

PRUEBA DE HABILIDADES PRÁCTICAS **CCNA**

Presentado a:
Efraín Alejandro Pérez

Estudiante:
Jennifer María Mireya Vargas Villalobos

GRUPO: 203092_4

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA - UNAD
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA
ENERO 2020

TABLA DE CONTENIDO

RESUMEN	5
ABSTRACT	6
INTRODUCCIÓN	7
OBJETIVOS	8
Objetivo general.....	8
Objetivos específicos	8
ESCENARIO 1	9
TOPOLOGÍA:	9
DESARROLLO.....	10
PARTE 1: ASIGNACIÓN DE DIRECCIONES IP	10
Serial 0/0/0	12
Int gigabit 0/0.....	12
Puerto gigabit 0/0 en el router BOGOTA	13
Router Cali, configuración puertos s0/0/0 y gigabit 0/0	13
IP (PC0) MEDELLIN.....	14
IP (PC1) MEDELLIN.....	14
PARTE 2: CONFIGURACIÓN BÁSICA.	14
Tabla de direccionamiento con protocolo EIGRP	15
Show ip interface brief – router Medellín	15
Show ip interface brief – router Bogotá	16
Show ip interface brief – router Cali	16
Show IP route – Medellín	17
Show IP route – Bogotá.....	18
Show IP route – Cali	18
Salida del comando show cdp interface	19
Ping fallido entre el PC1, red Medellín y router Cali	19
Ping correcto entre la red Medellín.....	20
Ping correcto entre PC WS 1 y servidor red Bogotá	20
Ping correcto entre la red de Cali	21
PARTE 3: CONFIGURACIÓN DE ENRUTAMIENTO.	21

Protocolo EIGRP router Medellín.....	21
Topología pin con solicitud de configuración del reloj.....	22
Protocolo EIGRP router Bogotá.....	23
Protocolo EIGRP router Cali	23
Vecindad de routers con protocolo EIGRP	24
Verificación tablas de enrutamiento de los tres routers	26
Dirección IP del PC0 en la red de MEDELLIN.....	27
Ping exitoso entre PC2 RED Cali y PC0 red MEDELLIN	27
Ping exitoso entre PC2 RED Cali y Servidor red BOGOTA.....	28
PARTE 4: CONFIGURACIÓN DE LAS LISTAS DE CONTROL DE ACCESO.....	28
Conexión Telnet correcta entre PC2 (red CALI) y router MEDELLIN.....	30
Configuración router CALI – telnet.....	31
Conexión por Telnet	31
PARTE 5: COMPROBACIÓN DE LA RED INSTALADA.	33
Resultado de ping Entre PC2 y PC0	33
Ping entre PC2 y servidor	34
ESCENARIO 2.....	35
DESARROLLO	36
ROUTERS BUCARAMANGA	41
R Bucaramanga S0/0/0	43
R Bucaramanga G0/0	43
R Tunja puerto s0/0/0.....	44
R Tunja puerto S0/0/1	44
R Tunja puerto G0/0	45
R Tunja Puerto G0/1.....	45
R Cundinamarca S0/0/0	46
R Cundinamarca – interfaz G0/0	46
Switch Red Bucaramanga – config VLAN 10	47
Switch Red CUNDINAMARCA – config VLAN 20	47
Switch Red CUNDINAMARCA – config VLAN 10 y 88	48
Switch Red CUNDINAMARCA – config VLAN 20 y 30	48
CONCLUSIONES	49

AGRADECIMIENTOS.....	50
BIBLIOGRAFÍA.....	51

RESUMEN

A lo largo del curso se pudo evidenciar la gran cantidad de conocimientos enfocados completamente al Networking, motivo por el cual en este documento se presentarán algunas habilidades adquiridas con la aplicación de la prueba CCNA enfocadas a unos escenarios debidamente desarrollados con el fin de establecer una correcta comprensión del uso de estas tecnologías. Las configuraciones necesarias para dar correcto funcionamiento se realizarán a través del software Packet Tracer, desarrollado por la empresa certificadora CISCO.

ABSTRACT

Throughout the course it was possible to demonstrate the great amount of knowledge focused entirely on Networking, which is why in this document some skills acquired with the application of the CCNA test focused on properly developed scenarios will be presented in order to establish a correct Understanding the use of these technologies. The necessary configurations to function correctly will be made through the Packet Tracer software, developed by the CISCO certification company.

INTRODUCCIÓN

A lo largo de los años, se ha evidenciado un gran crecimiento por parte de las comunicaciones debido al gran impacto tecnológico que este aporta a la comunidad; logrando así generar grandes competencias que cada día se destacan por brindar a los usuarios un mejor servicio.

Las redes (especialmente las de nueva generación) como complemento principal son aquellas las cuales requieren de gran conocimiento específico al área de las telecomunicaciones, por lo que en este documento se procederá a presentar una solución aplicada a esta problemática. Se procederá a desarrollar una dedicada compilación de conocimientos adquiridos junto con la ejecución de protocolos y servicios predeterminados.

OBJETIVOS

Objetivo general

Desarrollar una alternativa de solución a dos escenarios planteados por la prueba de competencias de CISCO CCNA.

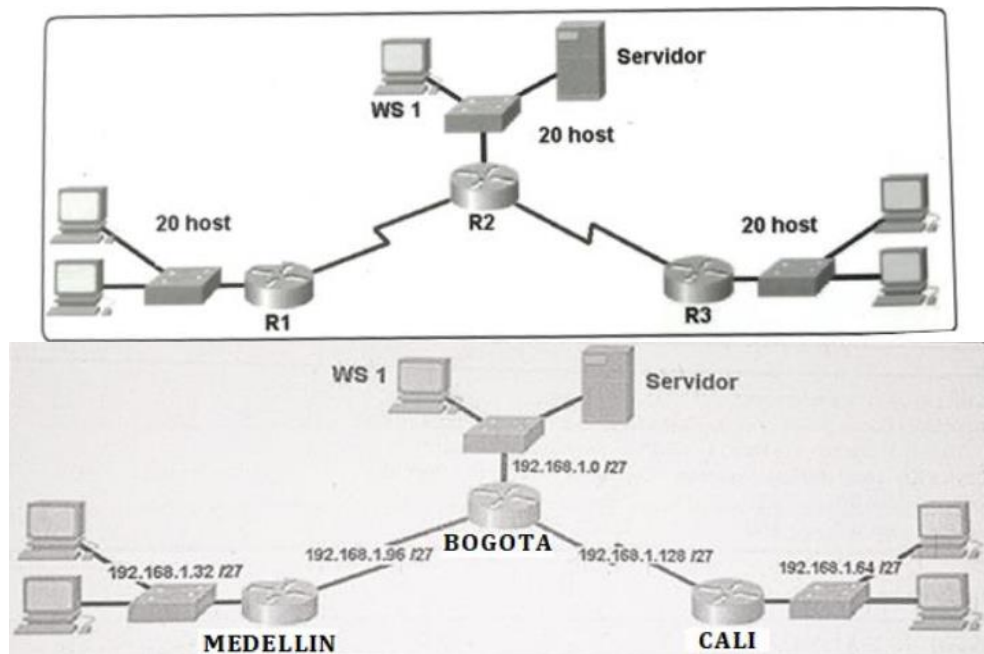
Objetivos específicos

- Especificar los servicios a brindar por cada uno de los servidores de cada escenario.
- Realizar las configuraciones necesarias a los diferentes dispositivos de comunicación como lo son routers, servidores y demás.
- Verificar el funcionamiento entre los diferentes elementos que conforman la red dando uso del comando ping.

ESCENARIO 1

Una empresa posee sucursales distribuidas en las ciudades de Bogotá, Medellín y Cali en donde el estudiante será el administrador de la red, el cual deberá configurar e interconectar entre sí cada uno de los dispositivos que forman parte del escenario, acorde con los lineamientos establecidos para el direccionamiento IP, protocolos de enrutamiento y demás aspectos que forman parte de la topología de red. ("IT Networking Career Opportunities - Cisco," n.d.)

TOPOLOGÍA:



DESARROLLO

Inicialmente se proceden a realizar rutinas de diagnóstico y proceder a dejar los equipos listos para su configuración (asignar nombres de equipos, asignar claves de seguridad, etc), con la intención de establecer un método eficiente de la configuración determinada; consiguiente a ello se procede a realizar la conexión física de los equipos con base en la topología de red y configurar la topología de red, de acuerdo con las siguientes especificaciones:

PARTE 1: ASIGNACIÓN DE DIRECCIONES IP

Se procede a subnetear la red creando una segmentación en ocho partes, para permitir crecimiento futuro de la red corporativa. La mejor forma de poder subnetear la red, es tomando del último octeto ciertos bits más significativos prestados, y el número de bits prestados está dado por la siguiente formula de 2^n donde sabemos que n es el número de bits.

Un ejemplo podría ser que, si deseamos tener un Subneteo de red en ocho partes, se toman 3 bits, como $2^3 = 8$. La forma de poder conocer el número de host que cada sub red puede tener como máximo está dado por:

$$2^n - 2 = 2^5 - 2 = 32 - 2 = 30$$

Se procede de esta manera a crear las ocho tablas respectivas que nos darán las direcciones de cada red para obtener:

SUBRED	DIRECCIÓN DE RED	RANGO DE HOST	DIRECCIÓN DE DIFUSIÓN
0	192.168.1.0/27	192.168.1.1 – 192.168.1.30	192.168.1.31
1	192.168.1.32/27	192.168.1.33 – 192.168.1.62	192.168.1.63
2	192.168.1.64/27	192.168.1.65 – 192.168.1.94	192.168.1.95
3	192.168.1.96/27	192.168.1.97 – 192.168.1.126	192.168.1.127
4	192.168.1.128/27	192.168.1.129 – 192.168.1.158	192.168.1.159
5	192.168.1.160/27	192.168.1.161 – 192.168.1.190	192.168.1.191
6	192.168.1.192/27	192.168.1.193 – 192.168.1.222	192.168.1.223
7	192.168.1.224/27	192.168.1.225 – 192.168.1.254	192.168.1.255

Se puede evidenciar que las redes obtenidas mediante el Excel corresponden en su mayoría y de las 8 obtenidas usaremos las primeras 5 (se reservan 3 redes más para un posible crecimiento de la red).

Ahora se procede a asignar una dirección IP a la red, por lo vamos al simulador de Packet Tracer y se tendría:

MEDELLIN

Physical Config CLI Attributes

IOS Command Line Interface

```
MEDELLIN(config)#int s0/0
%Invalid interface type and number
MEDELLIN(config)#int s0/0/0
MEDELLIN(config-if)#ip add
MEDELLIN(config-if)#ip address 192.168.1.99 255.255.255.224
MEDELLIN(config-if)#no shu
MEDELLIN(config-if)#no shutdown
```

Serial 0/0/0

MEDELLIN

Physical Config CLI Attributes

IOS Command Line Interface

```
MEDELLIN(config-if)#exit
MEDELLIN(config)#int g0/0
MEDELLIN(config-if)#ip add
MEDELLIN(config-if)#ip address 192.168.1.33 255.255.255.224
MEDELLIN(config-if)#no shut
MEDELLIN(config-if)#no shutdown
```

Int gigabit 0/0

BOGOTA

Physical Config CLI Attributes

IOS Command Line Interface

```
BOGOTA>enable
BOGOTA#config t
Enter configuration commands, one per line. End with CNTL/Z.
BOGOTA(config)#int s0/0/1
BOGOTA(config-if)#ip add
BOGOTA(config-if)#ip address 192.168.1.130 225.225.225.224
Bad mask 0xE1E1E1E0 for address 192.168.1.130
BOGOTA(config-if)#ip address 192.168.1.130 225.225.225.0
Bad mask 0xE1E1E100 for address 192.168.1.130
BOGOTA(config-if)#ip address 192.168.1.130 255.255.255.224
BOGOTA(config-if)#no shu
BOGOTA(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/1, changed state to down
BOGOTA(config-if)#exit
BOGOTA(config)#int s0/0/0
BOGOTA(config-if)#ip add
BOGOTA(config-if)#ip address 192.168.1.98 255.255.255.224
BOGOTA(config-if)#no shut
BOGOTA(config-if)#no shutdown

BOGOTA(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
```

Router Bogotá, puertos serial 0/0 y 0/1

```
BOGOTA(config-if)#exit
BOGOTA(config)#int g0/0
BOGOTA(config-if)#ip add
BOGOTA(config-if)#ip address 192.168.1.1 255.255.255.224
BOGOTA(config-if)#no shu
BOGOTA(config-if)#no shutdown
BOGOTA(config-if)#exit
BOGOTA(config)#
```

Puerto gigabit 0/0 en el router BOGOTA

CALI

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```
CALI>enable
CALI#config t
Enter configuration commands, one per line. End with CNTL/Z.
CALI(config)#int s0/0/0
CALI(config-if)#ip add
CALI(config-if)#ip address 192.168.1.131 255.255.255.224
CALI(config-if)#no shut
CALI(config-if)#no shutdown

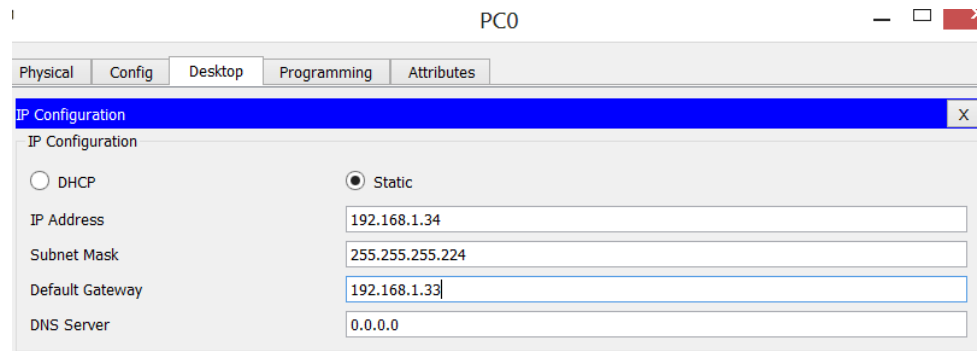
CALI(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

CALI(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up

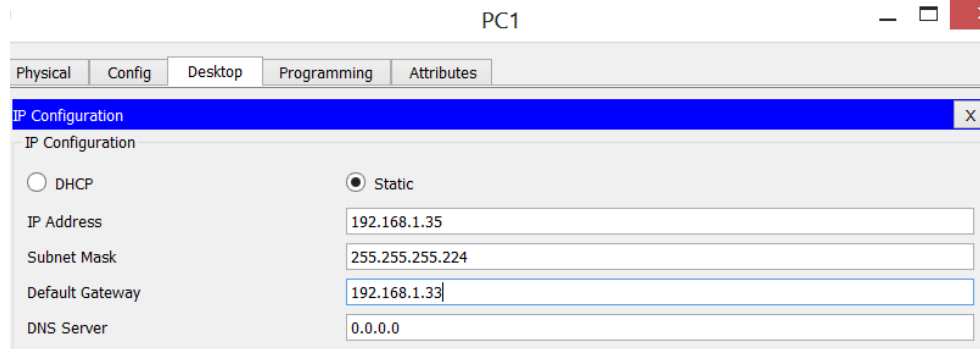
CALI(config-if)#exit
CALI(config)#int g0/0
CALI(config-if)#ip add
CALI(config-if)#ip address 192.168.1.65 255.255.255.224
CALI(config-if)#no shut
CALI(config-if)#no shutdown

CALI(config-if)#
```

Router Cali, configuración puertos s0/0/0 y gigabit 0/0



IP (PC0) MEDELLIN



IP (PC1) MEDELLIN

Ahora continuamos con la siguiente parte donde realizaremos la configuración básica:

PARTE 2: CONFIGURACIÓN BÁSICA.

- Completar la siguiente tabla con la configuración básica de los routers, teniendo en cuenta las subredes diseñadas.

	R1	R2	R3
Nombre de Host	MEDELLIN	BOGOTA	CALI
Dirección de Ip en interfaz Serial 0/0	192.168.1.99	192.168.1.98	192.168.1.131
Dirección de Ip en interfaz Serial 0/1		192.168.1.130	
Dirección de Ip en interfaz FA 0/0	192.168.1.33	192.168.1.1	192.168.1.65
Protocolo de enrutamiento	Eigrp	Eigrp	Eigrp
Sistema Autónomo	200	200	200
Afirmaciones de red	192.168.1.0	192.168.1.0	192.168.1.0

Tabla de direccionamiento con protocolo EIGRP

- b. Después de cargada la configuración en los dispositivos, verificar la tabla de enrutamiento en cada uno de los routers para comprobar las redes y sus rutas.

Lo ideal para comprobar que se logró configurar apropiadamente cada uno de los router con respecto a la tabla es usando el comando *show ip interface brief*, por lo que se tiene:

```

MEDELLIN
-----
Physical  Config  CLI  Attributes
-----
IOS Command Line Interface
MEDELLIN#show ip int
MEDELLIN#show ip interface bri
MEDELLIN#show ip interface brief
Interface          IP-Address      OK? Method Status
Protocol
GigabitEthernet0/0  192.168.1.33   YES manual up
up
GigabitEthernet0/1  unassigned     YES unset
administratively down down
Serial0/0/0        192.168.1.99   YES manual up
up
Serial0/0/1        unassigned     YES unset
administratively down down
FastEthernet0/1/0  unassigned     YES unset up
down
FastEthernet0/1/1  unassigned     YES unset up
down
FastEthernet0/1/2  unassigned     YES unset up
down
FastEthernet0/1/3  unassigned     YES unset up
down
Vlan1              unassigned     YES unset
administratively down down
MEDELLIN#

```

Show Ip interface brief – router Medellín

BOGOTA

Physical Config CLI Attributes

IOS Command Line Interface

```

Last input 00:00:08, output 00:00:05, output hang never

BOGOTA#sh
BOGOTA#show ip inte
BOGOTA#show ip interface brie
BOGOTA#show ip interface brief
Interface                IP-Address      OK? Method Status
Protocol
GigabitEthernet0/0      192.168.1.1    YES manual up
up
GigabitEthernet0/1      unassigned      YES unset
administratively down down
Serial0/0/0              192.168.1.98   YES manual up
up
Serial0/0/1              192.168.1.130  YES manual up
up
FastEthernet0/1/0       unassigned      YES unset up
down
FastEthernet0/1/1       unassigned      YES unset up
down

```

Show ip interface brief – router Bogotá

CALI

Physical Config CLI Attributes

IOS Command Line Interface

```

CALI>enable
CALI#sh
CALI#show ip inte
CALI#show ip interface brief
Interface                IP-Address      OK? Method Status
Protocol
GigabitEthernet0/0      192.168.1.65   YES manual up
down
GigabitEthernet0/1      unassigned      YES unset
administratively down down
Serial0/0/0              192.168.1.131  YES manual up
up
Serial0/0/1              unassigned      YES unset
administratively down down
FastEthernet0/1/0       unassigned      YES unset up
up
FastEthernet0/1/1       unassigned      YES unset up
down
FastEthernet0/1/2       unassigned      YES unset up
down
FastEthernet0/1/3       unassigned      YES unset up

```

Show ip interface brief – router Cali

- c. Verificar el balanceo de carga que presentan los routers.

Para lograr el balanceo de cargas de los routers se procede a usar el comando *show ip route*, por lo que se tiene:


```
MEDELLIN
Physical Config CLI Attributes
IOS Command Line Interface
E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia -
IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
192.168.1.0/24 is variably subnetted, 6 subnets, 2 masks
D 192.168.1.0/27 [90/2172416] via 192.168.1.98, 00:48:11,
Serial0/0/0
C 192.168.1.32/27 is directly connected, GigabitEthernet0/0
L 192.168.1.33/32 is directly connected, GigabitEthernet0/0
C 192.168.1.96/27 is directly connected, Serial0/0/0
L 192.168.1.99/32 is directly connected, Serial0/0/0
D 192.168.1.128/27 [90/2681856] via 192.168.1.98, 00:48:11,
Serial0/0/0
MEDELLIN#
```

Show IP route – Medellín

```

BOGOTA
-----
Physical  Config  CLI  Attributes
-----
IOS Command Line Interface

P - periodic downloaded static route

Gateway of last resort is not set

  192.168.1.0/24 is variably subnetted, 7 subnets, 2
masks
C       192.168.1.0/27 is directly connected,
GigabitEthernet0/0
L       192.168.1.1/32 is directly connected,
GigabitEthernet0/0
D       192.168.1.32/27 [90/2172416] via 192.168.1.99,
00:50:33, Serial0/0/0
C       192.168.1.96/27 is directly connected, Serial0/0/0
L       192.168.1.98/32 is directly connected, Serial0/0/0
C       192.168.1.128/27 is directly connected,
Serial0/0/1
L       192.168.1.130/32 is directly connected,
Serial0/0/1
BOGOTA#

```

Show IP route – Bogotá

```

CALI
-----
Physical  Config  CLI  Attributes
-----
IOS Command Line Interface

Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia -
IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

  192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.128/27 is directly connected, Serial0/0/0
L       192.168.1.131/32 is directly connected, Serial0/0/0
CALI#

```

Show IP route – Cali

d. Realizar un diagnóstico de vecinos usando el comando `cdp`.

Inicialmente es probable que debido a que en ningún momento se usó el comando `cdp run` en ninguna interfaz la salida pueda ser negativa, por lo que se tiene:

```
BOGOTA#sh
BOGOTA#show cdp inter
BOGOTA#show cdp interface
% CDP is not enabled
BOGOTA#
```

```
CALI#sh
CALI#show cdp inter
CALI#show cdp interface
% CDP is not enabled
CALI#
```

```
MEDELLIN#sh
MEDELLIN#show cdp inter
MEDELLIN#show cdp interface
% CDP is not enabled
MEDELLIN#
```

Salida del comando show cdp interface

- e. Realizar una prueba de conectividad en cada tramo de la ruta usando Ping.

Se procede a realizar *Ping* en cada uno de los dispositivos:

```
C:\>ping 192.168.1.65

Pinging 192.168.1.65 with 32 bytes of data:

Reply from 192.168.1.33: Destination host unreachable.
Reply from 192.168.1.33: Destination host unreachable.
Reply from 192.168.1.33: Destination host unreachable.
Reply from 192.168.1.33: Destination host unreachable.

Ping statistics for 192.168.1.65:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Ping fallido entre el PC1, red Medellín y router Cali

```
C:\>ping 192.168.1.34

Pinging 192.168.1.34 with 32 bytes of data:

Reply from 192.168.1.34: bytes=32 time<1ms TTL=128
Reply from 192.168.1.34: bytes=32 time<1ms TTL=128
Reply from 192.168.1.34: bytes=32 time<1ms TTL=128
Reply from 192.168.1.34: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.34:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 192.168.1.33

Pinging 192.168.1.33 with 32 bytes of data:

Reply from 192.168.1.33: bytes=32 time<1ms TTL=255
Reply from 192.168.1.33: bytes=32 time<1ms TTL=255
Reply from 192.168.1.33: bytes=32 time<1ms TTL=255
Reply from 192.168.1.33: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.33:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping correcto entre la red Medellín

```
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=4ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

Ping correcto entre PC WS 1 y servidor red Bogotá

```
C:\>ping 192.168.1.66

Pinging 192.168.1.66 with 32 bytes of data:

Reply from 192.168.1.66: bytes=32 time<1ms TTL=128
Reply from 192.168.1.66: bytes=32 time<1ms TTL=128
Reply from 192.168.1.66: bytes=32 time<1ms TTL=128
Reply from 192.168.1.66: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ping correcto entre la red de Cali PARTE 3: CONFIGURACIÓN DE ENRUTAMIENTO.

- a. Asignar el protocolo de enrutamiento EIGRP a los routers considerando el direccionamiento diseñado.

Se entiende que aunque los dispositivos cuentan con las direcciones IP de la tabla (especificada anteriormente), se debe iniciar el protocolo de comunicación EIGRP en cada uno de los routers, por lo que se tiene:

```
MEDELLIN
-----
Physical  Config  CLI  Attributes
-----
IOS Command Line Interface

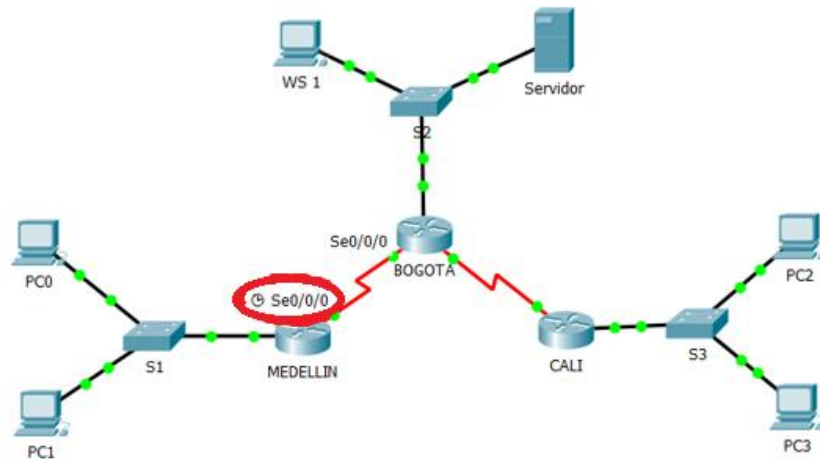
MEDELLIN(config)#int s0/0/0
MEDELLIN(config-if)#clock rate 64000
MEDELLIN(config-if)#router eigrp 200
MEDELLIN(config-router)#192.168.1.33 0.0.0.31
^
% Invalid input detected at '^' marker.

MEDELLIN(config-router)#network 192.168.1.33 0.0.0.31
MEDELLIN(config-router)#network 192.168.1.99 0.0.0.31
MEDELLIN(config-router)#no auto
MEDELLIN(config-router)#no auto-summary
```

Protocolo EIGRP router Medellín

Se evidencia como al puerto *serial 0/0/0* se le configura (únicamente en el extremo de Medellín) el reloj de 64.000, para verificar en cuál de los dos extremos se debe configurar el reloj se puede en Packet Tracer ubicar el puntero del cursor y únicamente en uno de ellos se verá reflejado un pequeño reloj, lo cual se evidencia en la siguiente imagen

resaltada con rojo:



Topología pin con solicitud de configuración del reloj.

Ahora, para la configuración del protocolo **EIGRP** del router Bogotá y Cali tenemos lo siguiente:

```
BOGOTA
Physical Config CLI Attributes
IOS Command Line Interface
CNTL/Z.
BOGOTA(config)#router eigrp 200
BOGOTA(config-router)#net
BOGOTA(config-router)#network 192.168.1.1 0.0.0.224
EIGRP: Invalid address/mask combination (discontiguous
mask)
BOGOTA(config-router)#network 192.168.1.130 0.0.0.31
BOGOTA(config-router)#network 192.168.1.1 0.0.0.31
BOGOTA(config-router)#network 192.168.1.98 0.0.0.31
BOGOTA(config-router)#no auto
BOGOTA(config-router)#no auto-summary
```

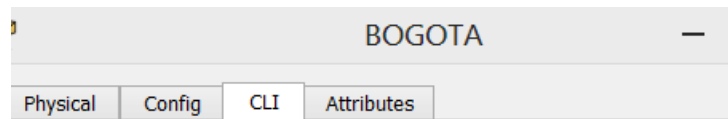
Protocolo EIGRP router Bogotá

```
CALI
Physical Config CLI Attributes
IOS Command Line Interface
CALI>enable
CALI#config t
Enter configuration commands, one per line. End with CNTL/Z.
CALI(config)#router eigrp 200
CALI(config-router)#net
CALI(config-router)#network 192.168.1.65 0.0.0.31
CALI(config-router)#network 192.168.1.131 0.0.0.31
CALI(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 200: Neighbor 192.168.1.130
(Serial0/0/0) is up: new adjacency
```

Protocolo EIGRP router Cali

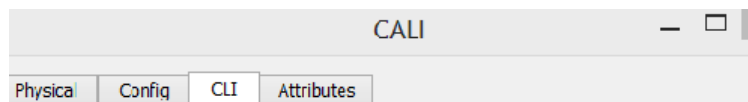
- b. Verificar si existe vecindad con los routers configurados con EIGRP.

Ahora, para la verificación se hace uso del comando *show ip eigrp neighbors*, donde el número del sistema autónomo es 200 según la tabla dada, así:



IOS Command Line Interface

```
BOGOTA>enable
BOGOTA#show ip eigrp neighbors 200
IP-EIGRP neighbors for process 200
H Address Interface Hold Uptime SRTT
RTO Q Seq (sec) (ms)
Cnt Num
0 192.168.1.99 Se0/0/0 11 02:54:37 40
1000 0 5
1 192.168.1.131 Se0/0/1 13 01:14:08 40
1000 0 7
```



IOS Command Line Interface

```
CALI>enable
CALI#show ip eigrp neighbors 200
IP-EIGRP neighbors for process 200
H Address Interface Hold Uptime SRTT RTO Q
Seq (sec) (ms) Cnt
Num
0 192.168.1.130 Se0/0/0 13 01:14:39 40 1000 0
4
```



IOS Command Line Interface

```
MEDELLIN>enable
MEDELLIN#show ip eigrp neighbors 200
IP-EIGRP neighbors for process 200
H Address Interface Hold Uptime SRTT RTO
Q Seq (sec) (ms)
Cnt Num
0 192.168.1.98 Se0/0/0 14 02:54:53 40 1000
0 3
```

Vecindad de routers con protocolo EIGRP

- c. Realizar la comprobación de las tablas de enrutamiento en cada uno de los routers para verificar cada una de las rutas establecidas.

Este procedimiento se realiza mediante el comando *show ip eigrp topology 200*, por lo que se tiene:

BOGOTA

Physical Config **CLI** Attributes

IOS Command Line Interface

```

BOGOTA#show ip eigrp topology 200
IP-EIGRP Topology Table for AS 200/ID(192.168.1.130)

Codes: P - Passive, A - Active, U - Update, Q - Query, R -
Reply,
       r - Reply status

P 192.168.1.0/27, 1 successors, FD is 5120
   via Connected, GigabitEthernet0/0
P 192.168.1.32/27, 1 successors, FD is 2172416
   via 192.168.1.99 (2172416/5120), Serial0/0/0
P 192.168.1.96/27, 1 successors, FD is 2169856
   via Connected, Serial0/0/0
P 192.168.1.128/27, 1 successors, FD is 2169856
   via Connected, Serial0/0/1

```

MEDELLIN

Physical Config **CLI** Attributes

IOS Command Line Interface

```

MEDELLIN#show ip eigrp topology 200
IP-EIGRP Topology Table for AS 200/ID(192.168.1.99)

Codes: P - Passive, A - Active, U - Update, Q - Query, R -
Reply,
       r - Reply status

P 192.168.1.0/27, 1 successors, FD is 2172416
   via 192.168.1.98 (2172416/5120), Serial0/0/0
P 192.168.1.32/27, 1 successors, FD is 5120
   via Connected, GigabitEthernet0/0
P 192.168.1.96/27, 1 successors, FD is 2169856
   via Connected, Serial0/0/0
P 192.168.1.128/27, 1 successors, FD is 2681856
   via 192.168.1.98 (2681856/2169856), Serial0/0/0

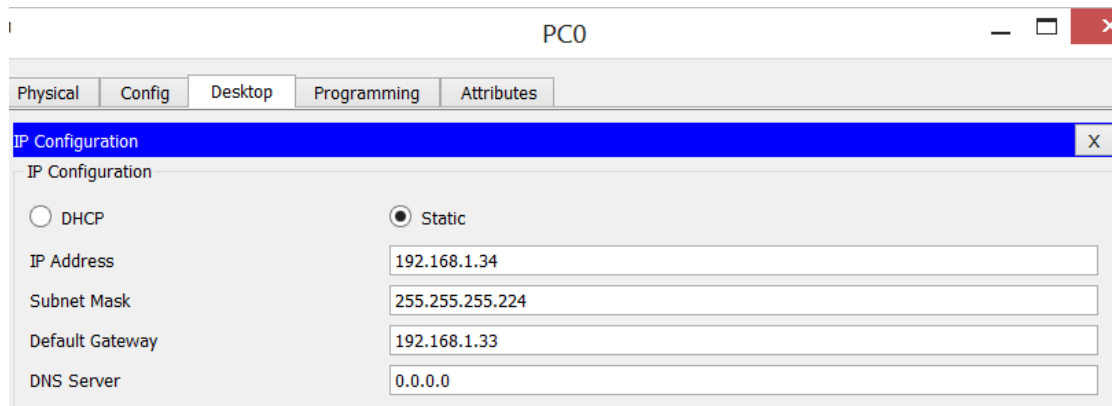
```

```
IOS Command Line Interface
CALI#show ip eigrp topology 200
IP-EIGRP Topology Table for AS 200/ID(192.168.1.131)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
P 192.168.1.0/27, 1 successors, FD is 2172416
   via 192.168.1.130 (2172416/5120), Serial0/0/0
P 192.168.1.32/27, 1 successors, FD is 2684416
   via 192.168.1.130 (2684416/2172416), Serial0/0/0
P 192.168.1.96/27, 1 successors, FD is 2681856
   via 192.168.1.130 (2681856/2169856), Serial0/0/0
P 192.168.1.128/27, 1 successors, FD is 2169856
   via Connected, Serial0/0/0
```

Verificación tablas de enrutamiento de los tres routers

- d. Realizar un diagnóstico para comprobar que cada uno de los puntos de la red se puedan ver y tengan conectividad entre sí. Realizar esta prueba desde un host de la red LAN del router CALI, primero a la red de MEDELLIN y luego al servidor.

Analizando el problema se evidencia que la forma más adecuada para poder lograr la verificación es por medio de *ping* en cualquier host de la red de CALI, para estos efectos desde el PC2 a la PC0 que pertenece a la red de MEDELLIN. Para ello se muestra la dirección IP asignada en el PC0 dada por:



Dirección IP del PC0 en la red de MEDELLIN

Por lo que se procede a acceder a la consola del PC2 (perteneciente a la red CALI) y hacer ping, teniendo así:

```
C:\>ping 192.168.1.34

Pinging 192.168.1.34 with 32 bytes of data:

Reply from 192.168.1.34: bytes=32 time=2ms TTL=125
Reply from 192.168.1.34: bytes=32 time=2ms TTL=125
Reply from 192.168.1.34: bytes=32 time=2ms TTL=125
Reply from 192.168.1.34: bytes=32 time=5ms TTL=125

Ping statistics for 192.168.1.34:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 5ms, Average = 2ms
```

Ping exitoso entre PC2 RED Cali y PC0 red MEDELLIN

De la misma forma se toma la dirección IP del servidor de la red BOGOTA para hacer desde el mismo PC2 ping:

```
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=1ms TTL=126
Reply from 192.168.1.3: bytes=32 time=1ms TTL=126
Reply from 192.168.1.3: bytes=32 time=1ms TTL=126
Reply from 192.168.1.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Ping exitoso entre PC2 RED Cali y Servidor red BOGOTA

PARTE 4: CONFIGURACIÓN DE LAS LISTAS DE CONTROL DE ACCESO.

En este momento cualquier usuario de la red tiene acceso a todos sus dispositivos y estaciones de trabajo. El jefe de redes le solicita implementar seguridad en la red. Para esta labor se decide configurar listas de control de acceso (ACL) a los routers.

Las condiciones para crear las ACL son las siguientes:

- a. Cada router debe estar habilitado para establecer conexiones Telnet con los demás routers y tener acceso a cualquier dispositivo en la red.

Vemos que para realizar las conexiones telnet se hace uso de la VLAN1 y se configura la clave como *MEDELLIN* para el primer router teniendo así:

Physical

Config

CLI

Attributes

IOS Command Line Interface

```
MEDELLIN#config t
Enter configuration commands, one per line. End with CNTL/Z.
MEDELLIN(config)#int
MEDELLIN(config)#interface vlan 1
MEDELLIN(config-if)#ip add
MEDELLIN(config-if)#ip address 192.168.1.99 255.255.255.224
% 192.168.1.96 overlaps with Serial0/0/0
MEDELLIN(config-if)#no shu
MEDELLIN(config-if)#no shutdown

MEDELLIN(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, change
state to up

MEDELLIN(config-if)#exit
MEDELLIN(config)#line vty 0 15
MEDELLIN(config-line)#trans
MEDELLIN(config-line)#transport input tel
MEDELLIN(config-line)#transport input telnet
MEDELLIN(config-line)#pass
MEDELLIN(config-line)#password MEDELLIN
MEDELLIN(config-line)#login
MEDELLIN(config-line)#exit
MEDELLIN(config)#
```

Vemos adicionalmente que después de la configuración, se puede desde cualquier computador de otra red acceder mediante telnet, lo primero es realizar ping al puerto y de ser exitoso desde la consola escribir *ping*:

```
C:\>ping 192.168.1.99

Pinging 192.168.1.99 with 32 bytes of data:

Reply from 192.168.1.99: bytes=32 time=3ms TTL=253
Reply from 192.168.1.99: bytes=32 time=2ms TTL=253
Reply from 192.168.1.99: bytes=32 time=2ms TTL=253
Reply from 192.168.1.99: bytes=32 time=2ms TTL=253

Ping statistics for 192.168.1.99:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\>telnet 192.168.1.99
Trying 192.168.1.99 ...Open

User Access Verification

Password:
MEDELLIN>enable
```

Conexión Telnet correcta entre PC2 (red CALI) y router MEDELLIN

BOGOTA

```
Physical  Config  CLI  Attributes
IOS Command Line Interface
BOGOTA>enable
BOGOTA#config t
Enter configuration commands, one per line. End with CNTL/Z.
BOGOTA(config)#int vlan 1
BOGOTA(config-if)#ip add
BOGOTA(config-if)#ip address 192.168.1.98 255.255.255.224
% 192.168.1.96 overlaps with Serial0/0/0
BOGOTA(config-if)#no shut
BOGOTA(config-if)#no shutdown

BOGOTA(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to up

BOGOTA(config-if)#exit
BOGOTA(config)#line vty 0 15
BOGOTA(config-line)#tran
BOGOTA(config-line)#transport input telnet
BOGOTA(config-line)#pass
BOGOTA(config-line)#password BOGOTA
BOGOTA(config-line)#login
BOGOTA(config-line)#exit
BOGOTA(config)#
```

Vemos que se tiene la configuración para Bogotá, y como prueba se toma un PC de la red de Cali o Medellín, así:

```

C:\>telnet 192.168.1.130
Trying 192.168.1.130 ...Open

User Access Verification

Password:
BOGOTA>enable

```

Debido a que en el router BOGOTA se tiene en dos redes que se conectan entre Medellín y Cali, se puede configurar las líneas vty para ambas direcciones IP.

Para el tercer router, la configuración de la ip 192.168.1.131, sería:

```

CALI
-----
Physical  Config  CLI  Attributes
IOS Command Line Interface
CALI>enable
CALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
CALI(config)#int vlan 1
CALI(config-if)#ip add
CALI(config-if)#ip address 192.168.1.131 255.255.255.224
% 192.168.1.128 overlaps with Serial0/0/0
CALI(config-if)#no shut
CALI(config-if)#no shutdown

CALI(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to up

CALI(config-if)#exit
CALI(config)#
CALI(config)#line vty 0 15
CALI(config-line)#trans
CALI(config-line)#transport input telnet
CALI(config-line)#pass
CALI(config-line)#password CALI
CALI(config-line)#login
CALI(config-line)#exit
CALI(config)#

```

Configuración router CALI – telnet

```

C:\>telnet 192.168.1.131
Trying 192.168.1.131 ...Open

User Access Verification

Password:
CALI>

```

Conexión por Telnet

- b. El equipo WS1 y el servidor se encuentran en la subred de administración. Solo el servidor de la subred de administración debe tener acceso a cualquier otro dispositivo en cualquier parte de la red.

```
BOGOTA
```

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```
BOGOTA>enable
BOGOTA#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BOGOTA(config)#int g0/0
BOGOTA(config-if)#no ip acc
BOGOTA(config-if)#no ip access-group ListaDeAcceso out
BOGOTA(config-if)#ipacc
BOGOTA(config-if)#ip acc
BOGOTA(config-if)#ip access-group ListaDeAcceso in
BOGOTA(config-if)#no ip access-group ListaDeAcceso in
BOGOTA(config-if)#ip access-group ListaDeAcceso out
BOGOTA(config-if)#no ip access-group ListaDeAcceso out
BOGOTA(config-if)#ip access-group ListaDeAcceso out
BOGOTA(config-if)#no ip access-group ListaDeAcceso out
BOGOTA(config-if)#ip access-group ListaDeAcceso out
BOGOTA(config-if)#exit
BOGOTA(config)#ip acc
BOGOTA(config)#ip access-list stan
BOGOTA(config)#ip access-list standard Lista
BOGOTA(config-std-nacl)#permit host 192.167.1.2
BOGOTA(config-std-nacl)#permit host 192.167.1.3
BOGOTA(config-std-nacl)#exit
BOGOTA(config)#int g0/0
```

- c. Las estaciones de trabajo en las LAN de MEDELLIN y CALI no deben tener acceso a ningún dispositivo fuera de su subred, excepto para interconectar con el servidor.

```
CALI#config t
Enter configuration commands, one per line.  End with CNTL/Z.
CALI(config)#ip acc
CALI(config)#ip access-list stan
CALI(config)#ip access-list standard ListaCali
CALI(config-std-nacl)#permit host 192.168.1.66
CALI(config-std-nacl)#permit host 192.168.1.67
CALI(config-std-nacl)#deny any
CALI(config-std-nacl)#exit
CALI(config)#int g0/0
CALI(config-if)#ip acc
CALI(config-if)#ip access-group ListaCali out
CALI(config-if)#
```


Lista de control de acceso CALI

```
MEDELLIN
```

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```
MEDELLIN>enable
MEDELLIN#config t
Enter configuration commands, one per line. End with CNTL/Z.
MEDELLIN(config)#ip acc
MEDELLIN(config)#ip access-list stan
MEDELLIN(config)#ip access-list standard ListaMedellin
MEDELLIN(config-std-nacl)#permit host 192.168.1.34
MEDELLIN(config-std-nacl)#permit host 192.168.1.35
MEDELLIN(config-std-nacl)#deny any
MEDELLIN(config-std-nacl)#exit
MEDELLIN(config)#int g0/0
MEDELLIN(config-if)#ip acc
MEDELLIN(config-if)#ip access-group Lista out
MEDELLIN(config-if)#ip access-group ListaMedellin out
MEDELLIN(config-if)#
```

Lista de control de acceso MEDELLIN

PARTE 5: COMPROBACIÓN DE LA RED INSTALADA.

- Se debe probar que la configuración de las listas de acceso fue exitosa, por lo que se procede a realizar ping, así:

```
C:\>ping 192.168.1.34

Pinging 192.168.1.34 with 32 bytes of data:

Reply from 192.168.1.99: Destination host unreachable.
Reply from 192.168.1.99: Destination host unreachable.
Reply from 192.168.1.99: Destination host unreachable.
Reply from 192.168.1.99: Destination host unreachable.

Ping statistics for 192.168.1.34:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Resultado de ping Entre PC2 y PC0

```

C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.130: Destination host unreachable.
Reply from 192.168.1.130: Destination host unreachable.
Reply from 192.168.1.130: Destination host unreachable.
Reply from 192.168.1.130: Destination host unreachable.

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

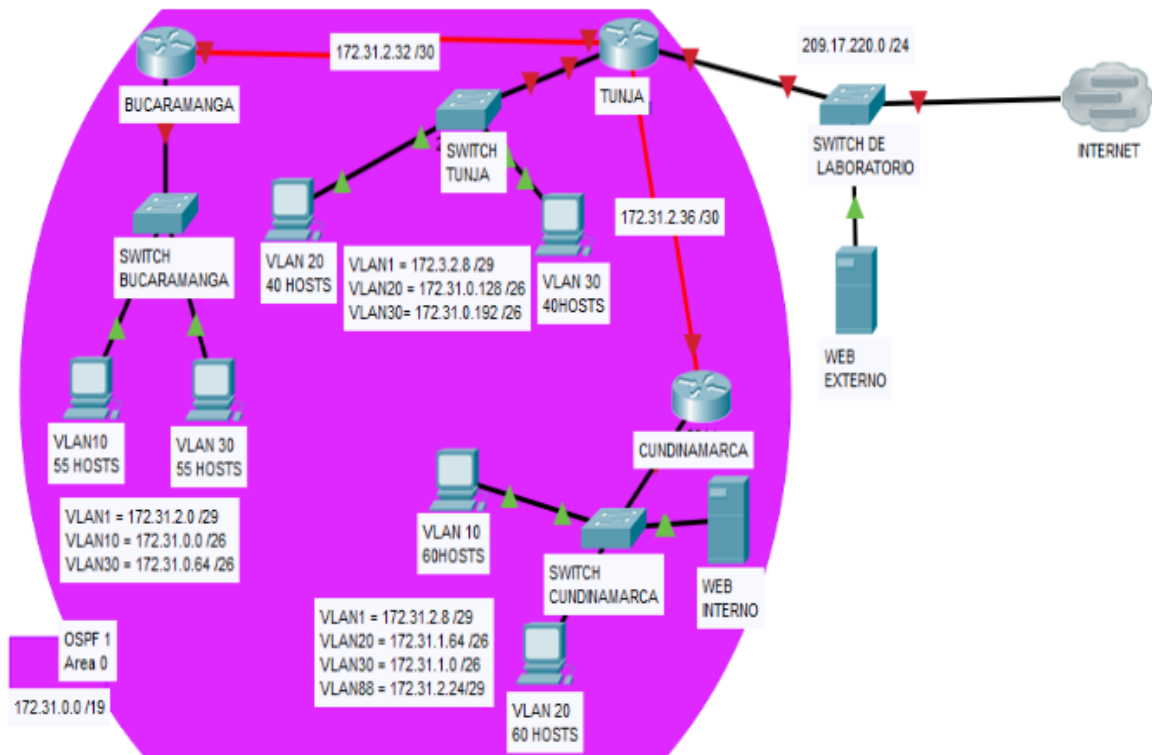
Ping entre PC2 y servidor

b. Comprobar y Completar la siguiente tabla de condiciones de prueba para confirmar el óptimo funcionamiento de la red e.

	ORIGEN	DESTINO	RESULTADO
TELNET	Router MEDELLIN	Router CALI	CORRECTO
	WS_1	Router BOGOTA	CORRECTO
	Servidor	Router CALI	CORRECTO
	Servidor	Router Medellín	CORRECTO
TELNET	Lan del Router MEDELLIN	Router CALI	CORRECTO
	LAN del Router CALI	Router CALI	INCORRECTO
	LAN del Router MEDELLIN	Router Medellín	INCORRECTO
	LAN del Router CALI	Router Medellín	CORRECTO
PING	LAN del Router CALI	WS_1	INCORRECTO
	LAN del Router MEDELLIN	WS_1	INCORRECTO
	LAN del Router MEDELLIN	LAN del Router CALI	INCORRECTO
PING	LAN del Router CALI	Servidor	INCORRECTO
	LAN del Router MEDELLIN	Servidor	INCORRECTO
	Servidor	LAN del Router MEDELLIN	CORRECTO
	Servidor	LAN del Router CALI	CORRECTO
	Router CALI	LAN del Router MEDELLIN	INCORRECTO
	Router Medellín	LAN del Router CALI	INCORRECTO

ESCENARIO 2

Una empresa tiene la conexión a internet en una red Ethernet, lo cual deben adaptarlo para facilitar que sus routers y las redes que incluyen puedan, por esa vía, conectarse a internet, pero empleando las direcciones de la red LAN original. ("IT Networking Career Opportunities - Cisco," n.d.)



DESARROLLO

Los siguientes son los requerimientos necesarios:

1. Todos los routers deberán tener lo siguiente:
 - Configuración básica.
 - Autenticación local con AAA.
 - Cifrado de contraseñas.
 - Un máximo de internos para acceder al router.
 - Máximo tiempo de acceso al detectar ataques.
 - Establezca un servidor TFTP y almacene todos los archivos necesarios de los routers.

Se procede a realizar la configuración solicitada, así:

```
TUNJA
```

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip domain-loo
Router(config)#no ip domain-lookup
Router(config)#enable pass
Router(config)#enable password AAA
Router(config)#line console 0
Router(config-line)#pass
Router(config-line)#password AAA
Router(config-line)#login
Router(config-line)#logging sybch
Router(config-line)#logging synch
Router(config-line)#logging synchronous
Router(config-line)#exit
Router(config)#hostname TUNJA
TUNJA(config)#exit
TUNJA#
%SYS-5-CONFIG_I: Configured from console by console

TUNJA#copy run
TUNJA#copy running-config star
TUNJA#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
```

Configuración básica Tunja

BUCARAMANGA

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip doma
Router(config)#no ip domain-loo
Router(config)#no ip domain-lookup
Router(config)#enable pass
Router(config)#enable password AAA
Router(config)#line consol
Router(config)#line console 0
Router(config-line)#pass
Router(config-line)#password cisco
Router(config-line)#logg
Router(config-line)#logging asy
Router(config-line)#logging synchr
Router(config-line)#logging synchronous
Router(config-line)#exit
Router(config)#ser
Router(config)#service pass
Router(config)#service password-encryption
Router(config)#copy run
Router(config)#copy runn
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

Configuración básica Bucaramanga

CUNDINAMARCA

Physical	Config	CLI	Attributes
----------	--------	-----	------------

IOS Command Line Interface

```

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip domain look-
Router(config)#no ip domain-
Router(config)#no ip domain-loo
Router(config)#no ip domain-lookup
Router(config)#enable pass
Router(config)#enable password AAA
Router(config)#line console 0
Router(config-line)#pass
Router(config-line)#password AAA
Router(config-line)#login
Router(config-line)#logging asynch
Router(config-line)#logging synch
Router(config-line)#logging synchronous
Router(config-line)#exit
Router(config)#servi
Router(config)#service pass
Router(config)#service password-encryption
Router(config)#copy
Router(config)#copy runn
Router(config)#hostname CUNDINAMARCA
CUNDINAMARCA(config)#exit
CUNDINAMARCA#
%SYS-5-CONFIG_I: Configured from console by console

```

Configuración básica Cundinamarca

Inicialmente se debe crear en cada switch las diversas VLAN solicitadas, por ejemplo, el *switch Bucaramanga* tiene tres VLAN las cuales

asignaremos a los puertos físicos, teniendo así:

```
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name VLAN10
Switch(config-vlan)#exit
Switch(config)#int f0/1
Switch(config-if)#switch
Switch(config-if)#switchport mode acc
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#vlan 1
Switch(config-vlan)#name VLAN1
Default VLAN 1 may not have its name changed.
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name VLAN30
Switch(config-vlan)#exit
Switch(config)#int f0/2
Switch(config-if)#switch
Switch(config-if)#switchport mode acc
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
```

Configuración VLAN switch Bucaramanga

Para el switch Tunja tenemos:

```
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 1
Switch(config-vlan)#name VLAN1
Default VLAN 1 may not have its name changed.
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name VLAN20
Switch(config-vlan)#exit
Switch(config)#int f0/2
Switch(config-if)#switch
Switch(config-if)#switchport mode acc
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name VLAN30
Switch(config-vlan)#exit
Switch(config)#int f0/3
Switch(config-if)#switch
Switch(config-if)#switchport mode acc
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
Switch(config)#
```

Configuración VLAN switch Bucaramanga

Por último, para el switch Cundinamarca se tiene:

```
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 1
Switch(config-vlan)#name VLAN1
Default VLAN 1 may not have its name changed.
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name VLAN20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name VLAN30
Switch(config-vlan)#exit
Switch(config)#vlan 88
Switch(config-vlan)#name VLAN88
Switch(config-vlan)#exit
Switch(config)#int f0/2
Switch(config-if)#swit
Switch(config-if)#switchport mode acce
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
```

Creación VLAN 1, 20, 30, 88 en Switch Cundinamarca

Vemos en la imagen anterior que al intentar asignar la VLAN 10 al puerto f0/2 genera error por no existir dicha VLAN, esto debido a un posible error en la gráfica dada por la segunda topología, así que se procede a crear otra VLAN y se asigna según corresponde:

```
Switch(config)#int f0/2
Switch(config-if)#swit
Switch(config-if)#switchport mode acce
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
Switch(config-if)#exit
Switch(config)#vlan 10
Switch(config-vlan)#name VLAN10
Switch(config-vlan)#exit
Switch(config)#int f0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#int f0/3
Switch(config-if)#
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int f0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
```

Asignación VLAN switch Cundinamarca a los puertos

Se procede a configurar el área 0 mediante el protocolo OSPF en cada router como, como paso inicial:

```
CUNDINAMARCA>enable
Password:
Password:
CUNDINAMARCA#config t
Enter configuration commands, one per line. End with CNTL/Z.
CUNDINAMARCA(config)#router ospf 20
OSPF process 20 cannot start. There must be at least one "up" IP
interface
CUNDINAMARCA(config-router)#network
CUNDINAMARCA(config-router)#network 172.31.0.0 0.0.31.255 area 0
CUNDINAMARCA(config-router)#

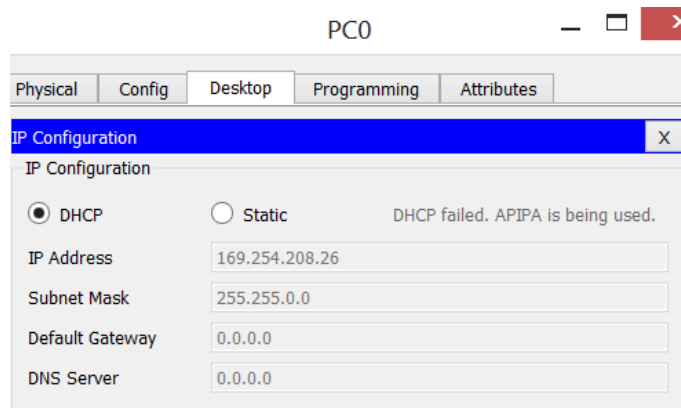
BUCARAMANGA>enable
Password: |
BUCARAMANGA#config t
Enter configuration commands, one per line. End with CNTL/Z.
BUCARAMANGA(config)#router ospf 1
OSPF process 1 cannot start. There must be at least one "up" IP
interface
BUCARAMANGA(config-router)#exit
BUCARAMANGA(config)#router ospf 20
OSPF process 20 cannot start. There must be at least one "up" IP
interface
BUCARAMANGA(config-router)#netw
BUCARAMANGA(config-router)#network 172.31.0.0 0.0.31.255
% Incomplete command.
BUCARAMANGA(config-router)#network 172.31.0.0 0.0.31.255 area 0
BUCARAMANGA(config-router)#

TUNJA>enable
Password:
TUNJA#config t
Enter configuration commands, one per line. End with CNTL/Z.
TUNJA(config)#router ospf 20
OSPF process 20 cannot start. There must be at least one "up" IP
interface
TUNJA(config-router)#192.31.0.0 0.0.31.255 area 0
^
% Invalid input detected at '^' marker.
```

Configuración OSPF de los 3 routers

2. El DHCP deberá proporcionar solo direcciones a los hosts de Bucaramanga y Cundinamarca

Analizando se tiene que para una óptima conectividad con el protocolo DHCP se debe tomar cada host de las redes y que cada uno solicite de manera dinámica las direcciones IP, por lo que se procede a realizar este proceso en cada Host:



ROUTERS BUCARAMANGA

Ahora para el router de Bucaramanga se tiene:

```
BUCARAMANGA>enable
Password:
BUCARAMANGA#config t
Enter configuration commands, one per line. End with CNTL/Z.
BUCARAMANGA(config)#ip dhcp exclu
BUCARAMANGA(config)#ip dhcp excluded-address 172.31.0.129
172.31.0.255|
BUCARAMANGA(config)#ip dhcp pool DHCP_BUCARA
BUCARAMANGA(dhcp-config)#172.31.0.0 255.255.255.192
^
% Invalid input detected at '^' marker.

BUCARAMANGA(dhcp-config)#network 172.31.0.0 255.255.255.192
BUCARAMANGA(dhcp-config)#default
BUCARAMANGA(dhcp-config)#default-router 172.31.0.1
BUCARAMANGA(dhcp-config)#dns-ser
BUCARAMANGA(dhcp-config)#dns-server 8.8.8.8
BUCARAMANGA(dhcp-config)#
```

```
CUNDINAMARCA>enable
Password:
CUNDINAMARCA#config t
Enter configuration commands, one per line. End with CNTL/Z.
CUNDINAMARCA(config)#ip dhcp exclu
CUNDINAMARCA(config)#ip dhcp excluded-address 172.31.1.129
172.31.1.255
CUNDINAMARCA(config)#ip dhcp exclude
CUNDINAMARCA(config)#ip dhcp pool DHCP_CUNDI
CUNDINAMARCA(dhcp-config)#network 172.31.1.0 255.255.255.192
CUNDINAMARCA(dhcp-config)#defau
CUNDINAMARCA(dhcp-config)#default-router 172.31.1.1
CUNDINAMARCA(dhcp-config)#dns-s
CUNDINAMARCA(dhcp-config)#dns-server 8.8.8.8
CUNDINAMARCA(dhcp-config)#
```

3. El web server deberá tener NAT estático y el resto de los equipos de la topología emplearan NAT de sobrecarga (PAT).
4. El enrutamiento deberá tener autenticación.

5. Listas de control de acceso:

- Los hosts de VLAN 20 en Cundinamarca no acceden a internet, solo a la red interna de Tunja.
- Los hosts de VLAN 10 en Cundinamarca si acceden a internet y no a la red interna de Tunja.
- Los hosts de VLAN 30 en Tunja solo acceden a servidores web y ftp de internet.
- Los hosts de VLAN 20 en Tunja solo acceden a la VLAN 20 de Cundinamarca y VLAN 10 de Bucaramanga.
- Los hosts de VLAN 30 de Bucaramanga acceden a internet y a cualquier equipo de VLAN 10.
- Los hosts de VLAN 10 en Bucaramanga acceden a la red de Cundinamarca (VLAN 20) y Tunja (VLAN 20), no internet.
- Los hosts de una VLAN no pueden acceder a los de otra VLAN en una ciudad.
- Solo los hosts de las VLAN administrativas y de la VLAN de servidores tienen acceso a los routers e internet.

6. VLSM: utilizar la dirección 172.31.0.0 /18 para el direccionamiento.

Para proceder a realizar este proceso se deben tener en cuenta determinados aspectos, como lo son:

- Habilitar VLAN en cada switch y permitir su enrutamiento.
- Enrutamiento OSPF con autenticación en cada router.
- Servicio DHCP en el router Tunja, mediante el helper address, para los routers Bucaramanga y Cundinamarca.
- Configuración de NAT estático y de sobrecarga.
- Establecer una lista de control de acceso de acuerdo con los criterios señalados.

➤ Habilitar las opciones en puerto consola y terminal virtual

```
BUCARAMANGA>enable
Password:
BUCARAMANGA#config t
Enter configuration commands, one per line.  End with CNTL/Z.
BUCARAMANGA(config)#int s0/0/0
BUCARAMANGA(config-if)#descri
BUCARAMANGA(config-if)#description conn
BUCARAMANGA(config-if)#description connectio
BUCARAMANGA(config-if)#ip add
BUCARAMANGA(config-if)#ip address 172.31.2.33 255.255.255.252
BUCARAMANGA(config-if)#no shut
BUCARAMANGA(config-if)#clock rate 128000
BUCARAMANGA(config-if)#no shut
BUCARAMANGA(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
BUCARAMANGA(config-if)#
```

R Bucaramanga S0/0/0

```
BUCARAMANGA(config)#int g0/0
BUCARAMANGA(config-if)#ip add
BUCARAMANGA(config-if)#ip address 172.31.0.1 255.255.225.192
Bad mask 0xFFFFE1C0 for address 172.31.0.1
BUCARAMANGA(config-if)#ip address 172.31.0.1 255.255.255.192
BUCARAMANGA(config-if)#no shu
BUCARAMANGA(config-if)#no shutdown

BUCARAMANGA(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up

BUCARAMANGA(config-if)#
```

R Bucaramanga G0/0

```

TUNJA>enable
Password:
TUNJA#config t
Enter configuration commands, one per line. End with CNTL/Z.
TUNJA(config)#int s0/0/0
TUNJA(config-if)#ip add
TUNJA(config-if)#ip address 172.31.2.34 255.255.255.252
TUNJA(config-if)#no shut
TUNJA(config-if)#no shutdown

TUNJA(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

TUNJA(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to up

TUNJA(config-if)#|

```

R Tunja puerto s0/0/0

```

TUNJA(config)#int s0/0/1
TUNJA(config-if)#ip add
TUNJA(config-if)#ip address 172.31.2.36 255.255.255.252
Bad mask /30 for address 172.31.2.36
TUNJA(config-if)#ip address 172.31.2.37 255.255.255.252
TUNJA(config-if)#clock rate 128000
TUNJA(config-if)#no shut
TUNJA(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/1, changed state to down
TUNJA(config-if)#exit
TUNJA(config)#|

```

R Tunja puerto S0/0/1

```
TUNJA(config)#int g0/0
TUNJA(config-if)#ip add
TUNJA(config-if)#ip address 172.31.2.129 255.255.255.192
TUNJA(config-if)#no shut
TUNJA(config-if)#no shutdown

TUNJA(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up

TUNJA(config-if)#|
```

R Tunja puerto G0/0

```
TUNJA(config)#int g0/1
TUNJA(config-if)#209.17.220.1 255.255.255.0
      ^
% Invalid input detected at '^' marker.

TUNJA(config-if)#ip add209.17.220.1 255.255.255.0
TUNJA(config-if)#ip address 209.17.220.1 255.255.255.0
TUNJA(config-if)#no shut
TUNJA(config-if)#no shutdown

TUNJA(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1, changed state to up

TUNJA(config-if)#exit
TUNJA(config)#|
```

R Tunja Puerto G0/1

```
CUNDINAMARCA>enable
Password:
CUNDINAMARCA#config t
Enter configuration commands, one per line.  End with CNTL/Z.
CUNDINAMARCA(config)#int s0/0/0
CUNDINAMARCA(config-if)#ip add
CUNDINAMARCA(config-if)#ip address 172.31.2.38 255.255.255.252
CUNDINAMARCA(config-if)#no shut
CUNDINAMARCA(config-if)#no shutdown

CUNDINAMARCA(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

CUNDINAMARCA(config-if)#|
```

R Cundinamarca S0/0/0

```
CUNDINAMARCA(config)#int g0/0
CUNDINAMARCA(config-if)#ip add
CUNDINAMARCA(config-if)#ip address 172.31.1.1 255.255.255.192
CUNDINAMARCA(config-if)#no shut
CUNDINAMARCA(config-if)#no shutdown

CUNDINAMARCA(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up

CUNDINAMARCA(config-if)#|
```

R Cundinamarca – interfaz G0/0

```
Switch(config)#int vlan 10
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed
state to up

Switch(config-if)#ip add
Switch(config-if)#ip address 172.31.0.2 255.255.255.192
Switch(config-if)#exit
Switch(config)#ip defa
Switch(config)#ip default-gateway 172.31.0.1 255.255.255.192
                                     ^
% Invalid input detected at '^' marker.

Switch(config)#ip default-gateway 172.31.0.1
Switch(config)#
```

Switch Red Bucaramanga – config VLAN 10

```
Switch(config)#int vlan 20
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed
state to up

Switch(config-if)#ip address 172.31.1.66 255.255.255.192
Switch(config-if)#exit
Switch(config)#
```

Switch Red CUNDINAMARCA – config VLAN 20

```

Switch(config-if)#ip address 172.31.1.66 255.255.255.192
Switch(config-if)#exit
Switch(config)#int vlan 10
Switch(config-if)#ip address 172.31.1.2 255.255.255.192
Switch(config-if)#exit
Switch(config)#int vlan 1
Switch(config-if)#ip address 172.31.2.9 255.255.255.248
Switch(config-if)#exit
Switch(config)#vlan 88
Switch(config-vlan)#name VLAN88
Switch(config-vlan)#exit
Switch(config)#int vlan 88
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan88, changed state to up

Switch(config-if)#ip address 172.31.2.25 255.255.255.248
Switch(config-if)#exit
Switch(config)#

```

Switch Red CUNDINAMARCA – config VLAN 10 y 88

```

Switch(config)#int vlan 20
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed
state to up

Switch(config-if)#ip address 172.31.0.129 255.255.255.192
Switch(config-if)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name VLAN30
Switch(config-vlan)#exit
Switch(config)#int vlan 30
Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan30, changed
state to up

Switch(config-if)#172.31.0.193 255.255.255.192
^
% Invalid input detected at '^' marker.

Switch(config-if)#ip address 172.31.0.193 255.255.255.192

```

Switch Red CUNDINAMARCA – config VLAN 20 y 30

CONCLUSIONES

- A lo largo del desarrollo de estas aplicaciones podemos evidenciar una amplia gama de caminos a tomar debido a que la solicitud también fue amplia, adicionalmente es importante mencionar que conocer las líneas de comandos puntuales determinaron correctos resultados.
- Se evidenció a lo largo de las dos prácticas que es necesario poseer conocimientos clave a la hora de realizar este tipo de aplicaciones, pues con ellos logramos obtener los resultados deseados.

AGRADECIMIENTOS

Primeramente doy gracias a Dios por permitirme lograr uno de mis sueños y estar presente siempre en mi vida en segunda instancia a mi familia por ser el apoyo que mi padre celestial permitió en la tierra, por su apoyo incondicional.

BIBLIOGRAFÍA

IT Networking Career Opportunities - Cisco. (n.d.). Retrieved December 14, 2019, from <https://www.cisco.com/c/en/us/training-events/training-certifications/it-networking-career-opportunities.html>