.11.12.26 255.255.255.252
```

```
Ra(config-if)#no shutdown
```

```
Ra(config-if)#exit
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	190.124.48.1	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.26	YES	NVRAM	up	up
Loopback0	192.168.1.1	YES	NVRAM	up	up

Imagen 2. Tabla de las interfaces del router Ra.

Una vez configurado el router R1 se procede a configurar el router R2, se configurarán los puertos de enlace.

```
Rb#enable
```

```
Rb#config terminal
```

```
Rb(config)# interface Loopback0
```

```
Rb(config-if)#description loopback
```

```
Rb(config-if)# ip address 192.170.1.3 255.255.255.255
```

```
Rb(config-if)#no shutdown
```

```
Rb(config-if)#exit
```

```
Rb#enable
```

```
Rb#config terminal
```

```
Rb(config)#interface fastethernet 0/0
```

```
Rb(config-if)# ip address 190.124.48.129 255.255.255.128
```

```
Rb(config-if)#no shutdown
```

```
Rb(config-if)#exit
```

```
Rb#enable
```

```
Rb#config terminal
```

```
Rb(config)#interface fastethernet 0/1
```

```
Rb(config-if)# ip address 20.11.12.30 255.255.255.252
```

```
Rb(config-if)#no shutdown
```

```
Rb(config-if)#exit
```

```
Rb#sh ip int br | exclude unas
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	190.124.48.129	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.30	YES	NVRAM	up	up
Loopback0	192.170.1.3	YES	NVRAM	up	up

Imagen 3. Tabla de las interfaces del router Rb.

```
Rc#enable
Rc#config terminal
Rc(config)# interface Loopback0
Rc(config-if)#description loopback
Rc(config-if)# ip address 192.172.1.5 255.255.255.255
Rc(config-if)#no shutdown
Rc(config-if)#exit
```

```
Rc#enable
Rc#config terminal
Rc(config)#interface fastethernet 0/0
Rc(config-if)# ip address 190.124.49.1 255.255.255.128
Rc(config-if)#no shutdown
Rc(config-if)#exit
```

```
Rc#enable
Rc#config terminal
Rc(config)#interface fastethernet 0/1
Rc(config-if)# ip address 20.11.12.34 255.255.255.252
Rc(config-if)#no shutdown
Rc(config-if)#exit
```

```
Rc#sh ip int br | exclude unas
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	190.124.49.1	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.34	YES	NVRAM	up	up
Loopback0	192.172.1.5	YES	NVRAM	up	up

Imagen 4. Tabla de las interfaces del router Rc.

```
Rd#enable
Rd#config terminal
Rd(config)# interface Loopback0
Rd(config-if)#description loopback
Rd(config-if)# ip address 192.174.1.7 255.255.255.255
Rd(config-if)#no shutdown
```

Rd(config-if)#exit

Rd#enable

Rd#config terminal

Rd(config)#interface fastethernet 0/0

Rd(config-if)# ip address 190.124.49.129 255.255.255.128

Rd(config-if)#no shutdown

Rd(config-if)#exit

Rd#enable

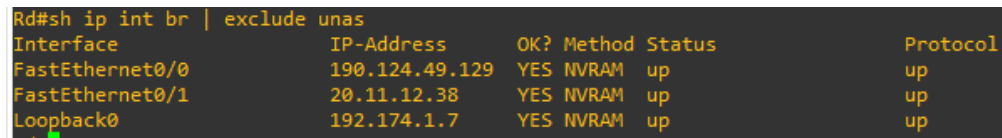
Rd#config terminal

Rd(config)#interface fastethernet 0/1

Rd(config-if)# ip address 20.11.12.38 255.255.255.252

Rd(config-if)#no shutdown

Rd(config-if)#exit



Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	190.124.49.129	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.38	YES	NVRAM	up	up
Loopback0	192.174.1.7	YES	NVRAM	up	up

Imagen 5. Tabla de las interfaces del router Rd.

R1#enable

R1#config terminal

R1(config)# interface Loopback0

R1(config-if)#description loopback

R1(config-if)# ip address 192.169.1.2 255.255.255.255

R1(config-if)#no shutdown

R1(config-if)#exit

R1#enable

R1#config terminal

R1(config)#interface fastethernet 0/0

R1(config-if)# ip address 20.11.12.25 255.255.255.252

R1(config-if)#no shutdown

R1(config-if)#exit

R1#enable

R1#config terminal

R1(config)#interface fastethernet 0/1

R1(config-if)# ip address 20.11.12.1 255.255.255.252

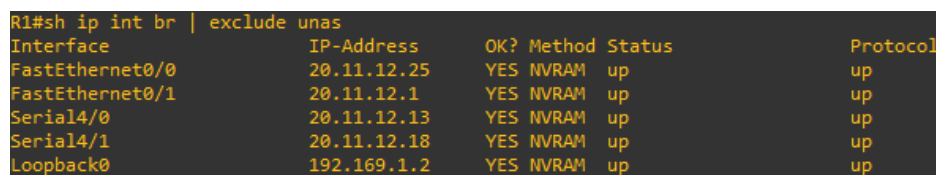
R1(config-if)#no shutdown

R1(config-if)#exit

R1#enable

```
R1#config terminal
R1(config)#interface Serial4/0
R1(config-if)# ip address 20.11.12.13 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit
```

```
R1#enable
R1#config terminal
R1(config)#interface Serial4/1
R1(config-if)# ip address 20.11.12.18 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit
```



Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	20.11.12.25	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.1	YES	NVRAM	up	up
Serial4/0	20.11.12.13	YES	NVRAM	up	up
Serial4/1	20.11.12.18	YES	NVRAM	up	up
Loopback0	192.169.1.2	YES	NVRAM	up	up

Imagen 6. Tabla de las interfaces del router R1.

```
R2#enable
R2#config terminal
R2(config)# interface Loopback0
R2(config-if)#description loopback
R2(config-if)# ip address 192.171.1.4 255.255.255.255
R2(config-if)#no shutdown
R2(config-if)#exit
```

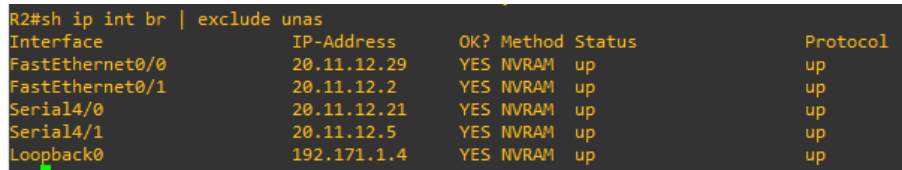
```
R2#enable
R2#config terminal
R2(config)#interface fastethernet 0/0
R2(config-if)# ip address 20.11.12.29 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
```

```
R2#enable
R2#config terminal
R2(config)#interface fastethernet 0/1
R2(config-if)# ip address 20.11.12.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
```

```
R2#enable
R2#config terminal
R2(config)#interface Serial4/0
R2(config-if)# ip address 20.11.12.21 255.255.255.252
```

```
R2(config-if)#no shutdown
R2(config-if)#exit
```

```
R2#enable
R2#config terminal
R2(config)#interface Serial4/1
R2(config-if)# ip address 20.11.12.5 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
```



Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	20.11.12.29	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.2	YES	NVRAM	up	up
Serial4/0	20.11.12.21	YES	NVRAM	up	up
Serial4/1	20.11.12.5	YES	NVRAM	up	up
Loopback0	192.171.1.4	YES	NVRAM	up	up

Imagen 7. Tabla de las interfaces del router R2.

```
R3#enable
R3#config terminal
R3(config)# interface Loopback0
R3(config-if)#description loopback
R3(config-if)# ip address 192.173.1.6 255.255.255.255
R3(config-if)#no shutdown
R3(config-if)#exit
```

```
R3#enable
R3#config terminal
R3(config)#interface fastethernet 0/0
R3(config-if)# ip address 20.11.12.33 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit
```

```
R3#enable
R3#config terminal
R3(config)#interface fastethernet 0/1
R3(config-if)# ip address 20.11.12.9 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit
```

```
R3#enable
R3#config terminal
R3(config)#interface Serial4/0
R3(config-if)# ip address 20.11.12.14 255.255.255.252
R3(config-if)#no shutdown
```



```
R3(config-if)#exit
```

```
R3#enable
```

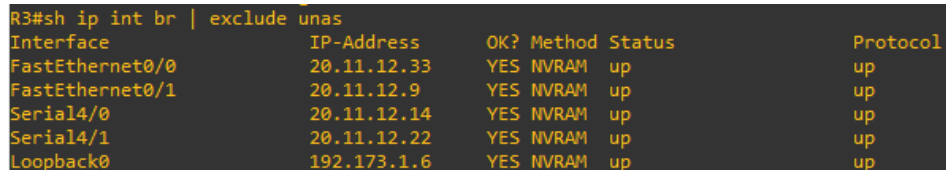
```
R3#config terminal
```

```
R3(config)#interface Serial4/1
```

```
R3(config-if)# ip address 20.11.12.22 255.255.255.252
```

```
R3(config-if)#no shutdown
```

```
R3(config-if)#exit
```



Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	20.11.12.33	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.9	YES	NVRAM	up	up
Serial4/0	20.11.12.14	YES	NVRAM	up	up
Serial4/1	20.11.12.22	YES	NVRAM	up	up
Loopback0	192.173.1.6	YES	NVRAM	up	up

Imagen 8. Tabla de las interfaces del router R3.

```
R4#enable
```

```
R4#config terminal
```

```
R4(config)# interface Loopback0
```

```
R4(config-if)#description loopback
```

```
R4(config-if)# ip address 192.175.1.8 255.255.255.255
```

```
R4(config-if)#no shutdown
```

```
R4(config-if)#exit
```

```
R4#enable
```

```
R4#config terminal
```

```
R4(config)#interface fastethernet 0/0
```

```
R4(config-if)# ip address 20.11.12.37 255.255.255.252
```

```
R4(config-if)#no shutdown
```

```
R4(config-if)#exit
```

```
R4#enable
```

```
R4#config terminal
```

```
R4(config)#interface fastethernet 0/1
```

```
R4(config-if)# ip address 20.11.12.10 255.255.255.252
```

```
R4(config-if)#no shutdown
```

```
R4(config-if)#exit
```

```
R4#enable
```

```
R4#config terminal
```

```
R4(config)#interface Serial4/0
```

```
R4(config-if)# ip address 20.11.12.17 255.255.255.252
```

```
R4(config-if)#no shutdown
```

```
R4(config-if)#exit
```

```

R4#enable
R4#config terminal
R4(config)#interface Serial4/1
R4(config-if)# ip address 20.11.12.6 255.255.255.252
R4(config-if)#no shutdown
R4(config-if)#exit

```

```
R4#sh ip int br | exclude unas
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	20.11.12.37	YES	NVRAM	up	up
FastEthernet0/1	20.11.12.10	YES	NVRAM	up	up
Serial4/0	20.11.12.17	YES	NVRAM	up	up
Serial4/1	20.11.12.6	YES	NVRAM	up	up
Loopback0	192.175.1.8	YES	NVRAM	up	up

Imagen 9. Tabla de las interfaces del router R4.

Enrutamiento

A continuación, se realizará el enrutamiento de cada router. El enrutamiento a aplicar será el enrutamiento dinámico, haciendo uso del protocolo OSPF, con las LAN correspondiendo a las áreas:

```

Ra(config)# router ospf 1
Ra(config-router)# log-adjacency-changes
Ra(config-router)# passive-interface Loopback0
Ra(config-router)# network 20.11.12.24 0.0.0.3 area 0
Ra(config-router)# network 190.124.48.0 0.0.0.127 area 1
Ra(config-router)#exit

```

```

Gateway of last resort is not set

  20.0.0.0/30 is subnetted, 10 subnets
O   20.11.12.16 [110/65] via 20.11.12.25, 02:00:31, FastEthernet0/1
O   20.11.12.20 [110/66] via 20.11.12.25, 02:00:31, FastEthernet0/1
C   20.11.12.24 is directly connected, FastEthernet0/1
O   20.11.12.28 [110/3] via 20.11.12.25, 02:00:31, FastEthernet0/1
O   20.11.12.0 [110/2] via 20.11.12.25, 02:00:31, FastEthernet0/1
O   20.11.12.4 [110/66] via 20.11.12.25, 02:00:31, FastEthernet0/1
O   20.11.12.8 [110/66] via 20.11.12.25, 02:00:31, FastEthernet0/1
O   20.11.12.12 [110/65] via 20.11.12.25, 02:00:33, FastEthernet0/1
O   20.11.12.32 [110/66] via 20.11.12.25, 02:00:33, FastEthernet0/1
O   20.11.12.36 [110/66] via 20.11.12.25, 02:00:33, FastEthernet0/1
  190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/4] via 20.11.12.25, 02:00:33, FastEthernet0/1
O IA 190.124.49.128 [110/67] via 20.11.12.25, 02:00:35, FastEthernet0/1
C   190.124.48.0 is directly connected, FastEthernet0/0
O IA 190.124.49.0 [110/67] via 20.11.12.25, 02:00:36, FastEthernet0/1
  192.168.1.0/32 is subnetted, 1 subnets
C   192.168.1.1 is directly connected, Loopback0
Ra#

```

Imagen 10. Tabla de enrutamiento del router Ra.

```

Rb(config)# router ospf 1
Rb(config-router)# log-adjacency-changes
Rb(config-router)# passive-interface Loopback0
Rb(config-router)# network 20.11.12.28 0.0.0.3 area 0
Rb(config-router)# network 190.124.48.128 0.0.0.127 area 2
Rb(config-router)#exit

```

```

Gateway of last resort is not set

    20.0.0.0/30 is subnetted, 10 subnets
O       20.11.12.16 [110/66] via 20.11.12.29, 02:01:00, FastEthernet0/1
O       20.11.12.20 [110/65] via 20.11.12.29, 02:01:00, FastEthernet0/1
O       20.11.12.24 [110/3] via 20.11.12.29, 02:01:00, FastEthernet0/1
C       20.11.12.28 is directly connected, FastEthernet0/1
O       20.11.12.0 [110/2] via 20.11.12.29, 02:01:00, FastEthernet0/1
O       20.11.12.4 [110/65] via 20.11.12.29, 02:01:00, FastEthernet0/1
O       20.11.12.8 [110/66] via 20.11.12.29, 02:01:02, FastEthernet0/1
O       20.11.12.12 [110/66] via 20.11.12.29, 02:01:02, FastEthernet0/1
O       20.11.12.32 [110/66] via 20.11.12.29, 02:01:02, FastEthernet0/1
O       20.11.12.36 [110/66] via 20.11.12.29, 02:01:02, FastEthernet0/1
    190.124.0.0/25 is subnetted, 4 subnets
C       190.124.48.128 is directly connected, FastEthernet0/0
O IA    190.124.49.128 [110/67] via 20.11.12.29, 02:01:03, FastEthernet0/1
O IA    190.124.48.0 [110/4] via 20.11.12.29, 02:01:03, FastEthernet0/1
O IA    190.124.49.0 [110/67] via 20.11.12.29, 02:01:03, FastEthernet0/1
    192.170.1.0/32 is subnetted, 1 subnets
C       192.170.1.3 is directly connected, Loopback0
Rb#

```

Imagen 11. Tabla de enrutamiento del router Rb.

```

Rc(config)# router ospf 1
Rc(config-router)# log-adjacency-changes
Rc(config-router)# passive-interface Loopback0
Rc(config-router)# network 20.11.12.32 0.0.0.3 area 0
Rc(config-router)# network 190.124.49.0 0.0.0.127 area 3
Rc(config-router)#exit

```

```

Gateway of last resort is not set

    20.0.0.0/30 is subnetted, 10 subnets
O       20.11.12.16 [110/66] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.20 [110/65] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.24 [110/66] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.28 [110/66] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.0 [110/66] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.4 [110/66] via 20.11.12.33, 02:01:10, FastEthernet0/1
O       20.11.12.8 [110/2] via 20.11.12.33, 02:01:12, FastEthernet0/1
O       20.11.12.12 [110/65] via 20.11.12.33, 02:01:12, FastEthernet0/1
C       20.11.12.32 is directly connected, FastEthernet0/1
O       20.11.12.36 [110/3] via 20.11.12.33, 02:01:12, FastEthernet0/1
    190.124.0.0/25 is subnetted, 4 subnets
O IA    190.124.48.128 [110/67] via 20.11.12.33, 02:01:12, FastEthernet0/1
O IA    190.124.49.128 [110/4] via 20.11.12.33, 02:01:12, FastEthernet0/1
O IA    190.124.48.0 [110/67] via 20.11.12.33, 02:01:12, FastEthernet0/1
C       190.124.49.0 is directly connected, FastEthernet0/0
    192.172.1.0/32 is subnetted, 1 subnets
C       192.172.1.5 is directly connected, Loopback0
Rc#

```

Imagen 12. Tabla de enrutamiento del router Rc.

```

Rd(config)# router ospf 1
Rd(config-router)# log-adjacency-changes
Rd(config-router)# passive-interface Loopback0
Rd(config-router)# network 20.11.12.36 0.0.0.3 area 0
Rd(config-router)# network 190.124.49.128 0.0.0.127 area 4
Rd(config-router)#exit

```

```

Gateway of last resort is not set

20.0.0.0/30 is subnetted, 10 subnets
O   20.11.12.16 [110/65] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.20 [110/66] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.24 [110/66] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.28 [110/66] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.0 [110/66] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.4 [110/65] via 20.11.12.37, 02:01:20, FastEthernet0/1
O   20.11.12.8 [110/2] via 20.11.12.37, 02:01:22, FastEthernet0/1
O   20.11.12.12 [110/66] via 20.11.12.37, 02:01:22, FastEthernet0/1
O   20.11.12.32 [110/3] via 20.11.12.37, 02:01:22, FastEthernet0/1
C   20.11.12.36 is directly connected, FastEthernet0/1
190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/67] via 20.11.12.37, 02:01:22, FastEthernet0/1
C   190.124.49.128 is directly connected, FastEthernet0/0
O IA 190.124.48.0 [110/67] via 20.11.12.37, 02:01:22, FastEthernet0/1
O IA 190.124.49.0 [110/4] via 20.11.12.37, 02:01:22, FastEthernet0/1
192.174.1.0/32 is subnetted, 1 subnets
C   192.174.1.7 is directly connected, Loopback0
Rd#

```

Imagen 13. Tabla de enrutamiento del router Rd.

```

R1(config)# router ospf 1
R1(config-router)# log-adjacency-changes
R1(config-router)# passive-interface Loopback0
R1(config-router)# network 20.11.12.0 0.0.0.3 area 0
R1(config-router)# network 20.11.12.12 0.0.0.3 area 0
R1(config-router)# network 20.11.12.16 0.0.0.3 area 0
R1(config-router)# network 20.11.12.24 0.0.0.3 area 0
R1(config-router)#exit

```

```

Gateway of last resort is not set

20.0.0.0/30 is subnetted, 10 subnets
C   20.11.12.16 is directly connected, Serial4/1
O   20.11.12.20 [110/65] via 20.11.12.2, 02:01:27, FastEthernet0/1
C   20.11.12.24 is directly connected, FastEthernet0/0
O   20.11.12.28 [110/2] via 20.11.12.2, 02:01:27, FastEthernet0/1
C   20.11.12.0 is directly connected, FastEthernet0/1
O   20.11.12.4 [110/65] via 20.11.12.2, 02:01:27, FastEthernet0/1
O   20.11.12.8 [110/65] via 20.11.12.17, 02:01:27, Serial4/1
    [110/65] via 20.11.12.14, 02:01:29, Serial4/0
C   20.11.12.12 is directly connected, Serial4/0
O   20.11.12.32 [110/65] via 20.11.12.14, 02:01:29, Serial4/0
O   20.11.12.36 [110/65] via 20.11.12.17, 02:01:29, Serial4/1
190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/3] via 20.11.12.2, 02:01:31, FastEthernet0/1
O IA 190.124.49.128 [110/66] via 20.11.12.17, 02:01:31, Serial4/1
O IA 190.124.48.0 [110/2] via 20.11.12.26, 02:01:31, FastEthernet0/0
O IA 190.124.49.0 [110/66] via 20.11.12.14, 02:01:31, Serial4/0
192.169.1.0/32 is subnetted, 1 subnets
C   192.169.1.2 is directly connected, Loopback0
R1#

```

Imagen 14. Tabla de enrutamiento del router R1.

```
R2(config)# router ospf 1
R2(config-router)# log-adjacency-changes
R2(config-router)# passive-interface Loopback0
R2(config-router)# network 20.11.12.0 0.0.0.3 area 0
R2(config-router)# network 20.11.12.4 0.0.0.3 area 0
R2(config-router)# network 20.11.12.20 0.0.0.3 area 0
R2(config-router)# network 20.11.12.28 0.0.0.3 area 0
R2(config-router)#exit
```

```
Gateway of last resort is not set

 20.0.0.0/30 is subnetted, 10 subnets
O   20.11.12.16 [110/65] via 20.11.12.1, 02:01:37, FastEthernet0/1
C   20.11.12.20 is directly connected, Serial4/0
O   20.11.12.24 [110/2] via 20.11.12.1, 02:01:37, FastEthernet0/1
C   20.11.12.28 is directly connected, FastEthernet0/0
C   20.11.12.0 is directly connected, FastEthernet0/1
C   20.11.12.4 is directly connected, Serial4/1
O   20.11.12.8 [110/65] via 20.11.12.22, 02:01:37, Serial4/0
    [110/65] via 20.11.12.6, 02:01:37, Serial4/1
O   20.11.12.12 [110/65] via 20.11.12.1, 02:01:39, FastEthernet0/1
O   20.11.12.32 [110/65] via 20.11.12.22, 02:01:39, Serial4/0
O   20.11.12.36 [110/65] via 20.11.12.6, 02:01:39, Serial4/1
 190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/2] via 20.11.12.30, 02:01:39, FastEthernet0/0
O IA 190.124.49.128 [110/66] via 20.11.12.6, 02:01:39, Serial4/1
O IA 190.124.48.0 [110/3] via 20.11.12.1, 02:01:39, FastEthernet0/1
O IA 190.124.49.0 [110/66] via 20.11.12.22, 02:01:39, Serial4/0
 192.171.1.0/32 is subnetted, 1 subnets
C   192.171.1.4 is directly connected, Loopback0
R2#
```

Imagen 15. Tabla de enrutamiento del router R2.

```
R3(config)# router ospf 1
R3(config-router)# log-adjacency-changes
R3(config-router)# passive-interface Loopback0
R3(config-router)# network 20.11.12.8 0.0.0.3 area 0
R3(config-router)# network 20.11.12.12 0.0.0.3 area 0
R3(config-router)# network 20.11.12.20 0.0.0.3 area 0
R3(config-router)# network 20.11.12.32 0.0.0.3 area 0
R3(config-router)#exit
```

```

Gateway of last resort is not set

    20.0.0.0/30 is subnetted, 10 subnets
O   20.11.12.16 [110/65] via 20.11.12.10, 02:01:45, FastEthernet0/1
C   20.11.12.20 is directly connected, Serial4/1
O   20.11.12.24 [110/65] via 20.11.12.13, 02:01:45, Serial4/0
O   20.11.12.28 [110/65] via 20.11.12.21, 02:01:45, Serial4/1
O   20.11.12.0 [110/65] via 20.11.12.21, 02:01:45, Serial4/1
    [110/65] via 20.11.12.13, 02:01:45, Serial4/0
O   20.11.12.4 [110/65] via 20.11.12.10, 02:01:45, FastEthernet0/1
C   20.11.12.8 is directly connected, FastEthernet0/1
C   20.11.12.12 is directly connected, Serial4/0
C   20.11.12.32 is directly connected, FastEthernet0/0
O   20.11.12.36 [110/2] via 20.11.12.10, 02:01:47, FastEthernet0/1
    190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/66] via 20.11.12.21, 02:01:47, Serial4/1
O IA 190.124.49.128 [110/3] via 20.11.12.10, 02:01:47, FastEthernet0/1
O IA 190.124.48.0 [110/66] via 20.11.12.13, 02:01:47, Serial4/0
O IA 190.124.49.0 [110/2] via 20.11.12.34, 02:01:47, FastEthernet0/0
    192.173.1.0/32 is subnetted, 1 subnets
C   192.173.1.6 is directly connected, Loopback0
R3#

```

Imagen 16. Tabla de enrutamiento del router R3.

```

R4(config)# router ospf 1
R4(config-router)# log-adjacency-changes
R4(config-router)# passive-interface Loopback0
R4(config-router)# network 20.11.12.4 0.0.0.3 area 0
R4(config-router)# network 20.11.12.8 0.0.0.3 area 0
R4(config-router)# network 20.11.12.16 0.0.0.3 area 0
R4(config-router)# network 20.11.12.36 0.0.0.3 area 0
R4(config-router)#exit

```

```

Gateway of last resort is not set

    20.0.0.0/30 is subnetted, 10 subnets
C   20.11.12.16 is directly connected, Serial4/0
O   20.11.12.20 [110/65] via 20.11.12.9, 02:01:53, FastEthernet0/1
O   20.11.12.24 [110/65] via 20.11.12.18, 02:01:53, Serial4/0
O   20.11.12.28 [110/65] via 20.11.12.5, 02:01:53, Serial4/1
O   20.11.12.0 [110/65] via 20.11.12.18, 02:01:53, Serial4/0
    [110/65] via 20.11.12.5, 02:01:53, Serial4/1
C   20.11.12.4 is directly connected, Serial4/1
C   20.11.12.8 is directly connected, FastEthernet0/1
O   20.11.12.12 [110/65] via 20.11.12.9, 02:01:55, FastEthernet0/1
O   20.11.12.32 [110/2] via 20.11.12.9, 02:01:55, FastEthernet0/1
C   20.11.12.36 is directly connected, FastEthernet0/0
    190.124.0.0/25 is subnetted, 4 subnets
O IA 190.124.48.128 [110/66] via 20.11.12.5, 02:01:55, Serial4/1
O IA 190.124.49.128 [110/2] via 20.11.12.38, 02:01:55, FastEthernet0/0
O IA 190.124.48.0 [110/66] via 20.11.12.18, 02:01:55, Serial4/0
O IA 190.124.49.0 [110/3] via 20.11.12.9, 02:01:55, FastEthernet0/1
    192.175.1.0/32 is subnetted, 1 subnets
C   192.175.1.8 is directly connected, Loopback0
R4#

```

Imagen 17. Tabla de enrutamiento del router R4.

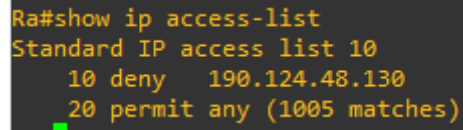
Configuración de las listas de acceso

Las listas de acceso a implementar serán del tipo estándar, con la especificación de denegar el acceso al host correspondiente, esto se implementará en los routers Ra y Rd.

```

Ra(config)# ip access-list standard 10
Ra(conf-std-nacl)# deny host 190.124.48.130
Ra(conf-std-nacl)# permit any
Ra(conf-std-nacl)# exit
Ra(config)# interface FastEthernet 0/1
Ra(config-if)# ip access-group 10 in

```



```

Ra#show ip access-list
Standard IP access list 10
  10 deny    190.124.48.130
  20 permit any (1005 matches)

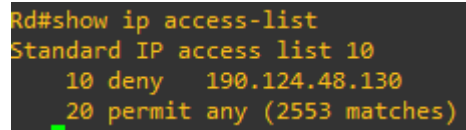
```

Imagen 18. Lista de acceso del router R4.

```

Rd(config)# ip access-list standard 10
Rd(conf-std-nacl)# deny host 190.124.48.130
Rd(conf-std-nacl)# permit any
Rd(conf-std-nacl)# exit
Rd(config)# interface FastEthernet 0/1
Rd(config-if)# ip access-group 10 in

```



```

Rd#show ip access-list
Standard IP access list 10
  10 deny    190.124.48.130
  20 permit any (2553 matches)

```

Imagen 19. Lista de acceso del router R4.

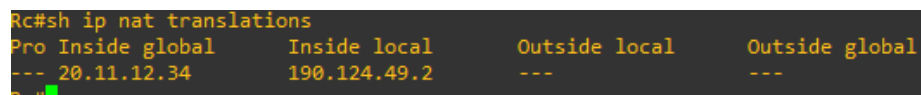
Configuración de las NAT

Para lo especificado en el proyecto, se implementarán NAT estática en el router Rc y Rd, especificado las interfaces de entrada y la de salida.

```

Rc# conf t
Rc(config)# int f 0/0
Rc(config-if)# ip nat inside
Rc(config-if)# int f 0/1
Rc(config-if)# ip nat outside
Rc(config-if)# end
Rc#conf t
Rc(config)# ip nat inside source static 190.124.49.2 20.11.12.34

```



```

Rc#sh ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 20.11.12.34        190.124.49.2      ---                ---

```

Imagen 20. Tabla de traducciones NAT de Rc.

```

Rd# conf t
Rd(config)# int f 0/0
Rd(config-if)# ip nat inside
Rd(config-if)# int f 0/1
Rd(config-if)# ip nat outside
Rd(config-if)# end
Rd#conf t
Rd(config)# ip nat inside source static 190.124.49.130 20.11.12.38

```

```

Rd#sh ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 20.11.12.38        190.124.49.130   ---                ---

```

Imagen 21. Tabla de traducciones NAT Rd.

Configuración del protocolo SSH y Telnet

Para poder hacer uso correctamente de los protocolos Telnet y SSH, que serán utilizados en el programa, se introducirán las siguientes líneas de comando, especificando el nombre del dominio, establecer el protocolo SSH, la generación de llaver del tipo RSA, el usuario y contraseña del servidor. Las líneas de comando se replicarán a todos los routers.

```

ip domain-name adminredes.escom.ipn.mx
ip ssh rsa keypair-name sshkey
crypto key generate rsa usage-keys label sshkey modulus 1024
ip ssh v 2
ip ssh time-out 30
ip ssh authentication-retries 3
line vty 0 15
password cisco
login local
transport input ssh telnet
exit
username cisco privilege 15 password cisco
end

```

Algo que cabe resaltar es el siguiente comando. Dicho comando se tendrá que ingresar cada que se apaguen los router o se cierre el programa.

crypto key generate rsa usage-keys label sshkey modulus 1024

Configuración de las VPC

Después de configurar los routers, se encienden las vpcs, y se configuran asignándole los hosts correspondientes:

PC1

La PC, correspondiente a la máquina virtual, se configurará dentro del SO, por lo que tendremos que asignarlo como Manual y configurar la dirección IP, la máscara de red y el puerto de enlace.



Cancel **Cableada** Aplicar

Detalles Identidad **IPv4** IPv6 Seguridad

Método IPv4

☐ Automático (DHCP) ☐ Sólo enlace local

☒ Manual ☐ Desactivar

☐ Compartida con otros equipos

Direcciones

Dirección	Máscara de red	Puerta de enlace	
190.124.48.2	255.255.255.128	190.124.48.1	

Imagen 22. Asignación de la dirección IP a la PC1.

En cambio, los demás VPC se configurarán de la siguiente manera:

PC2> ip add 170.10.10.10/25 170.10.10.1

```
PC2> show ip
NAME       : PC2[1]
IP/MASK    : 190.124.49.2/25
GATEWAY    : 190.124.49.1
DNS        :
MAC        : 00:50:79:66:68:00
LPORT     : 10076
RHOST:PORT : 127.0.0.1:10077
MTU       : 1500
```

Imagen 23. Asignación de la dirección IP a la PC2.

PC3> ip add 200.200.200.140/25 200.200.200.129

```
PC3> show ip
NAME       : PC3[1]
IP/MASK    : 190.124.49.130/25
GATEWAY    : 190.124.49.129
DNS        :
MAC        : 00:50:79:66:68:01
LPORT     : 10080
RHOST:PORT : 127.0.0.1:10081
MTU       : 1500
```

Imagen 24. Asignación de la dirección IP a la PC3.

PC4> ip add 200.200.200.140/25 200.200.200.129

```
PC4> show ip
NAME       : PC4[1]
IP/MASK     : 190.124.48.130/25
GATEWAY     : 190.124.48.129
DNS         :
MAC         : 00:50:79:66:68:02
LPORT      : 10078
RHOST:PORT  : 127.0.0.1:10079
MTU         : 1500
```

Imagen 25. Asignación de la dirección IP a la PC4.

Una vez configurados los routers, las vpc y la máquina virtual, se realizarán breves pruebas de envío de paquetes entre PCs. En la siguiente imagen podemos visualizar los resultados en los que la lista de acceso implementado en la PC 3 no permite la llegada de paquetes al host de destino.

```
PC4> ping 190.124.49.2
190.124.49.2 icmp_seq=1 timeout
190.124.49.2 icmp_seq=2 timeout
190.124.49.2 icmp_seq=3 timeout
190.124.49.2 icmp_seq=4 timeout
190.124.49.2 icmp_seq=5 timeout
```

Imagen 26. Envío de paquetes e información de la ruta recorrida de la PC4 a la PC3

En el PC2 se ve la llegada con éxito de los paquetes al PC4.

```
PC2> ping 190.124.48.130
84 bytes from 190.124.48.130 icmp_seq=1 ttl=60 time=109.088 ms
84 bytes from 190.124.48.130 icmp_seq=2 ttl=60 time=122.043 ms
84 bytes from 190.124.48.130 icmp_seq=3 ttl=60 time=125.052 ms
84 bytes from 190.124.48.130 icmp_seq=4 ttl=60 time=122.861 ms
84 bytes from 190.124.48.130 icmp_seq=5 ttl=60 time=108.068 ms
```

Imagen 27. Envío de paquetes e información de la ruta recorrida de la PC2 a la PC4.

Comandos

Show ip interface brief

Muestra la configuración de las interfaces del router junto con su IP asignada.

Show version

Muestra detalles físicos del router.

Show ip ssh

Muestra si el SSH está habilitado, así como los detalles de su configuración.

Show ip route

Muestra el enrutamiento del router, así como los enrutamientos que se le implementaron.

Show startup-config

Muestra las configuraciones en general que se le establecieron al router.

Show arp

Muestra el mapeo de las direcciones IP.

Show ip nat translations

Muestra la traducción de las nat implementadas en el router.

Show ip Access-list

Muestra las listas de control de acceso establecidas en el dispositivo.

Programa

Lo fundamental del proyecto es el programa que, desde el usuario, realizaremos las consultas a los servidores y guardar los resultados de las consultas. La siguiente imagen muestra el resultado de la interfaz gráfica que fue realizado con la herramienta de desarrollo de interfaces gráficas Glade, en la que dentro de definieron los componentes de la interfaz gráfica y los eventos que se realizan al momento de que el usuario interactúe con alguno de los elementos.

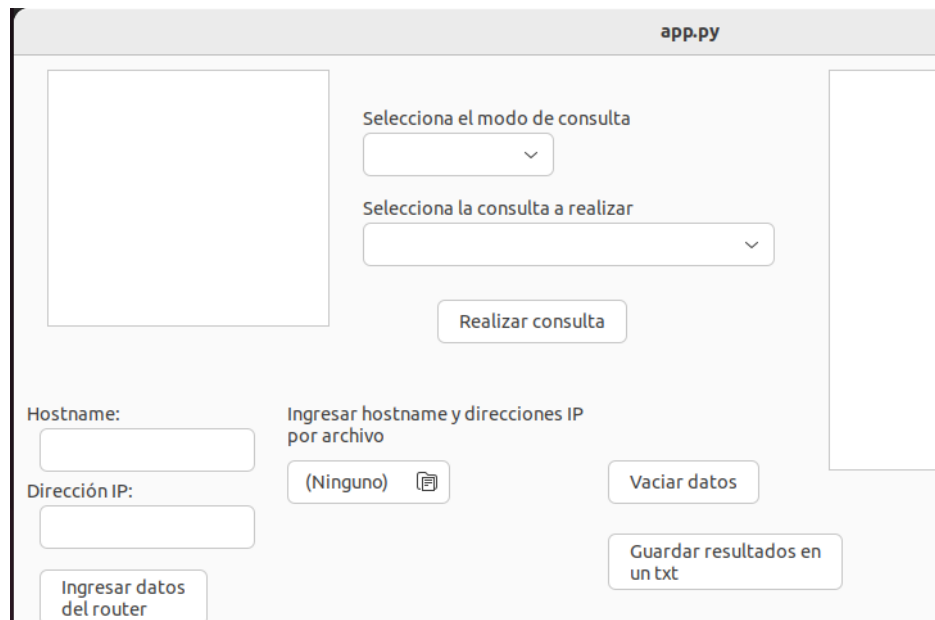


Imagen 28. Interfaz gráfica del programa.

Dentro de los cuadro de selección, podemos elegir el modo de consulta y el tipo de consulta a realizar. En el cuadro blanco del lateral izquierdo se reflejará el hostname y la dirección

IP que sean ingresados por el usuario, mientras que en el cuadro del lateral derecho se mostrarán los resultados de la consulta realizada. Para eliminar los datos ingresados por el usuario, se da click en el botón “Vaciar datos”. Si se quiere guardar los resultados obtenidos de la consulta se da click en “Guardar resultados en un txt”.

Para el ingreso del hostname y de la dirección IP de una de las interfaces, del router, se pueden ingresar manualmente o por medio de un archivo de tipo JSON. En caso de ser de tipo JSON, se deberá de ingresar la información de la siguiente manera (con doble comillas):

```
{“Nombre del hostname”:{“prompt”:”Nombre del hostname#”, “ip” :  
“Dirección_IP”}}
```

A continuación, se presentará el siguiente ejemplo:

```
{“iosv-1”:{“prompt”:”iosv-1#”, “ip”:  
“172.16.1.20”},  
“iosv-2”:{“prompt”:”iosv-2#”, “ip”:”172.16.1.21”}}
```

Imagen 29. Ejemplo del ingreso de datos por medio del archivo JSON.

Pasando a la codificación del programa, se declaran las siguientes librerías:

```
1 import gi, time, json, pexpect, paramiko, pytz  
2 from datetime import datetime  
3  
4 gi.require_version("Gtk", "3.0")  
5  
6 from gi.repository import Gtk
```

Imagen 30. Declaración de las librerías del programa.

Como se vio en la introducción, las librerías paramiko y pexpect se usaran para poder realizar las consultas, time para las pausas que se realicen en el protocolo SSH, datetime y pytz para las fechas, json para el uso de los archivos JSON ingresados por el usuario. La librería gi será de suma utilidad para poder hacer uso de la interfaz gráficas y las acciones con las que puede interactuar el usuario, entre lo importante es la versión requerida, ya que sin esa línea de código se pueden presentar errores al momento de entrar en ejecución.

Posteriormente se declaran las variables globales que serán utilizados y modificados con un valor en algunas funciones.

```

concatenate = ""
dispositivos = {}
resultado = ""
option_selected_command = None
option_selected_mode = None
send_comm = ""
type_route = {"C": "Conectado", "S": "Estático", "R": "RIP",
"B": "BGP", "D": "EIGRP", "EX": "EIGRP externo", "O": "OSFP", "IA": "OSPF area
interna", "N1": "OSPF NSSA externo tipo 1", "N2": "OSPF NSSA externo tipo
2", "E1": "OSPF externo tipo 1", "E2": "OSPF externo tipo 2", "L1": "IS-IS nivel
1", "L2": "IS-IS nivel 2", "*": "candidate default", "U": "Router estatico por
usuario", "P": "Ruta estatica descargada periodicamente"}

```

Imagen 31. Declaración de las variables globales.

La función `routeInNetwork` verifica los tipos de enrutamiento encontrados en la consulta, devolviendo el significado de cada tipo de enrutamiento dentro de una lista.

```

def routeInNetwork(result_route):
    route_used = []
    for enrut in type_route.keys():
        if enrut in result_route:
            route_used.append(type_route[enrut])
    return route_used

```

Imagen 32. Función `routeInNetwork`.

En la siguiente función se guarda el tipo de protocolo seleccionado por el usuario.

```

def selectionComboBoxMode(combo):
    global option_selected_mode
    option_selected_mode = combo.get_active()
    print(option_selected_mode)

```

Imagen 33. Función `selectionComboBoxMode`.

Para la captura de la instrucción a realizar se realiza por medio de la siguiente función, que guarda el comando seleccionado.

```
def selectionComboBoxCommand(combo):

    global option_selected_command
    global send_comm

    option_selected_command = combo.get_active()
    print(option_selected_command)

    if option_selected_command == 0:
        send_comm = "show ip interface brief" + " | exclude unas\n"
    elif option_selected_command == 1:
        send_comm = "show version" + " | i V\n"
    elif option_selected_command == 2:
        send_comm = "show ip ssh" + " \n"
    elif option_selected_command == 3 or option_selected_command == 4:
        send_comm = "show ip route" + " \n"
    elif option_selected_command == 5:
        send_comm = "show startup-config" + " \n"
    elif option_selected_command == 6:
        send_comm = "show arp" + " \n"
    elif option_selected_command == 7:
        send_comm = "show ip nat translations" + " \n"
    elif option_selected_command == 8:
        send_comm = "show ip access-list" + " \n"
```

Imagen 34. Función selectionComboBoxCommand.

Para la selección del hostname y la dirección IP, se utiliza la siguiente función, que dentro se actualizan los dispositivos ya guardados y realiza el proceso para mostrar en uno de los cuadros los dispositivos ingresados hasta el momento.

```
def seleccionarArchivo(filechooserbutton):
    name_file = filechooserbutton.get_filename()

    with open(name_file, 'r') as f:
        devices = json.load(f)
        print(devices)

    global dispositivos
    dispositivos.update(devices)

    list_keys = list(devices.keys())

    global concatenate

    for i in range(len(list_keys)):
        concatenate = concatenate + list_keys[i] + " " + dispositivos[list_keys[i]]
    ["ip"] + "\n"

    #print(concatenate)

    text_buffer = square_text.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    text_buffer.set_text(concatenate)
```

Imagen 35. Función seleccionarArchivo.

De las funciones sustanciales del programa, y que llega a hacer uso de funciones antes mencionadas es performConsult, dentro se realiza la consulta dependiendo del protocolo escogido, sin omitir que todo este proceso se realiza con ayuda de las variables globales previamente vistas. Dentro de lo que cabe mencionar, el username y password a utilizar será para todos los routers, esto solo será utilizado para la realización de este proyecto.

Entre lo que se puede mencionar es que dentro del condicional, dependiendo del protocolo escogido, se realizará el proceso de la consulta que como podemos ver, en caso de ser 0 el protocolo será del tipo Telnet.

```
def performConsult(button):  
  
    global dispositivos  
    global send_comm  
    global option_selected_mode  
    global option_selected_command  
    global resultado  
  
    username = 'cisco'  
    password = 'cisco'  
  
    resultado = ""  
    result_routes = []  
  
    if option_selected_mode == 0:  
  
        for dispositivo in dispositivos.keys():  
            device_prompt = dispositivos[dispositivo]['prompt']  
            child = pexpect.spawn('telnet ' + dispositivos[dispositivo]['ip'])  
  
            child.expect('Username:')  
            child.sendline(username)  
            child.expect('Password:')  
            child.sendline(password)  
            child.expect(device_prompt)  
            child.sendline(send_comm)
```

Imagen 36. Función performComsult (Parte 1).

Dentro de la selección del protocolo Telnet, se recorre los dispositivos ingresados mediante el ciclo for, en cada ciclo tendrá que realizar el envío de comando del router correspondiente, en caso de que la instrucción seleccionada con anterioridad sea el de consultar el enrutamiento general de la red, realizará un proceso para solo mostrar los enrutamientos y sus protocolos implementados.

```
if option_selected_mode == 0:  
  
    for dispositivo in dispositivos.keys():  
        device_prompt = dispositivos[dispositivo]['prompt']  
        child = pexpect.spawn('telnet ' + dispositivos[dispositivo]['ip'])  
  
        child.expect('Username:')  
        child.sendline(username)  
        child.expect('Password:')  
        child.sendline(password)  
        child.expect(device_prompt)  
        child.sendline(send_comm)  
        child.expect(device_prompt)  
        #print(child.before.decode('UTF-8'))  
  
        if option_selected_command == 4:  
  
            salida_decode = child.before.decode('UTF-8')  
            salida_decode = salida_decode[salida_decode.index("Gateway"):]  
  
            result_routes = result_routes + routeInNetwork(salida_decode)  
  
        else:  
  
            resultado = resultado + child.before.decode('UTF-8') + "\n"  
  
    child.sendline('exit')
```

Imagen 37. Función performComsult (Parte 2).

Al terminar de recorrer el diccionario, se procede a mostrar en pantalla los resultados, dependiendo de la instrucción seleccionada se aplicará su respectivo proceso.

```
if option_selected_command == 4:

    result_routes = list(dict.fromkeys(result_routes))

    result_print = "Enrutamiento configurado en la red: \n"

    for element in result_routes:
        result_print = result_print + element + "\n"

    text_buffer = etiqueta_resultado.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")
    text_buffer.set_text(result_print)

else:

    text_buffer = etiqueta_resultado.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")
    text_buffer.set_text(resultado)
```

Imagen 38. Función performComsult (Parte 3).

En el lado de que si el protocolo seleccionado es SSH, se realiza el proceso correspondiente para la conexión y consulta con el router correspondiente de la iteración. Lo que se ve diferente al anterior selección es el proceso para realizar la conexión y consulta por SSH, el guardado de resultado y muestra en pantalla de lo obtenido son similares a lo antes visto.

```
elif option_selected_mode == 1:

    for dispositivo in dispositivos.keys():

        conexion = paramiko.SSHClient()

        conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())

        conexion.connect(dispositivos[dispositivo]['ip'],username =
        username,password=password,look_for_keys=False,allow_agent=False)

        nueva_conexion = conexion.invoke_shell()

        salida = nueva_conexion.recv(5000)

        #print(salida.decode('UTF-8'))

        nueva_conexion.send(send_comm)

        time.sleep(3)

    if option_selected_command == 4:
        salida = nueva_conexion.recv(5000)
        salida_decode = salida.decode('UTF-8')

        salida_decode = salida_decode[salida_decode.index("Gateway"):]
```

Imagen 39. Función performComsult (Parte 4).


```

if option_selected_command == 4:
    salida = nueva_conexion.recv(5000)
    salida_decode = salida.decode('UTF-8')

    salida_decode = salida_decode[salida_decode.index("Gateway"):]

    result_routes = result_routes + routeInNetwork(salida_decode)
    resultado = resultado + salida_decode('UTF-8') + "\n"
else:

    salida = nueva_conexion.recv(5000)

    resultado = resultado + salida.decode('UTF-8') + "\n"

#print(salida.decode('UTF-8'))

nueva_conexion.close()

```

Imagen 40. Función performComsult (Parte 5).

```

if option_selected_command == 4:

    result_routes = list(dict.fromkeys(result_routes))

    result_print = "Enrutamiento configurado en la red: \n"

    for element in result_routes:
        result_print = result_print + element + "\n"

    text_buffer = etiqueta_resultado.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")
    text_buffer.set_text(result_print)

else:

    text_buffer = etiqueta_resultado.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")
    text_buffer.set_text(resultado)

```

Imagen 41. Función performComsult (Parte 6).

En la función ingresar_texto se incorpora los datos del hostname y dirección IP dadas al diccionario que se utilizará para las consultas de los dispositivos.

```

def ingresar_texto(button):

    text_buffer = square_text.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")

    global concatenate
    global dispositivos

    in_hostname = entrada_operando_1.get_text()
    in_ip = entrada_operando_2.get_text()

    concatenate = concatenate + in_hostname + " " + in_ip + "\n"

    hostname_prompt = in_hostname + "##"

    dispositivos[in_hostname] = {"prompt":hostname_prompt,"ip":in_ip}

    print(dispositivos)

    text_buffer.set_text(concatenate)

```

Imagen 42. Función ingresar_texto.

emptyData es utilizado para el vacío del contenido del diccionario dispositivos y de los dispositivos mostrados en la interfaz gráfica.

```
def emptyData(button):

    text_buffer = square_text.get_buffer()
    scrolled_window = Gtk.ScrolledWindow()
    #scrolled_window.add("Si se pudo")

    global dispositivos
    global concatenate

    concatenate = ""
    dispositivos.clear()

    text_buffer.set_text(concatenate)
```

Imagen 43. Función emptyData.

Para saveResults, se guardará en un archivo txt los resultados obtenidos después de la consulta, siempre y cuando el usuario lo indique al presionar el botón correspondiente.

```
def saveResults(button):
    global resultado
    datetime_Mex = datetime.now(pytz.timezone('Mexico/General'))
    format_date = datetime_Mex.strftime('%d%m%Y_%H%M%S')
    title_file = "Resultados_" + format_date + ".txt"

    with open(title_file, 'wb') as f:
        f.write(resultado.encode('UTF-8'))
```

Imagen 44. Función saveResults.

Lo último a mostrar es el guardado de los eventos, creación de las clases correspondientes a los elementos por los que se va a ingresar el contenido y el despliegue de la interfaz gráfica. Dentro de lo que se puede destacar es el diccionario handler, ya que se almacenan los eventos correspondientes a dichos elementos de la interfaz gráfica y relacionándolo con la alguna de las funciones anteriormente mostradas.

```
builder = Gtk.Builder()
builder.add_from_file("gui_prueba.glade")

handlers = {"cerrar_ventana":Gtk.main_quit,
"click_boton1":performConsult,"click_boton_2":ingresar_texto,
"select_file":seleccionarArchivo, "opt_select":selectionComboBoxCommand,
"opt_mode":selectionComboBoxMode,
"click_vaciar":emptyData,
"click_guardar":saveResults}

builder.connect_signals(handlers)

etiqueta_resultado = builder.get_object("cuadro_texto")
square_text = builder.get_object("cuadro_texto_2")

entrada_operando_1 = builder.get_object("in_1")
entrada_operando_2 = builder.get_object("in_2")

window = builder.get_object("ventana")

window.show_all()

Gtk.main()
```

Imagen 45. Contenido general del programa.

Ejecución del programa

Para el ejemplo utilizaremos 2 routers de la topología, que serán el router Ra y R1, por lo que tendremos que ingresar sus hostname y direcciones IP. También tendremos que escoger el protocolo y la instrucción a realizar, primero se realizara la prueba con Telnet y con la instrucción de mostrar la configuración de los puertos de enlace.

The screenshot shows a web application titled 'app.py'. It has a text area on the left containing 'Ra 190.124.48.1' and 'R1 20.11.12.25'. To the right, there are two dropdown menus: 'Selecciona el modo de consulta' set to 'Telnet' and 'Selecciona la consulta a realizar' set to 'Mostrar configuracion de los dispositivos'. Below these is a 'Realizar consulta' button. At the bottom, there are input fields for 'Hostname:' (containing 'R1') and 'Dirección IP:' (containing '20.11.12.25'). There is also a file upload button labeled '(Ninguno)' with a document icon, a 'Vaciar datos' button, and a 'Guardar resultados en un txt' button. A red box highlights an 'Ingresar datos del router' button at the bottom left.

Imagen 46. Ingreso de los datos de los routers y consulta a realizar.

Al dar clic en el botón de “realizar consulta”, nos aparecerá en el cuadro del lado de la derecha el resultado

```
show ip interface brief | exclude unas
Interface      IP-Address    OK? Method Status
Protocol
FastEthernet0/0 190.124.48.1 YES NVRAM up
up
FastEthernet0/1 20.11.12.26 YES NVRAM up
up
Loopback0       192.168.1.1  YES NVRAM up    up

show ip interface brief | exclude unas
Interface      IP-Address    OK? Method Status
Protocol
FastEthernet0/0 20.11.12.25 YES NVRAM up
up
FastEthernet0/1 20.11.12.1  YES NVRAM up
up
Serial4/0       20.11.12.13 YES NVRAM up    down
Serial4/1       20.11.12.18 YES NVRAM up    down
```

Imagen 47. Resultado de la consulta por el protocolo Telnet.

Se realizará la misma consulta, pero se cambiará el protocolo Telnet por SSH.

Selecciona el modo de consulta

SSH ▼

Selecciona la consulta a realizar

Mostrar configuracion de los dispositivos. ▼

Realizar consulta

Imagen 48. Cambio de protocolo Telnet a SSH.

Al término de la ejecución de la consulta, nos aparecerá el siguiente resultado.

```
show ip interface brief | exclude unas
Interface      IP-Address    OK? Method Status
Protocol
FastEthernet0/0 190.124.48.1 YES NVRAM up
up
FastEthernet0/1 20.11.12.26 YES NVRAM up
up
Loopback0       192.168.1.1 YES NVRAM up      up
Ra#
show ip interface brief | exclude unas
Interface      IP-Address    OK? Method Status
Protocol
FastEthernet0/0 20.11.12.25 YES NVRAM up
up
FastEthernet0/1 20.11.12.1 YES NVRAM up
up
Serial4/0       20.11.12.13 YES NVRAM up      down
Serial4/1       20.11.12.18 YES NVRAM up      down
```

Imagen 49. Contenido general del programa.

Conclusión

La realización de este proyecto dio un panorama sobre el cómo se puede desarrollar una aplicación o programa enfocado en el área de las redes de comunicación que, desde el lenguaje de programación Python, se realizó el programa para la realización de consultas a la red o topología correspondiente. Durante el desarrollo del programa, surgieron dificultades que iban desde la comprensión de las librerías pexpect y paramiko hasta el diseño de la interfaz gráfica.

Dentro de las dificultades que hubo en las librerías fue que, mediante el uso de la librería paramiko, al momento de realizar una consulta se presentaba un error relacionado con el puerto 22 que, después de reflexionar, se debió a la generación de llaves en los routers, esto fue provocado por el cierre del programa o cada que los routers se apaguen. Debido a eso, se tienen que ingresar las llaves cada que se prendan los routers.

De las dificultades relacionadas a la implementación de la interfaz gráfica fue el identificar un IDE óptimo que se pueda utilizar en el sistema operativo Linux, que aparte del análisis, también se tomó el tiempo para entender el funcionamiento del programa y realizar una investigación de la implementación de los elementos de la interfaz gráfica y realizar una

arquitectura aceptable que no se tenga dificultad de usar y que sea funcional con el archivo donde están implementados los algoritmos.

Dentro de las dificultades encontradas en la topología fueron las implementaciones de las listas de control de acceso y las NAT, ya que al momento de ingresar el hostname y la dirección IP (relacionado con la interfaz donde esté ubicada la lista de acceso) en el programa y realizar una consulta, entraba en un ciclo que no arrojaba una respuesta, por lo que forzosamente se cerraba el programa.

Se atravesó por varias problemáticas en el desarrollo del programa que, ante las dificultades, se lograron resolver por medio del análisis del problema y la propuesta de soluciones que fueron realizadas y descartar una a una hasta llegar con la solución óptima.

Bibliografía

<https://www.ibm.com/docs/es/aix/7.1?topic=protocols-telnet-protocol>

<https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/telnet>

<https://netwgeeks.com/telnet-vs-ssh/#:~:text=Telnet%20utiliza%20el%20puerto%20TCP,deseamos%20acceder%2C%20es%20el%20Servidor.>

<https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>