

Índice General

[Resumen Ejecutivo 2](#_Toc473706353)

[Planteamiento del Problema 3](#_Toc473706354)

[Requisitos Clientes 3](#_Toc473706355)

[Diagrama de la topología 4](#_Toc473706356)

[Tabla de Direccionamiento 5](#_Toc473706357)

[Configuración de un servicio aplicado para proteger la red corporativa, denotando los comandos de activación del servicio y verificación de la generación de logs en el enrutador 6](#_Toc473706358)

[Implementación de configuraciones 8](#_Toc473706359)

[Pruebas ejecutadas 16](#_Toc473706360)

[SERVIDOR DE MONITOREO 22](#_Toc473706361)

[Implementación: 22](#_Toc473706362)

[Conclusiones 36](#_Toc473706363)

[Recomendaciones 36](#_Toc473706364)

[Referencias 36](#_Toc473706365)

# **Resumen Ejecutivo**

El presente proyecto (Proyecto de Red Corporativa con Servicios de Seguridad y Mantenimiento Automático) ha sido elaborado en base a los requerimientos solicitados en la: ***Guía del Proyecto de Red Corporativa con Servicios de Seguridad y Mantenimiento Automático*** (vea anexo **A)** de la materia Tecnologías de Redes WAN. En este proyecto se ha aplicado la mayoría de los temas principales tratados a lo largo del curso, también se han incluido nuevos tópicos que han sido investigados, estudiados e implementados y que conforman El ***Componente de Emprendimiento*** que son las características más relevantes de la red, siendo a su vez el valor agregado que distingue a nuestro proyecto .

Todas las configuraciones han sido cuidadosamente seleccionadas e implementadas tratando de mantener los desarrollos y corrientes de TI que actualmente se aplican en la vida real, considerando también las limitaciones que disponemos para la implementación real del proyecto en los Laboratorios.

En el desarrollo de este proyecto la empresa ***Avangers.SA*** el cual es un ISP con sede en Quito-Ecuador. ***Avangers.SA*** en el contexto de este proyecto es ya una empresa establecida con su infraestructura de red y servicios ya implementados y en funcionamiento. De tal manera que para la elaboración del proyecto está basado en el desarrollo del “Diseño de 3 redes corporativas”, que es a su vez el proyecto de ***Avangers.SA*** como respuesta para 3 requerimientos presentados por 3 de sus clientes.

# **Planteamiento del Problema**

**En este proyecto se han utilizado y configurado:**

* 10 enrutadores
* 8 conmutadores
* Frame-relay, MPLS
* OSPF, EIGRP, BGP
* ACL, DHCP, NAT
* Script para el respaldo automático diario de las configuraciones
* Túnel GRE (Componente de Emprendimiento)
* L2TPV3 (Componente de Emprendimiento)
* Un software para el monitoreo de los routers del cliente desde el ISP (Componente de Emprendimiento)

## **Requisitos Clientes**

**Constructora Edebrecht (CE):** Constructora Edebrecht S. A. abrirá una sucursal en la ciudad de Guayaquil por lo cual requerirá de una conexión entre dicha sucursal y la oficina central que se encuentra en la ciudad de Quito. Requiere acceso internet en ambas sucursales.

##### Cooperativa Fedora (CF): Cooperativa Fedora. Ltda. requiere interconectar las sucursales de Naranjito y Milagro. Entre sus necesidades destaca mantener el mismo prefijo de red IP usado en la sucursal de Naranjito para la sucursal de Milagro.

**Corporación Ecuatoriana de GAS (CG):** Corporación Ecuatoriana de Gas Requiere la interconexión de su sucursal en Santa Elena y su oficina central en el COCA (Francisco de Orellana). Requiere acceso internet en ambas sucursales.

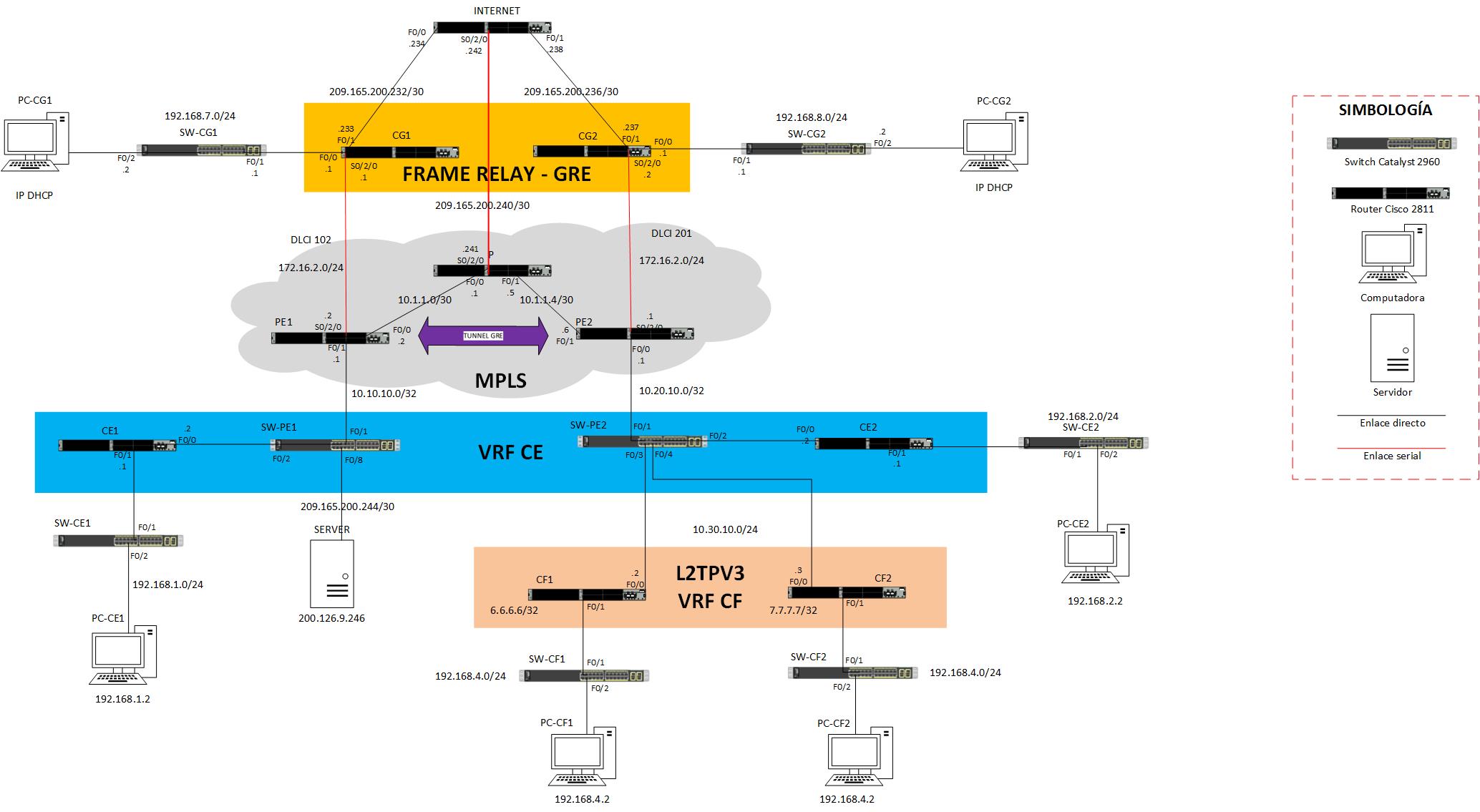
##### Para lograr la satisfacción de nuestros clientes en base a sus requerimientos presentamos las siguientes soluciones de conectividad:

* Para el cliente **CE** La interconexión entre las 2 sucursales se realizara mediante enlaces de fibra óptica con conexión directa a los nodos del Backbone de Avangers, permitiendo acceder a mayores velocidades de transferencia y a una mejor prestación del servicio. También se le asignara al cliente dos direcciones ip públicas (una para cada sucursal), las cuales mediante técnicas de NAT sobrecargado, darán el acceso a internet en ambas sucursales.

##### En el cliente CF la interconexión entre las 2 sucursales también se realizara mediante enlaces de fibra óptica con conexión directa a los nodos del Backbone de Avangers. Para mantener un mismo prefijo de red IP entre ambas sucursales se implementara el Protocolo L2TPV3 el cual permitirá la extensión del dominio de broadcast de Naranjito a Milagro a través de un túnel de capa 2.

* **Dado que nuestra plataforma de fibra óptica no cubre aun toda la región Amazónica, l**a interconexión entre las 2 sucursales se realizará mediante tecnología frame-relay, para lo cual los CPE de Santa Elena y el Coca se conectaran a los PE de **Avangers.,** los mismos **que actuaran como switches frame-relay a la vez** que serán la fuente y el destino de un túnel GRE **que se levantara entre ambos y** permitirá el trafico frame-relay a través de nuestra **plataforma** **de** red mpls.Se le asignara al cliente dos direcciones ip públicas (una para cada sucursal), las cuales mediante técnicas de NAT sobrecargado, darán el acceso a internet en ambas sucursales.

# **Diagrama de la topología**



# **Tabla de Direccionamiento**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dispositivo** | **Interfaz** | **Dirección IP** | **Mascara de subred** | **Gateway P.** |
| **P** | F0/0 | 10.1.1.1 | 255.255.255.252 | N/A |
| F0/0.100 | 10.1.1.9 | 255.255.255.252 | N/A |
| F0/1 | 10.1.1.5 | 255.255.255.252 | N/A |
| F0/1.100 | 10.1.1.13 | 255.255.255.252 | N/A |
| S0/2/0 | 209.165.200.241 | 255.255.255.252 | N/A |
| Lo0 | 1.1.1.1 | 255.255.255.255 | N/A |
| **PE1** | F0/0 | 10.1.1.2 | 255.255.255.252 | N/A |
| F0/0.100 | 10.1.1.10 | 255.255.255.252 | N/A |
| F0/1 | 209.165.200.225 | 255.255.255.252 | N/A |
| F0/1.100 | 10.10.10.1 | 255.255.255.252 | N/A |
| F0/1.99 | 209.165.200.245 | 255.255.255.252 | N/A |
| Lo0 | 2.2.2.2 | 255.255.255.255 | N/A |
| Tunnel1 | 172.16.1.1 | 255.255.255.0 | N/A |
| **PE2** | F0/0 | 209.165.200.229 | 255.255.255.252 | N/A |
| F0/0.100 | 10.20.10.1 | 255.255.255.252 | N/A |
| F0/0.300 | 10.30.10.1 | 255.255.255.0 | N/A |
| F0/1 | 10.1.1.6 | 255.255.255.252 | N/A |
| F0/1.100 | 10.1.1.14 | 255.255.255.252 | N/A |
| Lo0 | 3.3.3.3 | 255.255.255.255 | N/A |
| Tunnel1 | 172.16.1.2 | 255.255.255.0 | N/A |
| **CE1** | F0/0 | 209.165.200.226 | 255.255.255.252 | N/A |
| F0/0.100 | 10.10.10.2 | 255.255.255.252 | N/A |
| F0/1 | 192.168.1.1 | 255.255.255.0 | N/A |
| **CE2** | F0/0 | 209.165.200.230 | 255.255.255.252 | N/A |
| F0/0.100 | 10.20.10.2 | 255.255.255.252 | N/A |
| F0/1 | 192.168.2.1 | 255.255.255.0 | N/A |
| **CF1** | F0/0 | 10.30.10.2 | 255.255.255.0 | N/A |
| Lo0 | 6.6.6.6 | 255.255.255.255 | N/A |
| **CF2** | F0/0 | 10.30.10.3 | 255.255.255.0 | N/A |
| Lo0 | 7.7.7.7 | 255.255.255.255 | N/A |
| **CG1** | F0/0 | 192.168.7.1 | 255.255.255.0 | N/A |
| F0/1 | 209.165.200.233 | 255.255.255.252 | N/A |
| S0/2/0 | 172.16.2.1 | 255.255.255.0 | N/A |
| **CG1** | F0/0 | 192.168.8.1 | 255.255.255.0 | N/A |
| F0/1 | 209.165.200.237 | 255.255.255.252 | N/A |
| S0/2/0 | 172.16.2.2 | 255.255.255.0 | N/A |
| **INTERNET** | F0/0 | 209.165.200.234 | 255.255.255.252 | N/A |
| F0/1 | 209.165.200.238 | 255.255.255.252 | N/A |
| S0/2/0 | 209.165.200.242 | 255.255.255.252 | N/A |
| **PC-CE1** | NIC | 192.168.1.2 | 255.255.255.0 | 192.168.1.1 |
| **PC-CE2** | NIC | 192.168.2.2 | 255.255.255.0 | 192.168.2.1 |
| **PC-CF1** | NIC | 192.168.4.2 | 255.255.255.0 | N/A |
| **PC-CF2** | NIC | 192.168.4.5 | 255.255.255.0 | N/A |

Especificación de la asignación de puertos a los conmutadores

|  |  |
| --- | --- |
| **Puertos** | **Asignaciones** |
| **SW-PE1 F0/1** | Enlace troncal de 802.1Q |
| **SW-PE1 F0/2** | Enlace troncal de 802.1Q |
| **SW-PE1 F0/3** | VLAN 99: SERVIDOR |
| **SW-PE2 F0/1** | Enlace troncal de 802.1Q |
| **SW-PE2 F0/2** | Enlace troncal de 802.1Q |
| **SW-PE2 F0/3** | VLAN 500: CF |
| **SW-PE2 F0/4** | VLAN 500: CF |
| **SW-CF1 F0/1** | Enlace troncal de 802.1Q |
| **SW-CF1 F0/2** | VLAN 500: CF |
| **SW-CF1 F0/3** | VLAN 500: CF |
| **SW-CF2 F0/1** | Enlace troncal de 802.1Q |
| **SW-CF2 F0/2** | VLAN 500: CF |
| **SW-CF2 F0/3** | VLAN 500: CF |

# **Configuración de un servicio aplicado para proteger la red corporativa, denotando los comandos de activación del servicio y verificación de la generación de logs en el enrutador**

En todos los enrutadores se ha activado el registro de logs para la notificación cambios en las configuraciones, fallos, análisis forense de red y auditoría de seguridad.

CE1(config)#logging on

CE1(config)#logging buffered 4096

CE1(config)#end

Para proteger a nuestros clientes de potenciales ataques se ha hecho uso de ACLs las cuales bloquean el paso de todo tráfico que se encuentra en nuestra lista de IPs maliciosas, así mismo en las líneas vty se han aplicado ACLs para que permitan la conexión desde los dispositivos remotos que se consideren necesarios como es el caso del servidor de monitoreo o los routers de otra sucursal.

access-list 80 remark ACL DE LAN CE1

access-list 80 deny 169.254.0.0 0.0.255.255 log

access-list 80 deny 154.10.20.0 0.0.255.255 log

access-list 80 deny 142.10.10.0 0.0.0.255 log

access-list 80 permit any

access-list 1 remark ACL DE LINEAS VTY

access-list 1 permit 10.20.10.2 log

access-list 1 permit 209.165.200.246 log

access-list 1 permit 192.168.2.0 0.0.0.255 log

access-list 2 permit 192.168.1.0 0.0.0.255 log

Para proteger al servidor de monitoreo se ha configurado un firewall con iptables que permita solo el tráfico de entrada de las redes de los clientes, asegurando así los datos almacenados ante posibles ataques.

#!/bin/sh

# Script cortafuegos.sh para la configuración de iptables## #Primero borramos todas las reglas previas que puedan existir

iptables -F

iptables -X

iptables -Z

# Después definimos que las politicas por defecto

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

# Para evitar errores en el sistema, debemos aceptar

# todas las comunicaciones por la interfaz lo (localhost)

iptables -A INPUT -i lo -j ACCEPT

# Aceptamos las comunicaciones que nos interesan

iptables -A INPUT -s 209.165.200.244/30 -j ACCEPT

iptables -A INPUT -s 209.165.200.233 -j ACCEPT

iptables -A INPUT -s 209.165.200.237 -j ACCEPT

iptables -A INPUT -s 10.10.10.2 -j ACCEPT

iptables -A INPUT -s 10.20.10.2 -j ACCEPT

iptables -A INPUT -s 10.30.10.2 -j ACCEPT

iptables -A INPUT -s 10.30.10.3 -j ACCEPT

iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT

iptables -A INPUT -s 192.168.2.0/24 -j ACCEPT

iptables -A INPUT -s 192.168.7.0/24 -j ACCEPT

iptables -A INPUT -s 192.168.8.0/24 -j ACCEPT

iptables -A INPUT -s 192.168.4.0/24 -j ACCEPT

# Comprobamos cómo quedan las reglas

iptables -L –n

# **Implementación de configuraciones**

|  |  |
| --- | --- |
| Dispositivo | Parámetros aplicados |
| P | El router P es el encargado mediante el protocolo LDP del etiquetado en la red mpls, también brinda la sub-interface para la salida a internet para las sucursales del cliente CE, en P se aplicó OSPF para su comunicación con PE1 y PE2. Entre los principales comandos utilizados en P tenemos:  **router ospf 100**  **network 1.1.1.1 0.0.0.0 area 0**  **network 10.1.1.0 0.0.0.7 area 0**  **int f0/0.100**  **encapsulation dot1q 100**  **ip add 10.1.1.9 255.255.255.252**  **exit**  **ip host pe1 2.2.2.2**  Este comando asocia la ip 2.2.2.2 al nombre pe1, lo cual simplifica al momento de hacer ping.  **access-list 1 remark ACL DE LINEAS VTY**  **access-list 1 permit host 10.1.1.2 log**  **access-list 1 permit host 10.1.1.6 log**  **line vty 0 4**  **access-class 1 in**  **exit**  Estos comandos denotan la ACl 1 por default aplicada a las líneas vty, en ella se denota que host puede llegar a P via telnet o ssh. |
| PE1 | El router PE1 brinda una vrf para cada cliente, así como, levanta el túnel gre entre él y PE2, y actúa como frame-relay switching para el cliente CG. Entre sus principales comandos tenemos:  **ip vrf CE**  **rd 65535:100**  **route-target 100:100**  **route-target import 99:99**  **exit**  Este comando es para la creación de la vrf del cliente CE, la segunda línea distingue al cliente, y las 2 últimas líneas importan las tablas de enrutamiento hacia los demás vrf seleccionados.  **int f0/1**  **ip vrf forwarding INTERNET**  **ip address 209.165.200.225 255.255.255.252**  **no shutdown**  Este comando designa a la int f0/1 hacia la vrf designada en este caso la vrf de INTERNET.  **interface tunnel1**  **ip vrf forwarding CG**  **ip address 172.16.1.1 255.255.255.0**  **ip mtu 1400**  **ip tcp adjust-mss 1360**  **tunnel source 2.2.2.2**  **tunnel destination 3.3.3.3**  **exit**  Este comando permite la creación del túnel gre la segunda línea asigna el túnel a la vrf CG la cuarta línea define el tamaño de mtu a 1400, el sexto y séptimos comandos denotan la fuente y destino del túnel.  **frame-relay switching**  **interface Serial 0/2/0**  **encapsulation frame-relay**  **ip vrf forwarding CG**  **clock rate 64000**  **no sh**  **frame-relay intf-type dce**  **frame-relay route 102 interface Tunnel1 112**  Este comando asigna la interfaz serial 0/2/0 a la vrf CG y la encapsula en frame-relay, convirtiendo a PE1 en este caso en un frame-relay switching, la penúltima línea mapea el DLCI 102 hacia el túnel gre. |
| PE2 | El router PE2 brinda una vrf para cada cliente así como, levanta el túnel gre entre él y PE2, y actúa como frame-relay switching para el cliente CG. Entre sus principales comandos tenemos:  **ip vrf CE**  **rd 65535:100**  **route-target 100:100**  **route-target import 99:99**  **exit**  Este comando es para la creación de la vrf del cliente CE, la segunda línea distingue al cliente, y las 2 últimas líneas importan las tablas de enrutamiento hacia los demás vrf seleccionados.  **int f0/0**  **ip vrf forwarding INTERNET**  **ip address 209.165.200.229 255.255.255.252**  **no shutdown**  Este comando designa a la int f0/0 hacia la vrf designada en este caso la vrf de INTERNET.  **interface tunnel1**  **ip vrf forwarding CG**  **ip address 172.16.1.2 255.255.255.0**  **ip mtu 1400**  **ip tcp adjust-mss 1360**  **tunnel source 3.3.3.3**  **tunnel destination 2.2.2.2**  **exit**  Este comando permite la creación del túnel gre la segunda línea asigna el túnel a la vrf CG la cuarta línea define el tamaño de mtu a 1400, el sexto y séptimos comandos denotan la fuente y destino del túnel.  **frame-relay switching**  **interface Serial 0/2/0**  **encapsulation frame-relay**  **ip vrf forwarding CG**  **clock rate 64000**  **no sh**  **frame-relay intf-type dce**  **frame-relay route 201 interface Tunnel1 112**  **exit**  Este comando asigna la interfaz serial 0/2/0 a la vrf CG y la encapsula en frame-relay, convirtiendo a PE1 en este caso en un frame-relay switching, la penúltima línea mapea el DLCI 201 hacia el túnel gre. |
| CE1 | El router CE1 es el router de borde del cliente CE en la sucursal 1, comunica la sucursal CE1 con la sucursal CE2 mediante su conexión con el router PE1, también es el router Nat que permite a los clientes de la sucursal salir hacia internet. Sus principales configuraciones son:  **int f0/0 description CE1-INTERNET**  **ip address 209.165.200.226 255.255.255.252**  **ip nat outside**  **no sh**  En este comando le asigna una ip publica a la interfaz f0/0 para su salida hacia internet, el penúltimo comando denota que será la salida del nat.  **int f0/1**  **description LAN**  **ip address 192.168.1.1 255.255.255.0**  **ip nat inside**  **no keepalive**  **no sh**  **exit**  La cuarta línea indica que f0/1 será la entrada del nat.  **access-list 2 permit 192.168.1.0 0.0.0.255**  **ip nat inside source list 2 interface f0/0 overload**  Este comando es la creación de la ACL que indica que la red de la sucursal CE1 sera convertida por el nat con sobrecarga.  **access-list 80 remark ACL DE LAN CE1**  **access-list 80 deny 169.254.0.0 0.0.255.255 log**  **access-list 80 deny 154.10.20.0 0.0.255.255 log**  **access-list 80 deny 142.10.10.0 0.0.0.255 log**  **access-list 80 permit any**  **int f0/1**  **ip access-group 80 out**  **exit**  Este comando indica la creación de la ACL 80 la cual determina el conjunto de ip que no pueden acceder hacia la lan de la sucursal CE1, dando de esta manera una mayor seguridad a la red. |
| CE2 | El router CE2 es el router de borde del cliente CE en la sucursal 2, comunica la sucursal CE2 con la sucursal CE1 mediante su conexión con el router PE2, también es el router Nat que permite a los clientes de la sucursal salir hacia internet. Sus principales configuraciones son:  **int f0/0**  **description CE2-INTERNET**  **ip address 209.165.200.230 255.255.255.252**  **ip nat outside**  **no sh**  En este comando le asigna una ip publica a la interfaz f0/0 para su salida hacia internet, el penúltimo comando denota que será la salida del nat.  **int f0/1**  **description LAN**  **ip address 192.168.2.1 255.255.255.0**  **ip nat inside**  **no keepalive**  **no sh**  **exit**  La cuarta línea indica que f0/1 sera la entrada del nat.  access-list 2 permit 192.168.2.0 0.0.0.255  ip nat inside source list 2 interface f0/0 overload  Este comando es la creación de la ACL que indica que la red de la sucursal CE2 sera convertida por el nat con sobrecarga.  **access-list 80 remark ACL DE LAN CE2**  **access-list 80 deny 169.254.0.0 0.0.255.255 log**  **access-list 80 deny 154.10.20.0 0.0.255.255 log**  **access-list 80 deny 142.10.10.0 0.0.0.255 log**  **access-list 80 permit any**  **int f0/1**  **ip access-group 80 out**  **exit**  Este comando indica la creación de la ACL 80 la cual determina el conjunto de ip que no pueden acceder hacia la lan de la sucursal CE2, dando de esta manera una mayor seguridad a la red. |
| CF1 | El router CF1 sirve para levantar el túnel l2tpv3 que permitirá expandir el dominio de broadcast entre las sucursales del cliente CF.  **pseudowire-class prueba**  **encapsulation l2tpv3**  **ip local interface loopback 0**  **ip pmtu**  **exit**  **int f0/1**  **no sh**  **exit**  Este comando crea el pseudowire, con nombre prueba, en la segunda línea se encapsula el tipo de protocolo que pasara por el pseudowire en este caso l2tpv3.  **int f0/1.500**  **encapsulation dot1q 500**  **xconnect 7.7.7.7 1 encapsulation l2tpv3 pw-class prueba**  **no sh**  **exit**  El comando xconnect proporciona el destino del túnel en este caso CF2. |
| CF2 | El router CF2 sirve como destino del túnel l2tpv3 que permitirá expandir el dominio de broadcast entre las sucursales del cliente CF.  **pseudowire-class prueba**  **encapsulation l2tpv3**  **ip local interface loopback 0**  **ip pmtu**  **exit**  **int f0/1**  **no sh**  **exit**  Este comando crea el pseudowire, con nombre prueba, en la segunda línea se encapsula el tipo de protocolo que pasara por el pseudowire en este caso l2tpv3.  **int f0/1.500**  **encapsulation dot1q 500**  **xconnect 6.6.6.6 1 encapsulation l2tpv3 pw-class prueba**  **no sh**  **exit**  El comando xconnect proporciona el destino del túnel en este caso CF2. |
| CG1 | El router CG1 es el que contiene el DHCP POOL, es decir el rango de direcciones que asignará como DHCP a los clientes de la sucursal CG1 y CG2, el router CG1 envía a través de la serial s0/2/0 los datos con el DLCI 102 hacia PE1, también aplica NAT con sobrecarga para permitir la salida a Internet de la sucursal CG1.  **ip dhcp excluded-address 192.168.7.1**  **ip dhcp excluded-address 192.168.7.254**  **ip dhcp excluded-address 192.168.8.1**  **ip dhcp excluded-address 192.168.8.254**  **ip dhcp pool CG1**  **network 192.168.7.0 255.255.255.0**  **default-router 192.168.7.1**  **exit**  **ip dhcp pool CG2**  **network 192.168.8.0 255.255.255.0**  **default-router 192.168.8.1**  **exit**  En este comando las 4 primeras líneas excluyen las direcciones que no serán asignadas con en el DHCP en la línea 5 define el DHCP POOL para el cliente CG1, en la línea 6 denota la red asociada a dicho pool, en la línea 9 se define el pool para CG2 y se le asigna la red del cual tomara las direcciones.  **int s0/2/0**  **encapsulation frame-relay**  **no sh**  **exit**  **int s0/2/0**  **ip add 172.16.2.1 255.255.255.0**  **frame-relay map ip 172.16.2.2 102 broadcast**  **exit**  Este comando encapsula la serial 0/2/0 con frame-relay, le asigna una ip, y mapea el destino con el DLCI 102.  **int f0/0**  **ip add 192.168.7.1 255.255.255.0**  **ip nat inside**  **no keepalive**  **no sh**  **exit**  Este comando indica que f0/0 sera la entrada del nat.  **int f0/1**  **ip add 209.165.200.233 255.255.255.252**  **ip nat outside**  **no sh**  En este comando le asigna una ip publica a la interfaz f0/1 para su salida hacia internet, el penúltimo comando denota que será la salida del nat.  **access-list 2 permit 192.168.7.0 0.0.0.255**  **ip nat inside source list 2 interface f0/1 overload**  Este comando es la creación de la ACL que indica que la red de la sucursal CG1 sera convertida por el nat con sobrecarga.  **access-list 80 remark ACL DE LAN CG1**  **access-list 80 deny 169.254.0.0 0.0.255.255 log**  **access-list 80 deny 154.10.20.0 0.0.255.255 log**  **access-list 80 deny 142.10.10.0 0.0.0.255 log**  **access-list 80 permit any**  **int f0/0**  **ip access-group 80 out**  **exit**  Este comando indica la creación de la ACL 80 la cual determina el conjunto de ip que no pueden acceder hacia la lan de la sucursal CG1, dando de esta manera una mayor seguridad a la red. |
| CG2 | El router CG2 es el DHCP relay, es decir es el que enviará hacia CG1 las peticiones DHCP de los clientes de la sucursal CG2, el router CG2 envía a través de la interfaz serial s0/2/0 los datos con el DLCI 201 hacia PE2, también aplica NAT con sobrecarga para permitir la salida a Internet de la sucursal CG2.  **int s0/2/0**  **encapsulation frame-relay**  **no sh**  **exit**  **int s0/2/0**  **ip add 172.16.2.1 255.255.255.0**  **frame-relay map ip 172.16.2.2 102 broadcast**  **exit**  Este comando encapsula la interfaz serial 0/2/0 con frame-relay, le asigna una ip, y mapea el destino con el DLCI 102.  **int f0/0**  **ip add 192.168.8.1 255.255.255.0**  **ip nat inside**  **no keepalive**  **ip helper-address 172.16.2.1**  **no sh**  **exit**  Este comando indica que f0/0 enviara las peticiones de DHCP hacia CG1, también será la entrada del nat.  **int f0/1**  **ip add 209.165.200.237 255.255.255.252**  **ip nat outside**  **no sh**  En este comando le asigna una ip publica a la interfaz f0/1 para su salida hacia internet, el penúltimo comando denota que será la salida del nat.  **access-list 2 permit 192.168.8.0 0.0.0.255**  **ip nat inside source list 2 interface f0/1 overload**  Este comando es la creación de la ACL que indica que la red de la sucursal CG2 sera convertida por el nat con sobrecarga.  **access-list 80 remark ACL DE LAN CG2**  **access-list 80 deny 169.254.0.0 0.0.255.255 log**  **access-list 80 deny 154.10.20.0 0.0.255.255 log**  **access-list 80 deny 142.10.10.0 0.0.0.255 log**  **access-list 80 permit any**  **int f0/0**  **ip access-group 80 out**  **exit**  Este comando indica la creación de la ACL 80 la cual determina el conjunto de ip que no pueden acceder hacia la lan de la sucursal CG2, dando de esta manera una mayor seguridad a la red. |

# **Pruebas ejecutadas**

Comprobamos el protocolo mpls en cada uno de los enrutadores (P, PE1 y PE2) con el comando show mpls ip binding

**Enrutador P**

P#show mpls ip binding

1.1.1.1/32

in label: imp-null

out label: 17 lsr: 3.3.3.3:0

out label: 17 lsr: 2.2.2.2:0

2.2.2.2/32

in label: 21

out label: imp-null lsr: 2.2.2.2:0 inuse

out label: 18 lsr: 3.3.3.3:0

3.3.3.3/32

in label: 22

out label: imp-null lsr: 3.3.3.3:0 inuse

out label: 18 lsr: 2.2.2.2:0

10.1.1.0/30

in label: imp-null

out label: 16 lsr: 3.3.3.3:0

out label: imp-null lsr: 2.2.2.2:0

10.1.1.4/30

in label: imp-null

out label: imp-null lsr: 3.3.3.3:0

out label: 16 lsr: 2.2.2.2:0

10.1.1.8/30

in label: imp-null

10.1.1.12/30

in label: imp-null

209.165.200.224/30

in label: 16

209.165.200.228/30

in label: 17

209.165.200.232/30

in label: 18

209.165.200.236/30

in label: 19

209.165.200.240/30

in label: imp-null

209.165.200.244/30

in label: 20

**Enrutador PE1**

PE1#show mpls ip binding

1.1.1.1/32

in label: 17

out label: imp-null lsr: 1.1.1.1:0 inuse

2.2.2.2/32

in label: imp-null

out label: 21 lsr: 1.1.1.1:0

3.3.3.3/32

in label: 18

out label: 22 lsr: 1.1.1.1:0 inuse

10.1.1.0/30

in label: imp-null

out label: imp-null lsr: 1.1.1.1:0

10.1.1.4/30

in label: 16

out label: imp-null lsr: 1.1.1.1:0 inuse

10.1.1.8/30

out label: imp-null lsr: 1.1.1.1:0

10.1.1.12/30

out label: imp-null lsr: 1.1.1.1:0

209.165.200.224/30

out label: 16 lsr: 1.1.1.1:0

209.165.200.228/30

out label: 17 lsr: 1.1.1.1:0

209.165.200.232/30

out label: 18 lsr: 1.1.1.1:0

209.165.200.236/30

out label: 19 lsr: 1.1.1.1:0

209.165.200.240/30

out label: imp-null lsr: 1.1.1.1:0

209.165.200.244/30

out label: 20 lsr: 1.1.1.1:0

**Enrutador PE2**

PE2#show mpls ip binding

1.1.1.1/32

in label: 17

out label: imp-null lsr: 1.1.1.1:0 inuse

2.2.2.2/32

in label: 18

out label: 21 lsr: 1.1.1.1:0 inuse

3.3.3.3/32

in label: imp-null

out label: 22 lsr: 1.1.1.1:0

10.1.1.0/30

in label: 16

out label: imp-null lsr: 1.1.1.1:0 inuse

10.1.1.4/30

in label: imp-null

out label: imp-null lsr: 1.1.1.1:0

10.1.1.8/30

out label: imp-null lsr: 1.1.1.1:0

10.1.1.12/30

out label: imp-null lsr: 1.1.1.1:0

209.165.200.224/30

out label: 16 lsr: 1.1.1.1:0

209.165.200.228/30

out label: 17 lsr: 1.1.1.1:0

209.165.200.232/30

out label: 18 lsr: 1.1.1.1:0

209.165.200.236/30

out label: 19 lsr: 1.1.1.1:0

209.165.200.240/30

out label: imp-null lsr: 1.1.1.1:0

209.165.200.244/30

out label: 20 lsr: 1.1.1.1:0

Para verificar el tunel l2pt entre los enrutadores CF1 y CF2, emitimos el siguiente comando:

CF1#show l2tun tunnel l2tp all

L2TP Tunnel Information **Total tunnels 1 sessions 1**

Tunnel id 1367 is up, remote id is 8144, 1 active sessions

Tunnel state is established, time since change 00:04:39

Tunnel transport is IP (115)

**Remote tunnel name is CF2**

**Internet Address 7.7.7.7, port 0**

**Local tunnel name is CF1**

**Internet Address 6.6.6.6, port 0**

Tunnel domain unknown

VPDN group for tunnel is not available

L2TP class for tunnel is l2tp\_default\_class

0 packets sent, 0 received

0 bytes sent, 0 received

Last clearing of "show vpdn" counters never

Control Ns 8, Nr 6

Local RWS 800 (default), Remote RWS 800 (max)

Tunnel PMTU checking enabled

Retransmission time 1, max 1 seconds

Unsent queuesize 0, max 0

Resend queuesize 0, max 2

Total resends 0, ZLB ACKs sent 4

Current nosession queue check 0 of 5

Retransmit time distribution: 0 0 0 0 0 0 0 0 0

Sessions disconnected due to lack of resources 0

CF2#show l2tun tunnel all

L2TP Tunnel Information **Total tunnels 1 sessions 1**

Tunnel id 8144 is up, remote id is 1367, 1 active sessions

Tunnel state is established, time since change 00:05:48

Tunnel transport is IP (115)

**Remote tunnel name is CF1**

**Internet Address 6.6.6.6, port 0**

**Local tunnel name is CF2**

**Internet Address 7.7.7.7, port 0**

Tunnel domain unknown

VPDN group for tunnel is not available

L2TP class for tunnel is l2tp\_default\_class

0 packets sent, 0 received

0 bytes sent, 0 received

Last clearing of "show vpdn" counters never

Control Ns 7, Nr 11

Local RWS 800 (default), Remote RWS 800 (max)

Tunnel PMTU checking disabled

Retransmission time 1, max 1 seconds

Unsent queuesize 0, max 0

Resend queuesize 0, max 1

Total resends 0, ZLB ACKs sent 10

Current nosession queue check 0 of 5

Retransmit time distribution: 0 0 0 0 0 0 0 0 0

Sessions disconnected due to lack of resources 0

Para verificar el protocol frame-relay utilizado entre los enrutadores CG1 y CG2, emitimos el siguiente comando:

CG1#show frame-relay map

Serial1/0 (up): **ip 172.16.2.2 dlci 102**(0x66,0x1860), static,

broadcast,

CISCO, status defined, inactive

CG2#show frame-relay map

Serial1/0 (up**): ip 172.16.2.1 dlci 201**(0xC9,0x3090), static,

broadcast,

CISCO, status defined, active

Para verificar la traducción nat configurada en el enrutador CE1 para que el host conectado al conmutador SW-CE1 pueda navegar a internet, emitimos un ping desde dicha pc hacia el enrutador INTERNET, y una vez hecho esto verificamos el nateo con el comando show ip nat translations en el enrutador CE1.

PC5> ping 209.165.200.242

84 bytes from 209.165.200.242 icmp\_seq=1 ttl=252 time=68.089 ms

84 bytes from 209.165.200.242 icmp\_seq=2 ttl=252 time=60.388 ms

84 bytes from 209.165.200.242 icmp\_seq=3 ttl=252 time=64.078 ms

84 bytes from 209.165.200.242 icmp\_seq=4 ttl=252 time=39.810 ms

84 bytes from 209.165.200.242 icmp\_seq=5 ttl=252 time=56.125 ms

CE1#show ip nat translations

Pro Inside global Inside local Outside local Outside global

icmp 209.165.200.226:29670 192.168.1.2:29670 209.165.200.242:29670 209.165.200.242:29670

icmp 209.165.200.226:30182 192.168.1.2:30182 209.165.200.242:30182 209.165.200.242:30182

icmp 209.165.200.226:30694 192.168.1.2:30694 209.165.200.242:30694 209.165.200.242:30694

icmp 209.165.200.226:30950 192.168.1.2:30950 209.165.200.242:30950 209.165.200.242:30950

icmp 209.165.200.226:31206 192.168.1.2:31206 209.165.200.242:31206 209.165.200.242:31206

icmp 209.165.200.226:31974 192.168.1.2:31974 209.165.200.242:31974 209.165.200.242:31974

icmp 209.165.200.226:32230 192.168.1.2:32230 209.165.200.242:32230 209.165.200.242:32230

icmp 209.165.200.226:32742 192.168.1.2:32742 209.165.200.242:32742 209.165.200.242:32742

icmp 209.165.200.226:32998 192.168.1.2:32998 209.165.200.242:32998 209.165.200.242:32998

icmp 209.165.200.226:33254 192.168.1.2:33254 209.165.200.242:33254 209.165.200.242:33254

Asimismo, para verificar la traducción nat configurada en el enrutador CE2 para que el host conectado al conmutador SW-CE2 pueda navegar a internet, emitimos un ping desde dicha pc hacia el enrutador INTERNET, y una vez hecho esto verificamos el nateo con el comando show ip nat translations en el enrutador CE2.

PC6> ping 209.165.200.242

84 bytes from 209.165.200.242 icmp\_seq=1 ttl=252 time=56.075 ms

84 bytes from 209.165.200.242 icmp\_seq=2 ttl=252 time=52.047 ms

84 bytes from 209.165.200.242 icmp\_seq=3 ttl=252 time=80.225 ms

84 bytes from 209.165.200.242 icmp\_seq=4 ttl=252 time=56.076 ms

84 bytes from 209.165.200.242 icmp\_seq=5 ttl=252 time=88.108 ms

CE2#show ip nat translations

Pro Inside global Inside local Outside local Outside global

icmp 209.165.200.230:18664 192.168.2.2:18664 209.165.200.242:18664 209.165.200.242:18664

icmp 209.165.200.230:19176 192.168.2.2:19176 209.165.200.242:19176 209.165.200.242:19176

icmp 209.165.200.230:19688 192.168.2.2:19688 209.165.200.242:19688 209.165.200.242:19688

icmp 209.165.200.230:19944 192.168.2.2:19944 209.165.200.242:19944 209.165.200.242:19944

icmp 209.165.200.230:20200 192.168.2.2:20200 209.165.200.242:20200 209.165.200.242:20200

icmp 209.165.200.230:20712 192.168.2.2:20712 209.165.200.242:20712 209.165.200.242:20712

icmp 209.165.200.230:20968 192.168.2.2:20968 209.165.200.242:20968 209.165.200.242:20968

icmp 209.165.200.230:21224 192.168.2.2:21224 209.165.200.242:21224 209.165.200.242:21224

icmp 209.165.200.230:21480 192.168.2.2:21480 209.165.200.242:21480 209.165.200.242:21480

icmp 209.165.200.230:21992 192.168.2.2:21992 209.165.200.242:21992 209.165.200.242:21992

Para verificar el servidor DHCP, emitimos el commando show ip dhcp binding en el enrutador CG1, que es el encargado de asignar direcciones ip automáticas a los clientes conectados tanto en el conmutador SW-CG1 como en el SW-CG2.

CG1#show ip dhcp binding

Bindings from all pools not associated with VRF:

IP address Client-ID/ Lease expiration Type

Hardware address/

User name

192.168.7.3 0100.5079.6668.06 Mar 02 2002 12:26 AM Automatic

192.168.8.3 0100.5079.6668.07 Mar 02 2002 12:27 AM Automatic

En cada uno de los clientes se han aplicado listas de control de acceso, para verificar las mismas, emitimos en comando show ip access-lists.

**CE1#show ip access-lists**

Standard IP access list 1

10 permit 10.20.10.2 log

20 permit 209.165.200.246 log

30 permit 192.168.2.0, wildcard bits 0.0.0.255 log

Standard IP access list 2

10 permit 192.168.1.0, wildcard bits 0.0.0.255 (10 matches)

Standard IP access list 80

10 deny 169.254.0.0, wildcard bits 0.0.255.255 log

20 deny 154.10.0.0, wildcard bits 0.0.255.255 log

30 deny 142.10.10.0, wildcard bits 0.0.0.255 log

40 permit any (8 matches)

**CE2#show ip access-lists**

Standard IP access list 1

10 permit 10.10.10.2 log

20 permit 209.165.200.246 log

30 permit 192.168.1.0, wildcard bits 0.0.0.255 log

Standard IP access list 2

10 permit 192.168.2.0, wildcard bits 0.0.0.255 (10 matches)

Standard IP access list 80

10 deny 169.254.0.0, wildcard bits 0.0.255.255 log

20 deny 154.10.0.0, wildcard bits 0.0.255.255 log

30 deny 142.10.10.0, wildcard bits 0.0.0.255 log

40 permit any (8 matches)

**CG1#show ip access-lists**

Standard IP access list 1

10 permit 209.165.200.246 log

20 permit 172.16.2.2 log

30 permit 192.168.8.0, wildcard bits 0.0.0.255 log

Standard IP access list 2

10 permit 192.168.7.0, wildcard bits 0.0.0.255 (8 matches)

Standard IP access list 80

10 deny 169.254.0.0, wildcard bits 0.0.255.255 log

20 deny 154.10.0.0, wildcard bits 0.0.255.255 log

30 deny 142.10.10.0, wildcard bits 0.0.0.255 log

40 permit any

**CG2#show ip access-lists**

Standard IP access list 1

10 permit 209.165.200.246 log

20 permit 172.16.2.1 log

30 permit 192.168.7.0, wildcard bits 0.0.0.255 log

Standard IP access list 2

10 permit 192.168.8.0, wildcard bits 0.0.0.255 (5 matches)

Standard IP access list 80

10 deny 169.254.0.0, wildcard bits 0.0.255.255 log

20 deny 154.10.0.0, wildcard bits 0.0.255.255 log

30 deny 142.10.10.0, wildcard bits 0.0.0.255 log

40 permit any (3 matches)

# **SERVIDOR DE MONITOREO**

Como servicio adicional nosotros ofrecemos a nuestros clientes el monitoreo de los enrutadores que se encuentran conectados directamente a nuestro ISP, en el cual, por medio de scripts obtenemos información de parámetros importantes como la versión del dispositivo, el estado de las interfaces, la tabla de enrutamiento y el registro de logs. Así mismo ofrecemos a nuestros clientes el respaldo diario de las configuraciones de los enrutadores, para que en caso de que, por una mala configuración, y este deje de trabajar correctamente, sea posible contar con la última configuración en buen estado logrando así continuar con las actividades rutinarias de la empresa y evitar pérdida de tiempo a sus clientes, lo cual representa dinero.

## Implementación:

Para recaudar la información obtenida de los enrutadores se creó una base de datos en la cual se almacenan los clientes con sus respectivos enrutadores y la dirección en donde se encuentran almacenados los archivos de los parámetros mencionados anteriormente. Se hizo uso del siguiente script para crear la base datos e ingresar los valores:

#!/bin/bash

#Script createDB.sh crea la base de datos para almacenar información acerca de los routers de los clientes.

#CREA LA BASE DE DATOS AVENGERSDB

echo "create database avengersDB;" | mysql -uroot -proot

#Crea la tabla clientes para almacenar datos de nuestros clientes.

echo "create table clientes(

ID\_C INT NOT NULL,

NAME VARCHAR(50),

PRIMARY KEY(ID\_C) );" | mysql -uroot -proot avengersDB

#Crea la tabla routers para almacenar las rutas de los archivos con los parámetros obtenidos mediante el script getInfoRouter.pl.

echo "create table routers(

ID INT NOT NULL AUTO\_INCREMENT,

ID\_CLIENT INT NOT NULL,

HOSTNAME VARCHAR(20) NOT NULL,

IP VARCHAR(20),

USER VARCHAR(50),

PASSWD VARCHAR(50),

PATH\_VERSION VARCHAR(100),

PATH\_INTERFACES VARCHAR(100),

PATH\_ROUTING VARCHAR(100),

PATH\_LOGS VARCHAR(100),

PRIMARY KEY (ID),

FOREIGN KEY (ID\_CLIENT) REFERENCES clientes(ID\_C) );" | mysql -uroot -proot avengersDB

#Crea la tabla configs para almacenar las rutas de los archivos de configuración obtenidos mediante el script configsBackup.pl.

echo "create table configs(

ID int not null auto\_increment,

ID\_DEV int not null,

fecha DATE,

PATH varchar(200),

primary key(ID),

foreign key(ID\_DEV) references routers(ID));" | mysql -uroot -proot avengersDB

Para obtener información de los enrutadores tales como la versión, el estado de las interfaces, la tabla de enrutamiento y los logs, se hizo uso del siguiente script escrito en lenguaje perl haciendo uso de las librerías Net::Telnet::Cisco para iniciar sesión remotamente en los enrutadores y la librería DBI y DBD::mysql para conectarse a la base de datos.

#!/usr/bin/perl

#Script getInfoRouter.pl obtiene la infomación de los enrutadores tales como la versión, el estado de las interfaces, la tabla de enrutamiento y los logs. use DBI;

use DBD::mysql;

use Time::Piece;

use Net::Telnet::Cisco;

#Setteo de variables iniciales

my $date = localtime->strftime('%m-%d-%Y-%R');

$database = "avengersDB";

$tablename = "routers";

$user = "root";

$password = "root";

$tftp\_server = "209.165.200.246";

$dsn = "dbi:mysql:$database:localhost:3306"; # data source name

#Conexión a la base de datos

#PERL DBI connect

$dbi\_connect = DBI->connect( $dsn, $user, $password) or die "No se puede conectar: $DBI::errstr\n";

$query\_client = "SELECT \* FROM clientes;";

$query\_hclient = $dbi\_connect->prepare( $query\_client );

$query\_hclient->execute();

$query\_hclient->bind\_columns(undef, \$idCliente, \$nameCliente );

while ( $query\_hclient->fetch()){

#Preparar el query

$query\_select = "SELECT HOSTNAME,IP,USER,PASSWD FROM $tablename where routers.ID\_CLIENT=$idCliente ;";

$query\_handle = $dbi\_connect->prepare( $query\_select );

#Ejecutar el query

$query\_handle->execute();

#Enlazar las columnas a las variables

$query\_handle->bind\_columns(undef, \$name, \$ip, \$user, \$pass );

#Recorrer datos obtenidos

while ( $query\_handle->fetch()){

print "$name - $ip - $user - $pass \n";

#Inicia sesion en el enrutador remotamente

$session = Net::Telnet::Cisco->new(Host => $ip );

$session->login ( $user ,$pass );

$path\_data = "/home/soporte/Netmon/$nameCliente/$name";

#ejecuta los comandos y almacena los show en su variable correspondiente.

@output = $session->cmd( "show version" );

@outInterfaces = $session->cmd( "show ip int brief" );

@outIproute = $session->cmd( "show ip route" );

@outLogs = $session->cmd( "show logging | begin Log Buffer" );

#obtiene el show y lo guardar en un archivo.

open(SHVERSION, ">$path\_data/show\_version\_$date.txt");

foreach $out ( @output ){

print SHVERSION $out;

}

open(INTERFACES, ">$path\_data/show\_interfaces\_$date.txt");

foreach $out ( @outInterfaces ){

print INTERFACES $out;

}

open(IPROUTE, ">$path\_data/show\_iproute\_$date.txt");

foreach $out ( @outIproute ){

print IPROUTE $out;

}

open(LOGS, ">$path\_data/show\_logs\_$date.txt");

foreach $out ( @outLogs ){

print LOGS $out;

}

close(SHVERSION);

close(INTERFACES);

close(IPROUTE);

close(LOGS);

$query\_update = "UPDATE routers SET PATH\_VERSION='$path\_data/show\_version\_$date.txt' , PATH\_INTERFACES= '$path\_data/show\_interfaces\_$date.txt' , PATH\_ROUTING='$path\_data/show\_iproute\_$date.txt' , PATH\_LOGS='$path\_data/show\_logs\_$date.txt' WHERE HOSTNAME='$name';"; ;

$query\_hupdate = $dbi\_connect->prepare( $query\_update );

#Ejecutar el query

$query\_hupdate->execute();

$session->close;

}

}

$query\_hclient-> finish();

$query\_handle -> finish();

$dbi\_connect->disconnect;

Para realizar los respaldos diarios de las configuraciones de los equipos se realizó un script, llamado configsBackup.pl el cual almacena en un archivo la información de los running-config obtenido para cada enrutador. Este script está configurado de tal manera que se ejecute automáticamente a las 23 horas de cada día, para esto se hizo uso del crontab. El script está escrito en lenguaje perl haciendo uso de las librerías Net::Telnet::Cisco para iniciar sesión remotamente en los enrutadores y la librería DBI y DBD:mysql para conectarse a la base de datos.

#!/usr/bin/perl

use DBI;

use DBD::mysql;

use Time::Piece;

use Net::Telnet::Cisco;

#Setteo de variables iniciales

my $date = localtime->strftime('%Y-%m-%d-%R');

$database = "avengersDB";

$tablename = "routers";

$user = "root";

$password = "root";

$tftp\_server = "209.165.200.246";

$dsn = "dbi:mysql:$database:localhost:3306"; # data source name

#Conexión a la base de datos

#PERL DBI connect

$dbi\_connect = DBI->connect( $dsn, $user, $password) or die "No se puede conectar: $DBI::errstr\n";

$query\_client = "SELECT \* FROM clientes;";

$query\_hclient = $dbi\_connect->prepare( $query\_client );

$query\_hclient->execute();

$query\_hclient->bind\_columns(undef, \$idCliente, \$nameCliente );

while ( $query\_hclient->fetch()){

#Preparar el query

$query\_select = "SELECT ID,HOSTNAME,IP,USER,PASSWD FROM $tablename where routers.ID\_CLIENT=$idCliente ;";

$query\_handle = $dbi\_connect->prepare( $query\_select );

#Ejecutar el query

$query\_handle->execute();

#Enlazar las columnas a las variables

$query\_handle->bind\_columns(undef, \$id, \$name, \$ip, \$user, \$pass );

#Recorrer datos obtenidos

while ( $query\_handle->fetch()){

print "$id - $name - $ip - $user - $pass \n";

$session = Net::Telnet::Cisco->new(Host => $ip );

$session->login ( $user ,$pass );

$pathConfig = "tftp://$tftp\_server/RoutersConfig/$name-$date-config";

$pathSaveScript = "/srv/tftp/RoutersConfig/$name-$date-config";

#realiza un backup del show running.

$session->cmd( "copy running-config $pathConfig\n\n\n" );

$session->close;

#inserta un nuevo registro de la ruta del archivo de configuración

$query\_insert = "insert into configs (ID\_DEV, fecha, PATH) values ('$id' ,'$date' , '$pathSaveScript')";

$query\_hinsert = $dbi\_connect->prepare( $query\_insert );

$query\_hinsert->execute();

$query\_hinsert->finish();

}

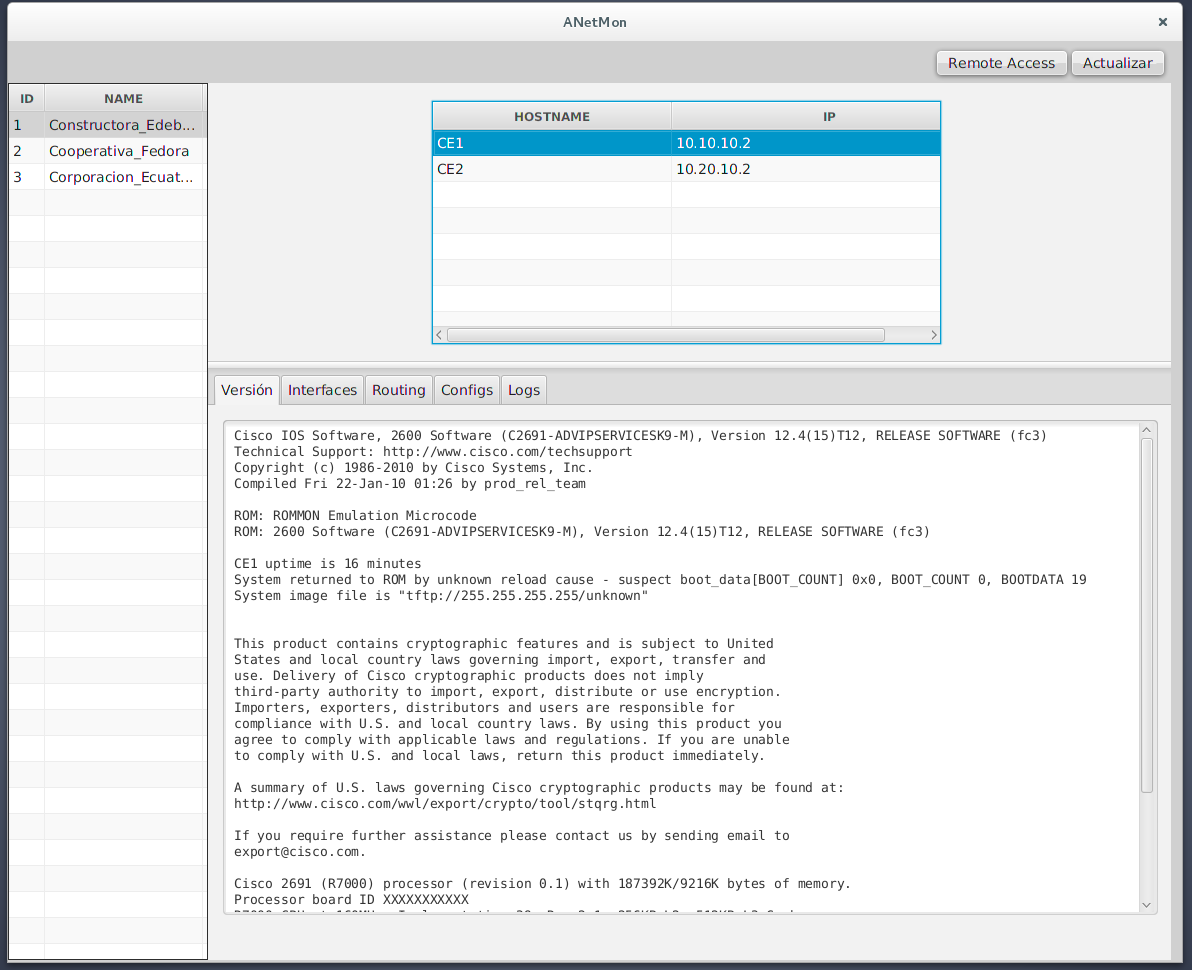
}

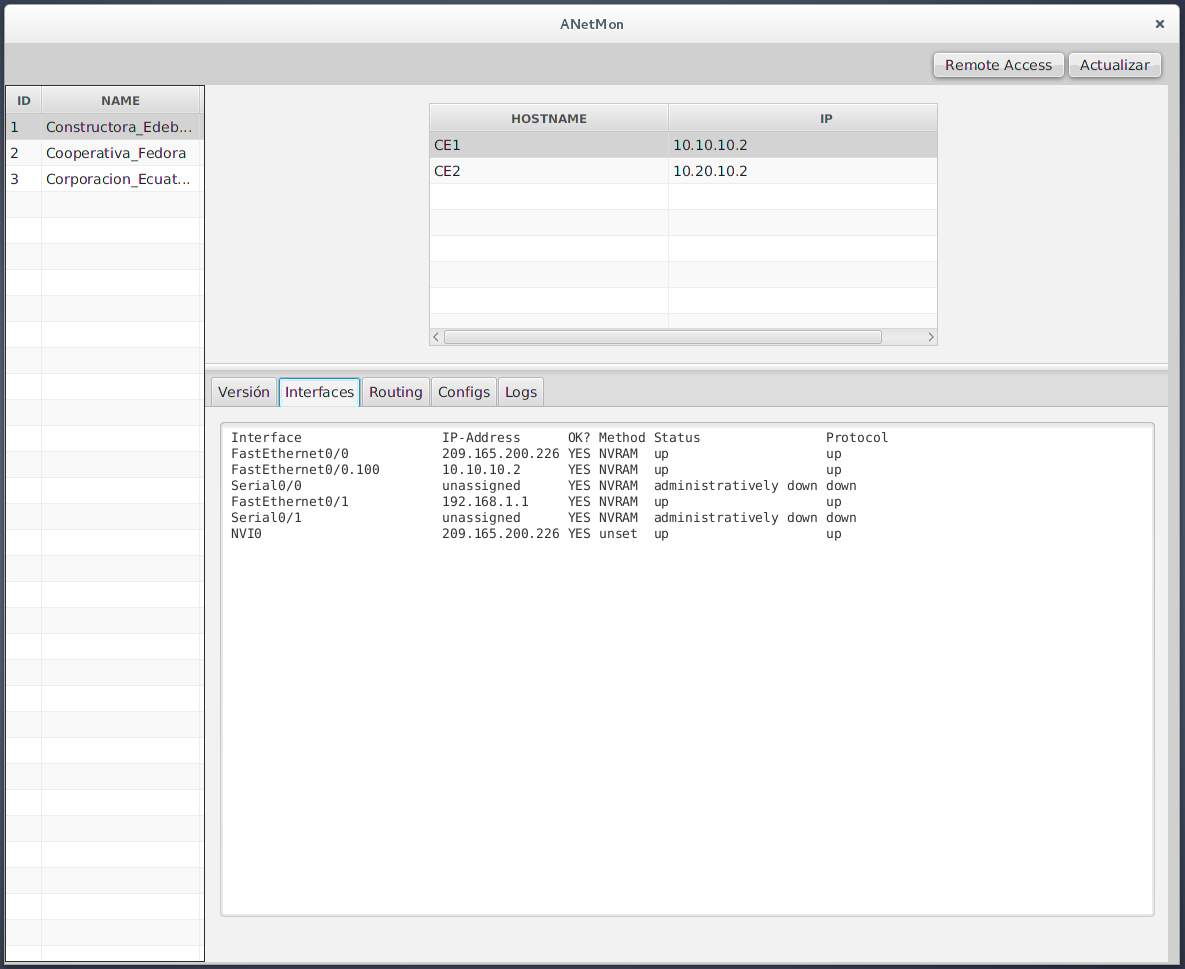
$query\_hclient-> finish();

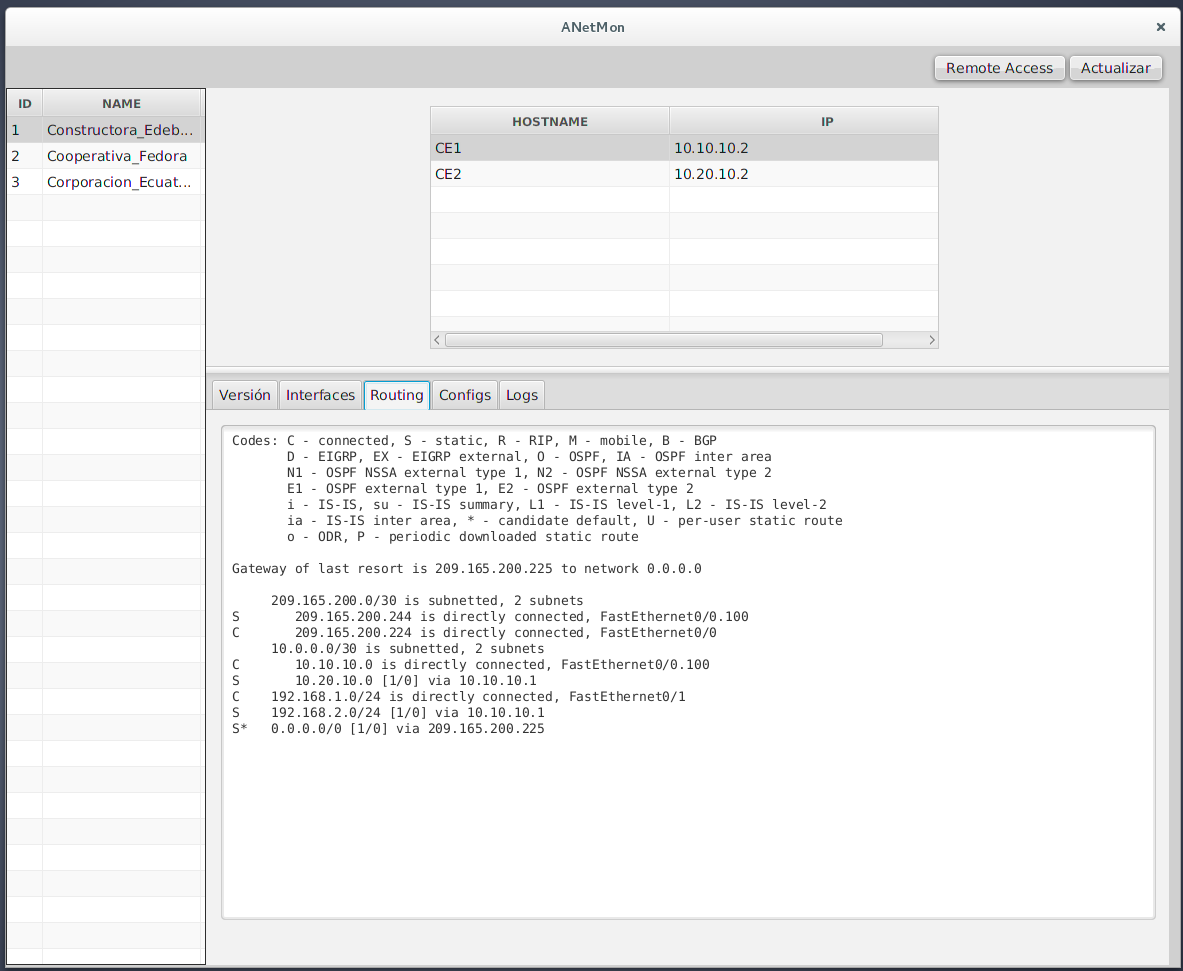
$query\_handle -> finish();

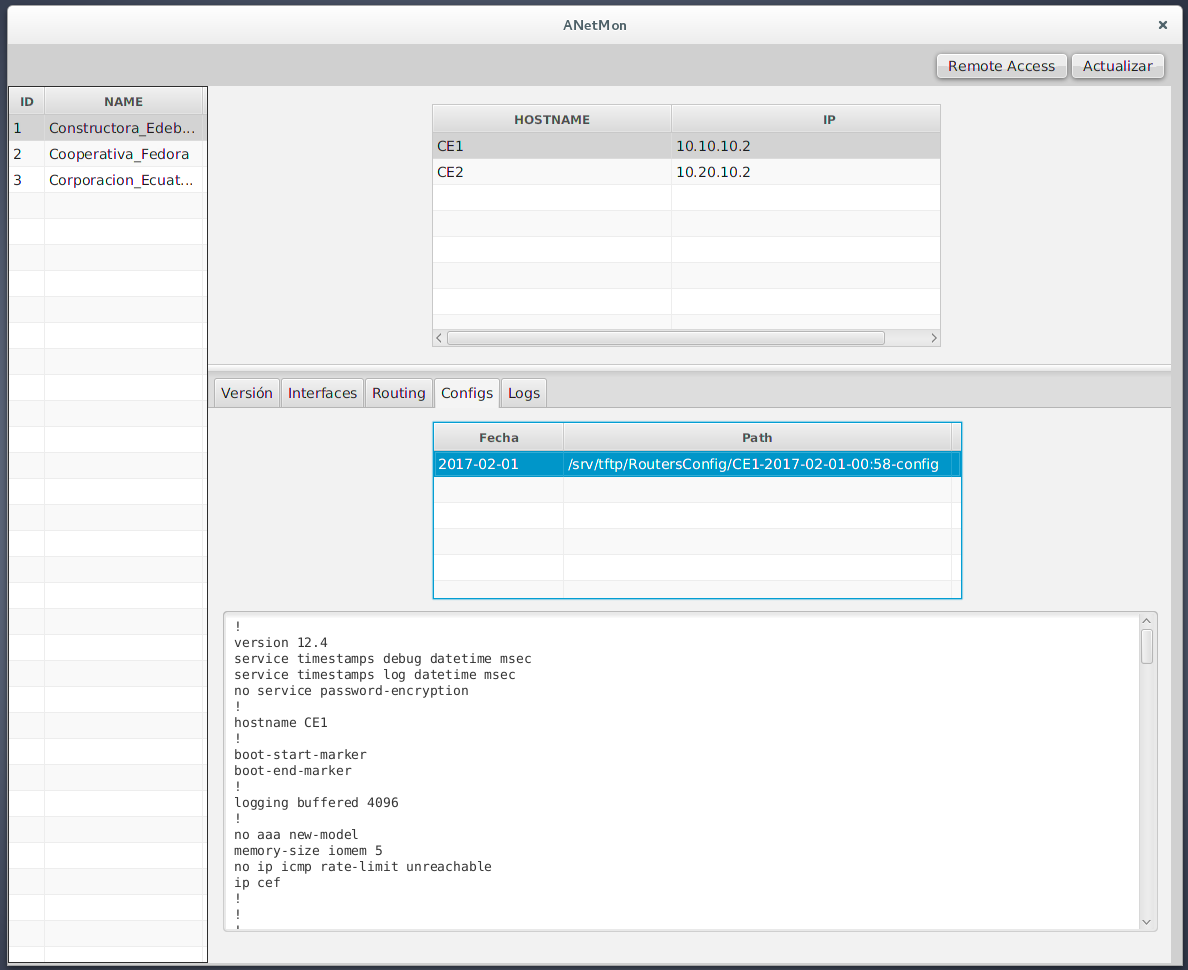
$dbi\_connect->disconnect;

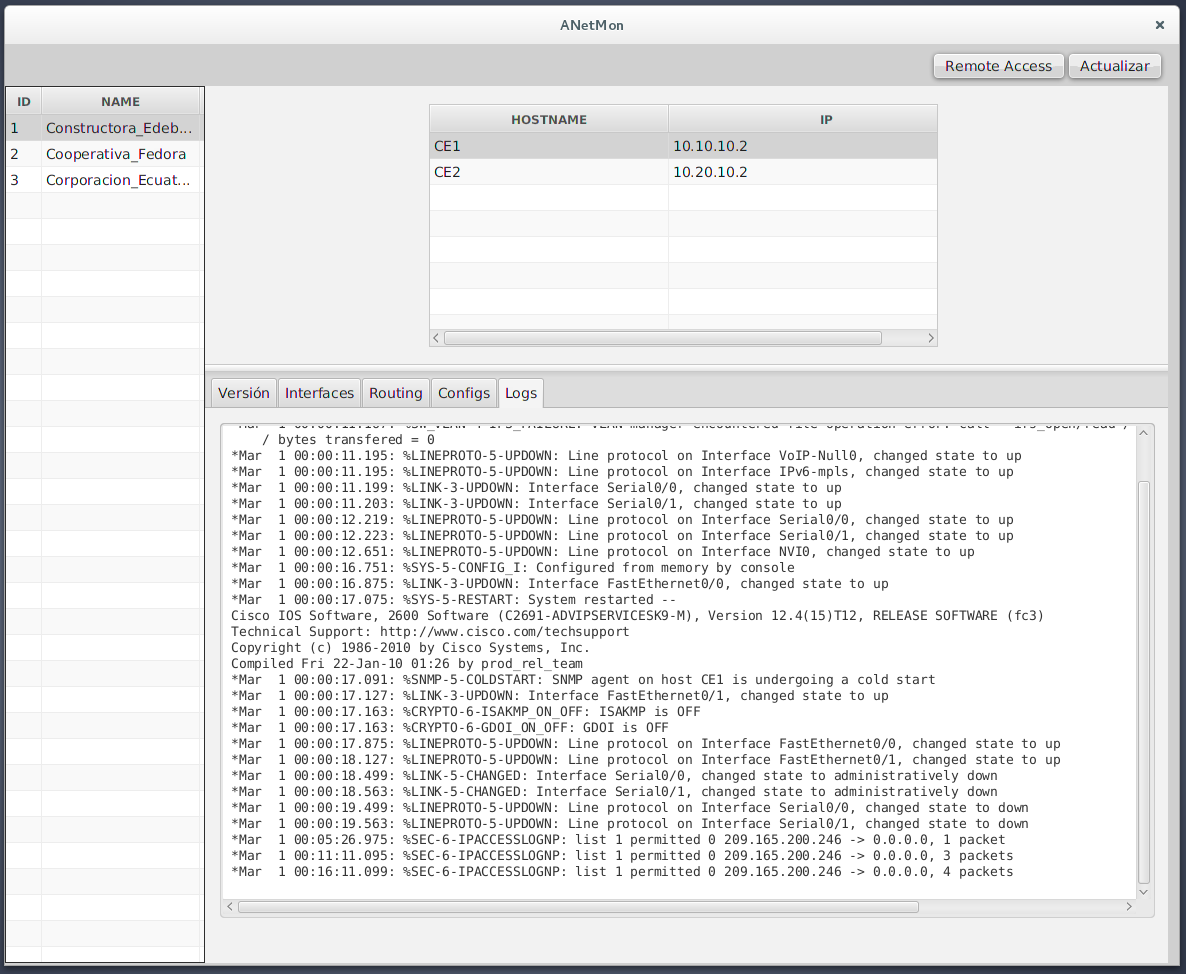
Para mostrar los datos obtenidos de los scripts anteriormente mencionados, hemos desarrollado una aplicación (ANetMon) la cual presenta los últimos datos obtenidos de los parámetros y una lista de todas las configuraciones guardadas. También la aplicación permite iniciar una conexión telnet hacia los dispositivos remotos introduciendo la ip.

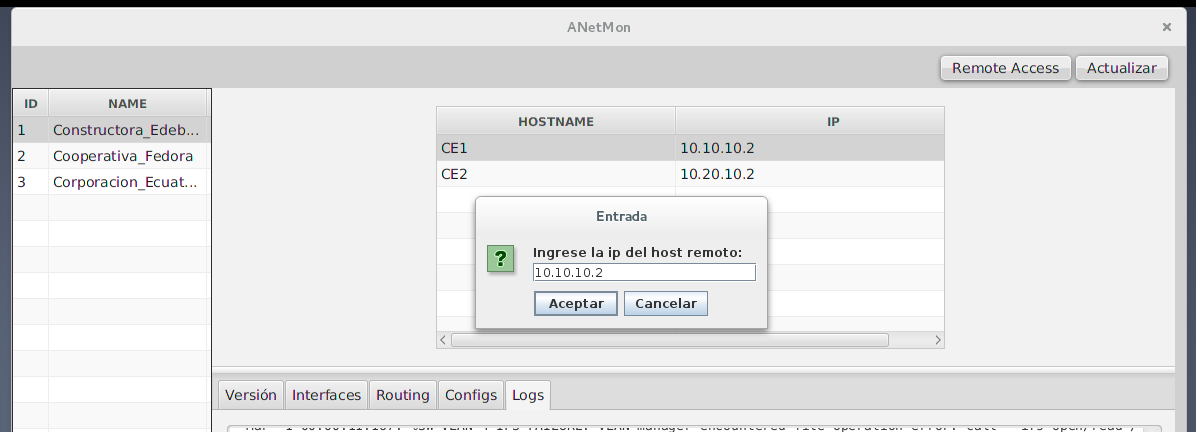


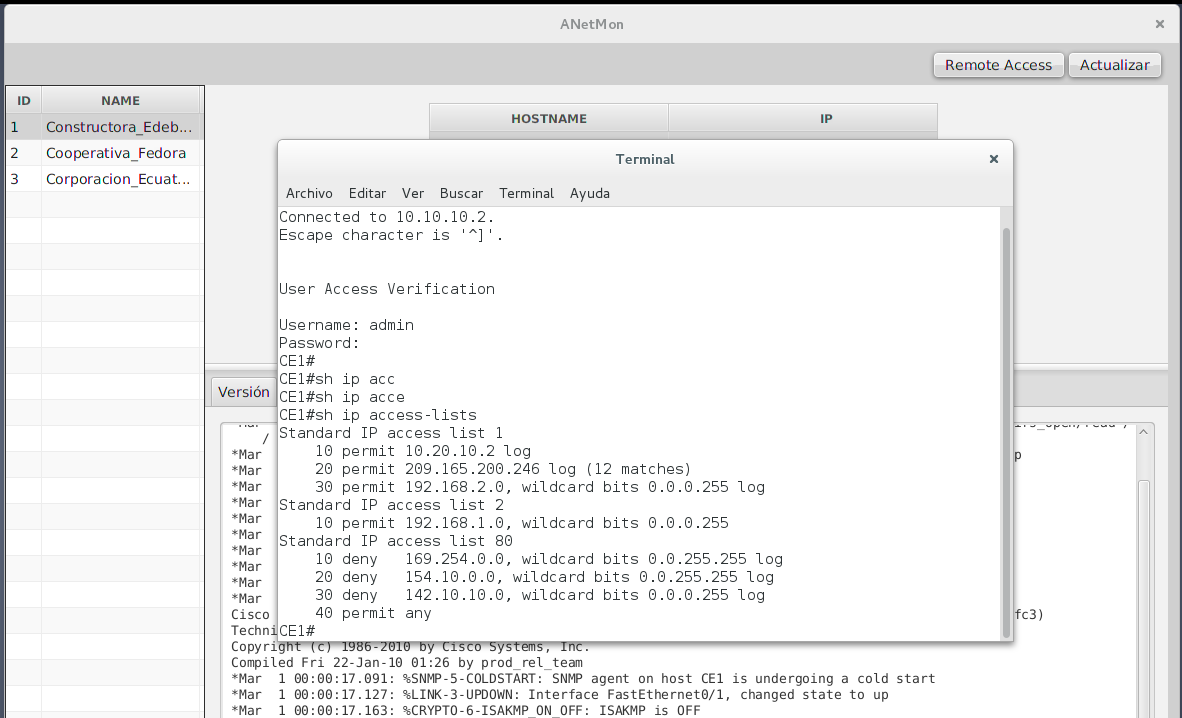












# **Conclusiones**

Es esencial entender el funcionamiento de cada uno de los protocolos utilizados, para aplicarlos de manera correcta en el ambiente de trabajo que lo necesite.

El protocolo l2pt crea un túnel para que dos o más redes con la misma dirección ip puedan comunicarse entre sí, permitiendo al usuario ahorrar espacio en el uso de direcciones ip cuando se tienen muchas sucursales.

Con el diseño de red propuesto, fácilmente se pueden atender las necesidades actuales que un proveedor de servicio de internet puede brindar a sus clientes, contamos con un enrutamiento avanzado usando MPLS en combinación con Frame Relay, brindando un servicio eficiente y con alto nivel de disponibilidad y seguridad.

# **Recomendaciones**

La red puede ser mejorada, ya que los usuarios el cliente CF no pueden navegar a internet, esto podría lograrse agregando una conexión más entre el enrutador y el switch para que el nuevo puerto conectado haga las veces de Gateway predeterminado.

Mejorar la redundancia de la red, porque la misma no ha sido tomada muy en cuenta en este proyecto. Si observamos detenidamente la topología, se puede notar que el nivel de redundancia es muy bajo.

Utilizar equipos más robustos, que tengan mayor nivel de procesamiento para poder abarcar aún con mayor número de clientes.

# **Referencias**

experts, N. (21 de Diciembre de 2012). *mikrotik xperts*. Obtenido de http://mikrotikxperts.com/index.php/informacion/configuraciones/dhcp-relay/20-dhcp-relay-basico

FiberCorp. (s.f.). *FiberCorp*. Obtenido de https://www.fibercorp.com.ar/grandes-empresas/productos/acceso\_mpls

Gerometta, O. (8 de Diciembre de 2006). *Mis libros de networking*. Obtenido de http://librosnetworking.blogspot.com/2006/12/introduccin-tneles-gre.html

Lewis, M. (24 de Marzo de 2008). *Network world*. Obtenido de http://www.networkworld.com/article/2343443/cisco-subnet/configuring-an-l2tpv3-ethernet-pseudowire.html

Prat, L. (13 de Diciembre de 2011). *The CCIE R&S*. Obtenido de https://aitaseller.wordpress.com/2011/12/13/l2tpv3-vlan-to-vlan/