

ESCUELA SUPERIOR DE CÓMPUTO

Materia: Administración de servicios en red

Práctica 1.2 Telnet y SSH

Autor: Elías Muñoz Primero

Contenido

Introducción	3
Objetivo	3
Requerimientos	3
Objetivo visual de la actividad	3
Información de la actividad	3
Tabla 1. Direccionamiento de enrutamiento	3
Tabla 2. Direccionamiento de equipo	4
Tabla 3. Tipos de equipos utilizados	4
Comandos utilizados en la actividad	4
Desarrollo Práctica Uno	5
Pantallas de funcionamiento	13
Bibliografía	25



Introducción

Objetivo

Realizar una interconexión básica de dos redes mediante tres encaminadores y enrutamiento estático.

Ejecutar una configuración básica en encaminadores para habilitar el enrutamiento estático IP en sus interfaces.

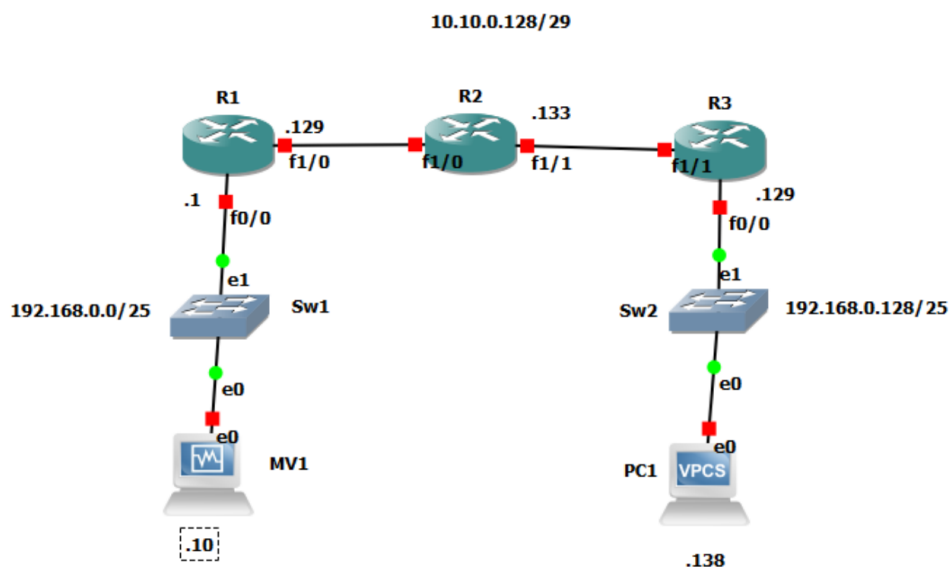
Levantar los servicios de telnet y SSH en los encaminadores con sus respectivas claves de acceso usando REST.

Requerimientos

Una computadora personal

Software para simulación de redes.

Objetivo visual de la actividad



Información de la actividad

Tabla 1. Direccionamiento de enrutamiento

Encaminador	Interfaz	Dirección IP
R1	FastEthernet 0/0	192.168.0.1/25
	FastEthernet 1/0	10.10.0.129/30
R2	FastEthernet 1/0	10.10.0.130/30
	FastEthernet 1/1	10.10.0.133/30
R3	FastEthernet 1/1	10.10.0.134/30
	FastEthernet 0/0	192.168.0.129/25

Tabla 2. Direccionamiento de equipo

Equipo	Dirección IP
MV1	192.168.0.10/25
PC1	192.168.0.138/25

Tabla 3. Tipos de equipos utilizados

Dispositivo	Tipo	Categoría
R1, R2, R3	C7200	Router
Sw1, Sw2	Ethernet switch	Switches
MV1	Máquina virtual	End device
PC1	Emulador de PC	End device

Comandos utilizados en la actividad

Comando	Descripción
enable	Habilita el modo EXEC privilegiado.
configure terminal	Ingresa al modo de configuración global desde la terminal
hostname	Configura el nombre del sistema, que forma parte del indicador de la CLI
interface FastEthernet <i>interface_id</i>	Permite configurar las interfaces para enrutamiento IP, ingresando al modo de configuración de interfaz referente a la interfaz especificada.
ip address <i>ip-address ip-mask</i> [secondary] no ip address <i>ip-address ip-mask</i>	Ajusta, remueve o deshabilita una dirección IP primaria o secundaria en una interfaz. La forma "no" de éste comando remueve la dirección IP especificada y deshabilita la interfaz para procesamiento IP.
no shutdown	Habilita una interfaz para enrutamiento IP y permite que la interfaz sea habilitada automáticamente en el inicio del dispositivo.
show running-config	Despliega los comandos no predeterminados de configuración del dispositivo realizados por el usuario.
show ip interface brief	Muestra la información acerca de la configuración IP y el estado de las interfaces del router
show ip route [destination-prefix destination-prefix-mask] connected ospf rip static summary	Proporciona la información acerca de las rutas IP, desplegando el estado de la tabla de enrutamiento
ip route dirección-red máscara-subred dirección-ip	Configuración de rutas estáticas entre routers

Desarrollo Práctica Dos

Los siguientes pasos describen la forma de realizar la actividad propuesta. Conteste las preguntas y complete la información que se le pide, realice capturas de pantalla en los puntos más importante, realice un pequeño reporte (solo parte práctica) mediante un documento pdf a moodle.

1. Inicie el programa de simulación de redes en su PC.
2. Construya la topología de red completa como está indicada en la imagen 1. Debe insertar los dispositivos indicados en el área de trabajo; para ello, seleccione los dispositivos de la parte izquierda de la ventana del programa y arrástrelos hacia el área de trabajo.
3. También debe seleccionar las conexiones acordes al diagrama, seleccionándolos igualmente de la parte izquierda. Para conectar dos dispositivos dé clic en alguno de ellos, y se desplegará un cuadro con las interfaces disponibles en el mismo; seleccione la interfaz y repita esta operación en el segundo dispositivo.
4. Inicialice la emulación y configure los nombres a cada uno de los dispositivos, así como las direcciones IP de cada uno de los dispositivos de red.
5. Active, enrutamiento estático en cada uno de los encaminadores y pruebe su configuración con un ping entre MV1 y PC1.
6. En las CLIs de cada encaminador configure los siguientes elementos:
 - a. Activar secret con password 12345678 y el servicio de encriptación
 - `config t`
 - `ip domain-name practica2`
 - `ip ssh rsa keypair-name sshkey`
 - `crypto key generate rsa usage-keys label sshkey modulus 1024`
 - `ip ssh v 2`
 - `ip ssh time-out 30`
 - `ip ssh authentication-retries 3`
 - b. Cree una clave de acceso para el modo router que sea 1234 y el acceso mediante telnet y ssh para el siguiente inciso.
 - `line vty 0 6`
 - `password 1234`
 - `login local`
 - `transport input ssh telnet`
 - c. Crear un usuario con permisos de administrador llamado admin y clave de acceso admin01 y que se conecte mediante telnet.
 - `username admin privilege 15 password admin01`

7. Diseñe una aplicación REST usando Python que permita crear, dar de baja y modificar usuarios que se conecten mediante SSH en los dispositivos de red.
8. Agregue una opción desde el REST que nos permita conectarnos usando SSH para probar el correcto funcionamiento de su configuración desde la MV1.




```
1. import telnetlib
2.
3. def conexion_telnet(host,usr,psw,cmds,opc=None):
4.     tn=telnetlib.Telnet(host)
5.     tn.read_until(b"Username: ")
6.     tn.write(usr.encode('ascii')+b"\n")
7.     tn.read_until(b"Password: ")
8.     tn.write(psw.encode('ascii') + b"\n")
9.     for i in cmds:
10.         tn.write(i.encode("ascii")+b"\n")
11.         if opc==1:
12.             result = str(tn.read_all().decode("ascii"))
13.             tn.close()
14.             return result
15.     print(tn.read_all().decode("ascii"))
16.     tn.close()
```

Ilustración 1 Implementación de telnetlib

```
1. import sqlite3
2.
3. def create_db(name):
4.     return sqlite3.connect(name)
5.
6. def close_db(conexion):
7.     conexion.close()
8.
9. def create_tb(conexion):
10.    cursor_tb = conexion.cursor()
11.    cursor_tb.execute(
12.        """
13.        create table if not exists
14.        usuarios(
15.            usr text primary key,
16.            rou text not null,
17.            psw text not null
18.        )
19.        """
20.    )
21. def valitate_usr(conexion,usr):
22.    cursor_tb = conexion.cursor()
23.    sentencia = "select * from usuarios where usr=?"
24.    respuesta = cursor_tb.execute(sentencia,(usr,))
25.    existencia = respuesta.fetchone()
26.    if existencia!=None:
27.        existe = 1
28.        # print("El usuario ya existe")
29.    else:
30.        existe = 0
31.        # print("El usuario NO existe")
32.    return existe
33.
34.
35. def insert_data(conexion,list_data):
36.    cursor_tb = conexion.cursor()
37.    valida = valitate_usr(conexion,list_data[0])
38.    if valida == 1:
39.        print("insert_data no valido")
40.    else:
41.        sentencia = "insert into usuarios(usr,rou,psw) values(?,?,?)"
42.        cursor_tb.execute(sentencia,list_data)
43.        conexion.commit()
44.        print("Usuario {} Registrado".format(list_data[0]))
45.
46. def select_all_data(conexion):
47.    cursor_tb = conexion.cursor()
48.    sentencia = "select * from usuarios"
49.    resultado = cursor_tb.execute(sentencia)
50.    return resultado
51.
52. def select_specific(conexion,usr):
53.    cursor_tb = conexion.cursor()
54.    valida = valitate_usr(conexion,usr)
55.    if valida == 0:
56.        print("Select_specific no valido")
57.        resultado = None
58.    else:
59.        sentencia = "select * from usuarios where usr=?"
60.        resultado = cursor_tb.execute(sentencia,(usr,))
61.    return resultado
62.
63. def update_data(conexion,list_data):
64.    cursor_tb = conexion.cursor()
65.    valida = valitate_usr(conexion,list_data[0])
66.    if valida == 0:
67.        print("update_data no valido")
68.    else:
```



```
69.         sentencia = "update usuarios set psw=? , rou=? where
        usr=?"
70.         lista_data = list(list_data)
71.         lista_data.reverse()
72.         cursor_tb.execute(sentencia, lista_data)
73.         conexion.commit()
74.         print("modificando {}".format(list_data[0]))
75.
76. def delete_data(conexion, usr):
77.     cursor_tb = conexion.cursor()
78.     valida = valitate_usr(conexion, usr)
79.     if valida == 0:
80.         print("delete_data no valido")
81.     else:
82.         sentencia = "delete from usuarios where usr=?"
83.         cursor_tb.execute(sentencia, (usr,))
84.         conexion.commit()
85.         print("eliminando {}".format(usr))
86.
```

Ilustración 2 Implementación de SQLite



```

1. """
2.     Instituto Politecnico Nacional
3.     Escuela Superior de Computo
4.     Administracion de servicios en red
5.     Practica 2 - SSH y Telnet
6. """
7.
8. from flask import Flask, request, render_template
9. from db_services import *
10. from conexion_telnet import *
11. import io
12. # app = Flask(__name__, template_folder="nombrefolder")
13. user_admin="admin"
14. psw_admin="admin01"
15. app = Flask(__name__)
16.
17. @app.route('/', methods = ['POST', 'GET'])
18. def inicio():
19.     return render_template("/index.html")
20.
21. @app.route('/alta', methods = ['POST', 'GET'])
22. def alta():
23.     if request.method == 'GET':
24.         conexion = create_db("P02.db")
25.         par1 = request.args.get('usr')
26.         par2 = request.args.get('rou')
27.         par3 = request.args.get('psw')
28.         par4 = request.args.get('registra')
29.         exist = valitate_usr(conexion, par1)
30.         insert_data(conexion, (par1, par2, par3))
31.         host = ""
32.         if par2 == "R1":
33.             host = "10.10.0.129"
34.         elif par2 == "R2":
35.             host = "10.10.0.130"
36.         elif par2 == "R3":
37.             host = "10.10.0.134"
38.         comandos = ["conf t", "username {} privilege 15 password
39.         {}".format(par1, par3), "end", "exit"]
40.         conexion_telnet(host, user_admin, psw_admin, comandos)
41.         return render_template("/alta.html", usr=par1, rou=par2, psw=par3, reg
42.         istra=par4, existe=exist)
43.     else:
44.         return render_template("/alta.html", usr=None, rou=None, psw=None, reg
45.         istra=0, existe=None)
46.
47. @app.route('/consulta', methods = ['POST', 'GET'])
48. def consulta():
49.     if request.method == 'GET':
50.         conexion = create_db("P02.db")
51.         par1 = request.args.get('usr')
52.         par2 = request.args.get('opc')
53.         if int(par2) == 1:
54.             atributos = select_specific(conexion, str(par1))
55.             return render_template("/consulta.html", filas=None, control
56.             =1, list_data=atributos)
57.         elif int(par2) == 3:
58.             atributos = select_specific(conexion, str(par1))
59.             return render_template("/consulta.html", filas=None, control
60.             =3, list_data=atributos)
61.         else:
62.             conexion = create_db("P02.db")
63.             resultad = select_all_data(conexion)
64.             return render_template("/consulta.html", filas=resultad, control=0)
65.
66. @app.route('/modifica', methods = ['POST', 'GET'])
67. def modifica():
68.     conexion = create_db("P02.db")
69.     par1 = request.args.get('usr')
70.     par2 = request.args.get('rou')

```

```

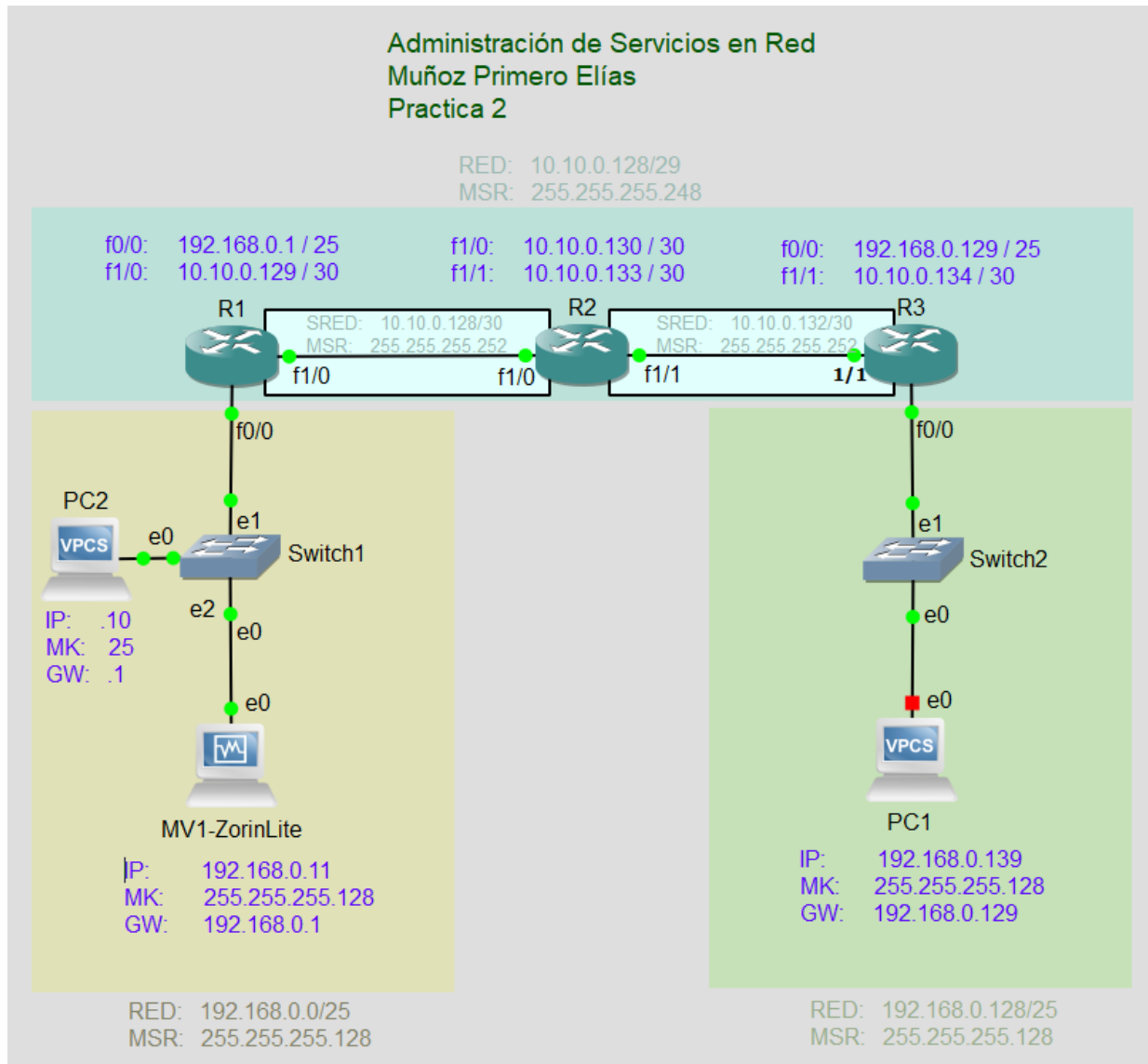
66.     par3 = request.args.get('psw')
67.     print("valores: {} {} {}".format(par1,par2,par3))
68.     update_data(conexion, (str(par1),str(par2),str(par3)))
69.     atributos = select_specific(conexion,str(par1))
70.     host = ""
71.     if par2 == "R1":
72.         host = "10.10.0.129"
73.     elif par2 == "R2":
74.         host = "10.10.0.130"
75.     elif par2 == "R3":
76.         host = "10.10.0.134"
77.     comandos = ["conf t","username {} privilege 15 password
{}"].format(par1,par3),"end","exit"]
78.     conexion_telnet(host,user_admin,psw_admin,comandos)
79.     return render_template("/consulta.html",filas=None,control=2,list_data=atributos)
80.
81. @app.route('/elimina',methods = ['POST','GET'])
82. def elimina():
83.     conexion = create_db("P02.db")
84.     par1 = request.args.get('usr')
85.     par2 = request.args.get('rou')
86.     par3 = request.args.get('psw')
87.     print("valores: {} {} {}".format(par1,par2,par3))
88.     delete_data(conexion,str(par1))
89.     host = ""
90.     if par2 == "R1":
91.         host = "10.10.0.129"
92.     elif par2 == "R2":
93.         host = "10.10.0.130"
94.     elif par2 == "R3":
95.         host = "10.10.0.134"
96.     comandos = ["conf t","no username {} privilege 15 password
{}"].format(par1,par3),"end","exit"]
97.     conexion_telnet(host,user_admin,psw_admin,comandos)
98.     return render_template("/consulta.html",filas=None,control=4)
99.
100. @app.route('/SSH',methods = ['POST','GET'])
101. def SSH():
102.     if request.method == 'GET':
103.         return render_template("/SSH.html",control=0)
104.     else:
105.         conexion = create_db("P02.db")
106.         usuario = request.form["usr"]
107.         contras = request.form["psw"]
108.         info = select_specific(conexion,usuario)
109.         return render_template("/SSH.html",control=1,list_data=info)
110.
111. @app.route('/SSH/CMD',methods = ["POST","GET"])
112. def CMD():
113.     if request.method == "GET":
114.         par1 = request.args.get("usr")
115.         par2 = request.args.get("rou")
116.         par3 = request.args.get("psw")
117.         par4 = request.args.get("comando")
118.         host = ""
119.         if par2 == "R1":
120.             host = "10.10.0.129"
121.         elif par2 == "R2":
122.             host = "10.10.0.130"
123.         elif par2 == "R3":
124.             host = "10.10.0.134"
125.
126.         comandos = []
127.
128.         # hacemos uso del buffer io, para poder parsear el textarea
del CMD.html
129.         aux_buf = io.StringIO(par4)
130.         aux_list = aux_buf.readlines()
131.         print("Comandos sin parsing: {}".format(aux_list))

```

```
132.
133.         for e in aux_list:
134.             aux_cmd = e[:-2]
135.             comandos.append(aux_cmd)
136.
137.         print(comandos)
138.         comandos.append("exit")
139.         par4 = conexion_telnet(host, par1, par3, comandos, 1)
140.         return render_template("/CMD.html", usr=par1, rou=par2, psw=par
3, control=1, cmd=par4)
141.     else:
142.         par1 = request.form["usr"]
143.         par2 = request.form["rou"]
144.         par3 = request.form["psw"]
145.         return render_template("/CMD.html", usr=par1, rou=par2, psw=par
3, control=0)
146.
147. if __name__ == '__main__':
148.     conexion = create_db("P02.db")
149.     create_tb(conexion)
150.     # close_db(conexion)
151.     app.run(host='0.0.0.0', debug=True)
152.
```

Ilustración 3 Aplicación en Flask

Pantallas de funcionamiento



```
*Apr 19 06:47:18.483: %LINK-5-CHANGED: Interface FastEthernet1/0, changed state to administratively down
R1#
R1#
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface fas
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.0.1 255.255.255.128
R1(config-if)#no shutdown
R1(config-if)#
*Apr 19 06:48:12.139: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R1(config-if)#
*Apr 19 06:48:12.139: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Down
*Apr 19 06:48:13.139: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#interface fastEthernet 1/0
R1(config-if)#ip address 10.10.0.129 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
*Apr 19 06:48:35.987: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
R1(config-if)#
*Apr 19 06:48:35.987: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa1/0 Physical Port Administrative State Down
*Apr 19 06:48:36.987: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
R1(config-if)#exit
R1(config)#ip route 10.10.0.128 255.255.255.248 10.10.0.130
R1(config)#ip route 10.10.0.128 255.255.255.248 10.10.0.130
R1(config)#end
R1#
*Apr 19 06:49:00.771: %SYS-5-CONFIG_I: Configured from console by console
R1#write
Building configuration...
[OK]
R1#show ip route
Codes: 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
       i - IS-IS, su - IS-IS summary, 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

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.0.128/30 is directly connected, FastEthernet1/0
S       10.10.0.128/29 [1/0] via 10.10.0.130
       192.168.0.0/25 is subnetted, 1 subnets
C       192.168.0.0 is directly connected, FastEthernet0/0
R1#
```

```
Gateway of last resort is not set

  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.0.128/30 is directly connected, FastEthernet1/0
S       10.10.0.128/29 [1/0] via 10.10.0.130
        192.168.0.0/25 is subnetted, 1 subnets
C       192.168.0.0 is directly connected, FastEthernet0/0
R1#show ip route
Codes: 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
        i - IS-IS, su - IS-IS summary, 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

  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.0.128/30 is directly connected, FastEthernet1/0
S       10.10.0.128/29 [1/0] via 10.10.0.130
        192.168.0.0/25 is subnetted, 1 subnets
C       192.168.0.0 is directly connected, FastEthernet0/0
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 192.168.0.128 255.255.255.128 10.10.0.130
R1(config)#end
R1#write
Building configuration...

*Apr 19 06:52:57.503: %SYS-5-CONFIG_I: Configured from console by console[OK]
R1#show ip route
Codes: 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
        i - IS-IS, su - IS-IS summary, 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

  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.0.128/30 is directly connected, FastEthernet1/0
S       10.10.0.128/29 [1/0] via 10.10.0.130
        192.168.0.0/25 is subnetted, 2 subnets
C       192.168.0.0 is directly connected, FastEthernet0/0
S       192.168.0.128 [1/0] via 10.10.0.130
R1#
```


Configuración Router Dos

```
*Apr 19 06:50:47.503: %LINK-5-CHANGED: Interface FastEthernet1/0, changed state to administratively down
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface fastEthernet 1/0
R2(config-if)#ip address 10.10.0.130 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*Apr 19 06:51:57.507: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
R2(config)#
*Apr 19 06:51:57.507: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa1/0 Physical Port Administrative State Down
*Apr 19 06:51:58.507: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
R2(config)#interface fastEthernet 1/1
R2(config-if)#ip address 10.10.0.133 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*Apr 19 06:52:15.007: %LINK-3-UPDOWN: Interface FastEthernet1/1, changed state to up
R2(config)#
*Apr 19 06:52:15.007: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa1/1 Physical Port Administrative State Down
*Apr 19 06:52:16.007: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
R2(config)#ip route 10.10.0.128 255.255.255.248 10.10.0.129
R2(config)#ip route 10.10.0.128 255.255.255.248 10.10.0.134
R2(config)#ip route 192.168.0.0 255.255.255.128 10.10.0.129
R2(config)#ip route 192.168.0.128 255.255.255.128 10.10.0.134
R2(config)#end
R2#write
*Apr 19 06:52:35.251: %SYS-5-CONFIG_I: Configured from console by console
R2#write
Building configuration...
[OK]
R2#show ip route
Codes: 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
       i - IS-IS, su - IS-IS summary, 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

    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.10.0.128/30 is directly connected, FastEthernet1/0
S       10.10.0.128/29 [1/0] via 10.10.0.134
           [1/0] via 10.10.0.129
C       10.10.0.132/30 is directly connected, FastEthernet1/1
S       192.168.0.0/25 is subnetted, 2 subnets
S       192.168.0.0 [1/0] via 10.10.0.129
S       192.168.0.128 [1/0] via 10.10.0.134
R2#
```

Configuración Router Tres

```
R3#
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface fastEthernet 0/0
R3(config-if)#ip address 192.168.0.129 255.255.255.128
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Apr 19 06:56:24.335: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
R3(config)#
*Apr 19 06:56:24.335: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa0/0 Physical Port Administrative State Down
*Apr 19 06:56:25.335: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R3(config)#interface fastEthernet 1/1
R3(config-if)#ip address 10.10.0.134 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#
*Apr 19 06:56:40.851: %LINK-3-UPDOWN: Interface FastEthernet1/1, changed state to up
R3(config)#
*Apr 19 06:56:40.851: %ENTITY_ALARM-6-INFO: CLEAR INFO Fa1/1 Physical Port Administrative State Down
*Apr 19 06:56:41.851: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
R3(config)#ip route 10.10.0.128 255.255.255.248 10.10.0.133
R3(config)#ip route 192.168.0.0 255.255.255.128 10.10.0.133
R3(config)#end
R3#write
*Apr 19 06:56:54.003: %SYS-5-CONFIG_I: Configured from console by console
R3#write
Building configuration...
[OK]
R3#show ip route
Codes: 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
       i - IS-IS, su - IS-IS summary, 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

    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       10.10.0.128/29 [1/0] via 10.10.0.133
C       10.10.0.132/30 is directly connected, FastEthernet1/1
        192.168.0.0/25 is subnetted, 2 subnets
S       192.168.0.0 [1/0] via 10.10.0.133
C       192.168.0.128 is directly connected, FastEthernet0/0
R3#
```

Ping entre la PC1 y PC2

```
Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 192.168.0.139 255.255.255.128 gateway 192.168.0.129

PC1> show ip

NAME       : PC1[1]
IP/MASK     : 192.168.0.139/25
GATEWAY     : 192.168.0.129
DNS         :
MAC         : 00:50:79:66:68:01
LPORT      : 10026
RHOST:PORT  : 127.0.0.1:10027
MTU         : 1500

PC1> ping 192.168.0.10
84 bytes from 192.168.0.10 icmp_seq=1 ttl=61 time=97.452 ms
84 bytes from 192.168.0.10 icmp_seq=2 ttl=61 time=97.958 ms
84 bytes from 192.168.0.10 icmp_seq=3 ttl=61 time=96.857 ms
84 bytes from 192.168.0.10 icmp_seq=4 ttl=61 time=97.188 ms
84 bytes from 192.168.0.10 icmp_seq=5 ttl=61 time=96.003 ms

PC1> █
```

```
Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

PC2> show ip

NAME       : PC2[1]
IP/MASK     : 0.0.0.0/0
GATEWAY     : 0.0.0.0
DNS         :
MAC         : 00:50:79:66:68:00
LPORT      : 10028
RHOST:PORT  : 127.0.0.1:10029
MTU         : 1500

PC2> ip 192.168.0.10 25 192.168.0.1
Checking for duplicate address...
PC1 : 192.168.0.10 255.255.255.128 gateway 192.168.0.1

PC2> save
Saving startup configuration to startup.vpc
. done

PC2> ping 192.168.0.139
192.168.0.139 icmp_seq=1 timeout
84 bytes from 192.168.0.139 icmp_seq=2 ttl=61 time=99.592 ms
84 bytes from 192.168.0.139 icmp_seq=3 ttl=61 time=96.665 ms
84 bytes from 192.168.0.139 icmp_seq=4 ttl=61 time=95.822 ms
84 bytes from 192.168.0.139 icmp_seq=5 ttl=61 time=96.433 ms

PC2> █
```

```
PC2> save
Saving startup configuration to startup.vpc
. done

PC2> ping 192.168.0.139
192.168.0.139 icmp_seq=1 timeout
84 bytes from 192.168.0.139 icmp_seq=2 ttl=61 time=99.592 ms
84 bytes from 192.168.0.139 icmp_seq=3 ttl=61 time=96.665 ms
84 bytes from 192.168.0.139 icmp_seq=4 ttl=61 time=95.822 ms
84 bytes from 192.168.0.139 icmp_seq=5 ttl=61 time=96.433 ms

PC2> ping 10.10.0.129
84 bytes from 10.10.0.129 icmp_seq=1 ttl=255 time=19.048 ms
84 bytes from 10.10.0.129 icmp_seq=2 ttl=255 time=19.580 ms
84 bytes from 10.10.0.129 icmp_seq=3 ttl=255 time=16.602 ms
84 bytes from 10.10.0.129 icmp_seq=4 ttl=255 time=22.099 ms
84 bytes from 10.10.0.129 icmp_seq=5 ttl=255 time=18.944 ms

PC2> ping 10.10.0.130
84 bytes from 10.10.0.130 icmp_seq=1 ttl=254 time=40.328 ms
84 bytes from 10.10.0.130 icmp_seq=2 ttl=254 time=43.403 ms
84 bytes from 10.10.0.130 icmp_seq=3 ttl=254 time=42.182 ms
84 bytes from 10.10.0.130 icmp_seq=4 ttl=254 time=41.451 ms
84 bytes from 10.10.0.130 icmp_seq=5 ttl=254 time=41.316 ms

PC2> ping 10.10.0.133
84 bytes from 10.10.0.133 icmp_seq=1 ttl=254 time=43.257 ms
84 bytes from 10.10.0.133 icmp_seq=2 ttl=254 time=31.409 ms
84 bytes from 10.10.0.133 icmp_seq=3 ttl=254 time=41.393 ms
84 bytes from 10.10.0.133 icmp_seq=4 ttl=254 time=46.172 ms
84 bytes from 10.10.0.133 icmp_seq=5 ttl=254 time=42.351 ms

PC2> ping 10.10.0.134
84 bytes from 10.10.0.134 icmp_seq=1 ttl=253 time=65.004 ms
84 bytes from 10.10.0.134 icmp_seq=2 ttl=253 time=54.198 ms
84 bytes from 10.10.0.134 icmp_seq=3 ttl=253 time=63.414 ms
84 bytes from 10.10.0.134 icmp_seq=4 ttl=253 time=62.899 ms
84 bytes from 10.10.0.134 icmp_seq=5 ttl=253 time=65.311 ms

PC2> ping 192.168.0.129
84 bytes from 192.168.0.129 icmp_seq=1 ttl=253 time=62.604 ms
84 bytes from 192.168.0.129 icmp_seq=2 ttl=253 time=65.754 ms
84 bytes from 192.168.0.129 icmp_seq=3 ttl=253 time=62.098 ms
84 bytes from 192.168.0.129 icmp_seq=4 ttl=253 time=57.678 ms
84 bytes from 192.168.0.129 icmp_seq=5 ttl=253 time=54.702 ms

PC2> █
```

Configuración del protocolo SSH en R1

```
R1
R1(config)#ip domain-name practica2
R1(config)#show line
^
% Invalid input detected at '^' marker.

R1(config)#do show line
  Tty Typ    Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*   0 CTY          -    -    -    -    -     0       0      0/0      -
  1 AUX  9600/9600  -    -    -    -    -     0       0      0/0      -
  2 VTY          -    -    -    -    -     0       0      0/0      -
  3 VTY          -    -    -    -    -     0       0      0/0      -
  4 VTY          -    -    -    -    -     0       0      0/0      -
  5 VTY          -    -    -    -    -     0       0      0/0      -
  6 VTY          -    -    -    -    -     0       0      0/0      -

R1(config)#ip ssh rsa keypair-name sshkey
Please create RSA keys (of atleast 768 bits size) to enable SSH v2.
R1(config)#crypto key generate rsa usage-keys label sshkey modulus 1024
The name for the keys will be: sshkey

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

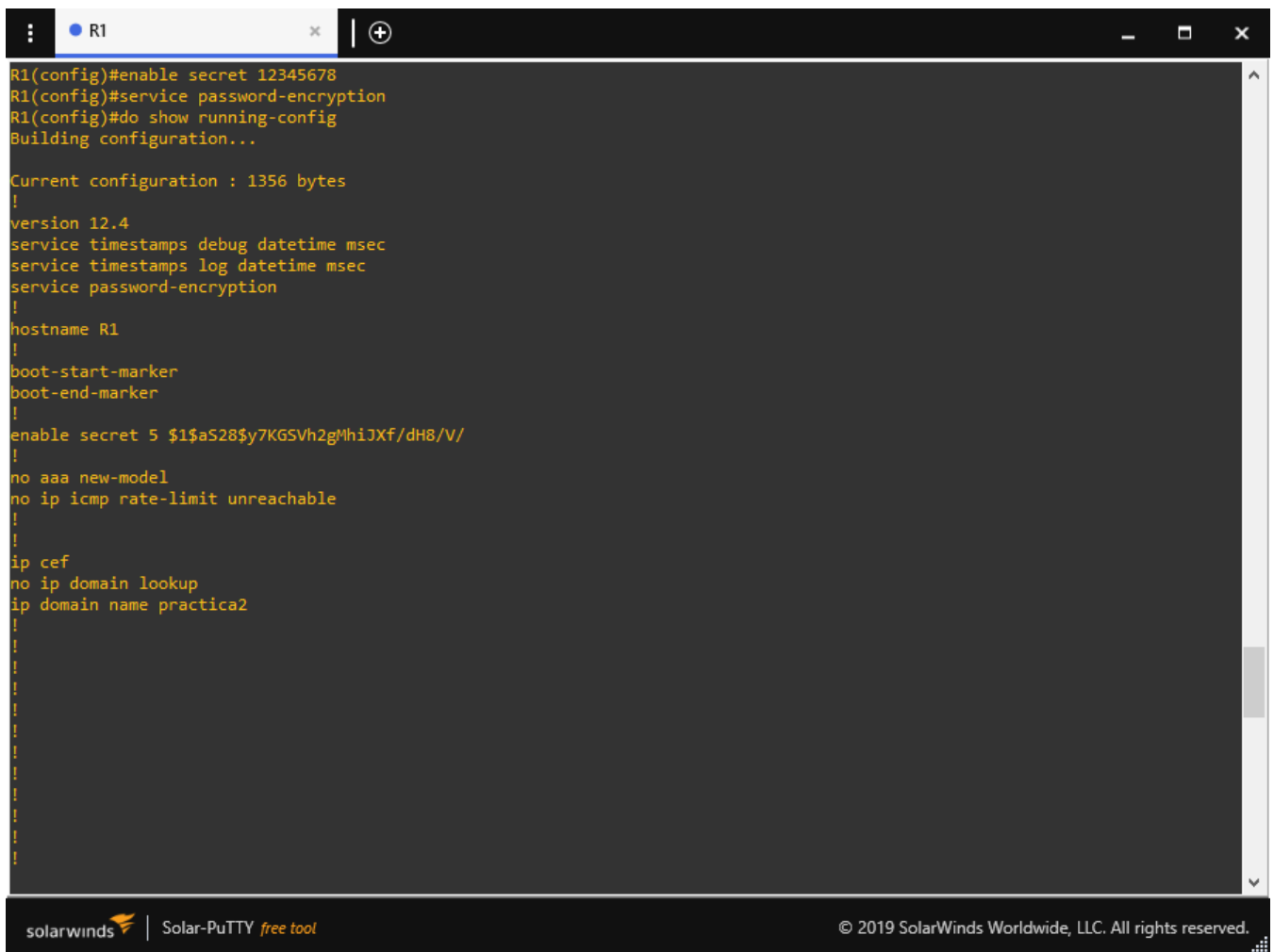
R1(config)#
*Apr 24 18:46:56.267: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#ip ssh v 2
R1(config)#
R1(config)#
R1(config)#ip ssh time-out 30
R1(config)#ip ssh authentication-retries 3
R1(config)#line vty 0 6
R1(config-line)#password 1234
R1(config-line)#login local
R1(config-line)#transport input ssh telnet
R1(config-line)#exit
R1(config)#username admin privilege 15 password admin01
R1(config)#show running-config
```

solarwinds | Solar-PuTTY free tool © 2019 SolarWinds Worldwide, LLC. All rights reserved.

```
R1(config)#do show running-config
Building configuration...

Current configuration : 1287 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
ip cef
no ip domain lookup
ip domain name practica2
!
!
!
!
!
!
!
!
!
!
!
!
```

```
username admin privilege 15 password 0 admin01
!
!
ip tcp synwait-time 5
ip ssh time-out 30
ip ssh rsa keypair-name sshkey
ip ssh version 2
!
!
!
!
!
interface FastEthernet0/0
 ip address 192.168.0.1 255.255.255.128
 duplex half
!
interface FastEthernet1/0
 ip address 10.10.0.129 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
!
ip forward-protocol nd
ip route 10.10.0.128 255.255.255.248 10.10.0.130
ip route 192.168.0.128 255.255.255.128 10.10.0.130
!
no ip http server
no ip http secure-server
!
!
no cdp log mismatch duplex
!
```

```
R1
R1(config)#enable secret 12345678
R1(config)#service password-encryption
R1(config)#do show running-config
Building configuration...

Current configuration : 1356 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$aS28$y7KGSVh2gMhiJXf/dH8/V/
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
ip cef
no ip domain lookup
ip domain name practica2
!
!
!
!
!
!
!
!
!
!
```

solarwinds | Solar-PuTTY free tool

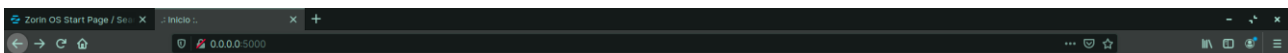
© 2019 SolarWinds Worldwide, LLC. All rights reserved.

```
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 password 7 1158485643
 login local
 transport input telnet ssh
line vty 5 6
 password 7 101F5B4A51
 login local
 transport input telnet ssh
!
!
end

R1(config)#do show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 30 secs; Authentication retries: 3
R1(config)#end
R1#write
Building configuration...

*Apr 24 18:51:56.583: %SYS-5-CONFIG_I: Configured from console by console[OK]
R1#
```

Interfaz de la práctica



Practica Dos SSH y Telnet

[Alta Usuarios](#)[Consulta Usuarios](#)[Prueba SSH](#)

Bibliografía

Última consulta realizada el día 07 de marzo del 2021

[1] Tecnológico Nacional de México, «Cisco Networking Academy,» Marzo 2021. [En línea]. Available: <http://itroque.edu.mx/cisco/cisco1/index.html>.

