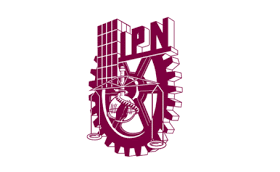
**Instituto Politécnico Nacional**

**Escuela Superior de Cómputo**

Aplicaciones para comunicaciones de red

“Práctica 2”

2do departamental

Profesor: Rangel González Josué

Alumnos:

-Frausto Hernández Omar

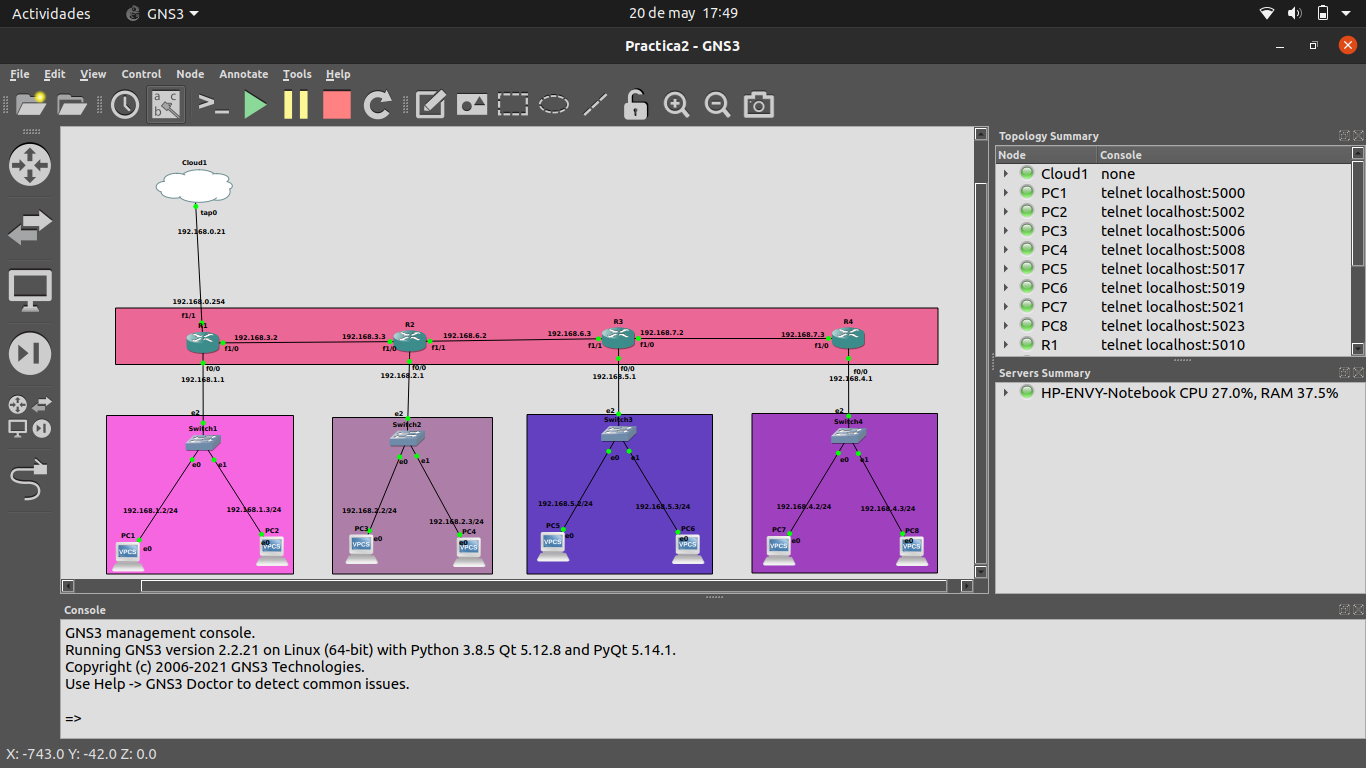
-Guzman Pérez Oscar

-Castro Cruces Jorge Eduardo

Grupo: 3CV18

**Manual de usuario:**

1. **El primer paso fue conectar y configurar la siguiente topología en GNS3:**



Los comandos que se utilizaron para configurar las VPCS son los siguientes:

PC1> ip 192.168.1.2/24 192.168.1.1

PC1> save

Para poder ver que la ip fue asignada correctamente, tecleamos:

PC1> show ip

Los comandos que se utilizaron para configurar los Routers Cisco 7200 son los siguientes:

R1# configure terminal

R1(config)# interface f0/0

R1(config-if)# ip address 192.168.1.1 255.255.255.0

R1(config-if)# no shutdown

R1(config-if)# exit

R1(config)# interface f1/0

R1(config-if)# ip address 192.168.3.2 255.255.255.0

R1(config-if)# no shutdown

R1(config-if)# exit

R1(config)# interface f1/1

R1(config-if)# ip address 192.168.0.254 255.255.255.0

R1(config-if)# no shutdown

R1(config-if)# exit

R1(config)# router rip

R1(config-router)# ver 2

R1(config-router)# network 192.168.1.0

R1(config-router)# network 192.168.2.0

R1(config-router)# network 192.168.3.0

R1(config-router)# network 192.168.4.0

R1(config-router)# network 192.168.5.0

R1(config-router)# network 192.168.6.0

R1(config-router)# end

R1# copy running-config startup-config

Para verificar que se configuró de forma correcta el Router, ingresamos los siguientes comandos:

R1# show ip interface brief

R1# show ip route

R1# show run

1. **El siguiente paso fue crear una tap en linux:**

Para este paso, tuvimos que instalar la siguiente biblioteca en nuestra terminal de Linux:

:~$ sudo apt-get install uml-utilities

Luego, tuvimos que crear la tap:

:~$ sudo tunctl -t tap0 -u eduardocastro

Después, asignamos dirección IP y levantamos la tap:

:~$ sudo ifconfig tap0 192.168.0.21 netmask 255.255.255.0 up

Verificamos que se haya creado de forma correcta la tap:

:~$ ifconfig

1. **Para esto, debemos agregar las rutas a la tabla de ruteo para que la tap sepa por donde enviar y recibir los datos:**

Agregamos las rutas indicadas en la tabla de ruteo de nuestro equipo Linux:

:~$ sudo ip route add 192.168.1.0/24 via 192.168.0.254 dev tap0

:~$ sudo ip route add 192.168.2.0/24 via 192.168.0.254 dev tap0

:~$ sudo ip route add 192.168.3.0/24 via 192.168.0.254 dev tap0

:~$ sudo ip route add 192.168.4.0/24 via 192.168.0.254 dev tap0

:~$ sudo ip route add 192.168.5.0/24 via 192.168.0.254 dev tap0

:~$ sudo ip route add 192.168.6.0/24 via 192.168.0.254 dev tap0

Para verificar que se agregó correctamente las rutas indicadas en la tabla de ruteo de nuestro equipo Linux, tecleamos:

:~$ ip route

:~$ route -n

Siguiendo la recomendación del profesor, se nos hizo buena idea codificar un Script que nos facilite la tarea que acabamos de hacer:

#!/bin/bash

#variables

ipTap=192.168.0.21

mask=255.255.255.0

ipTapBroadcast=192.168.0.254

#Crear tap

sudo tunctl -t tap0 -u eduardocastro

echo tap creada...

#Levantar y asignar IP

sudo ifconfig tap0 $ipTap netmask $mask up

echo tap lista y configurada...

echo ip = $ipTap

echo mascara = $mask

#Agregar rutas a la tabla de ruteo

sudo ip route add 192.168.1.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.2.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.3.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.4.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.5.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.6.0/24 via $ipTapBroadcast dev tap0

sudo ip route add 192.168.7.0/24 via $ipTapBroadcast dev tap0

echo Rutas agregadas a la tabla de ruteo

Sin duda, se le tuvo que dar permisos de ejecución al fichero:

:~$ sudo chmod +x ScriptTap0.sh

1. **Acto seguido, realizamos una modificación en el Router 1:**

La única modificación que se le hizo al Router, fue para que permitiese transmitir los mensajes de Broadcast a las demás subredes:

R1# configure terminal

R1(config)# line vty 0

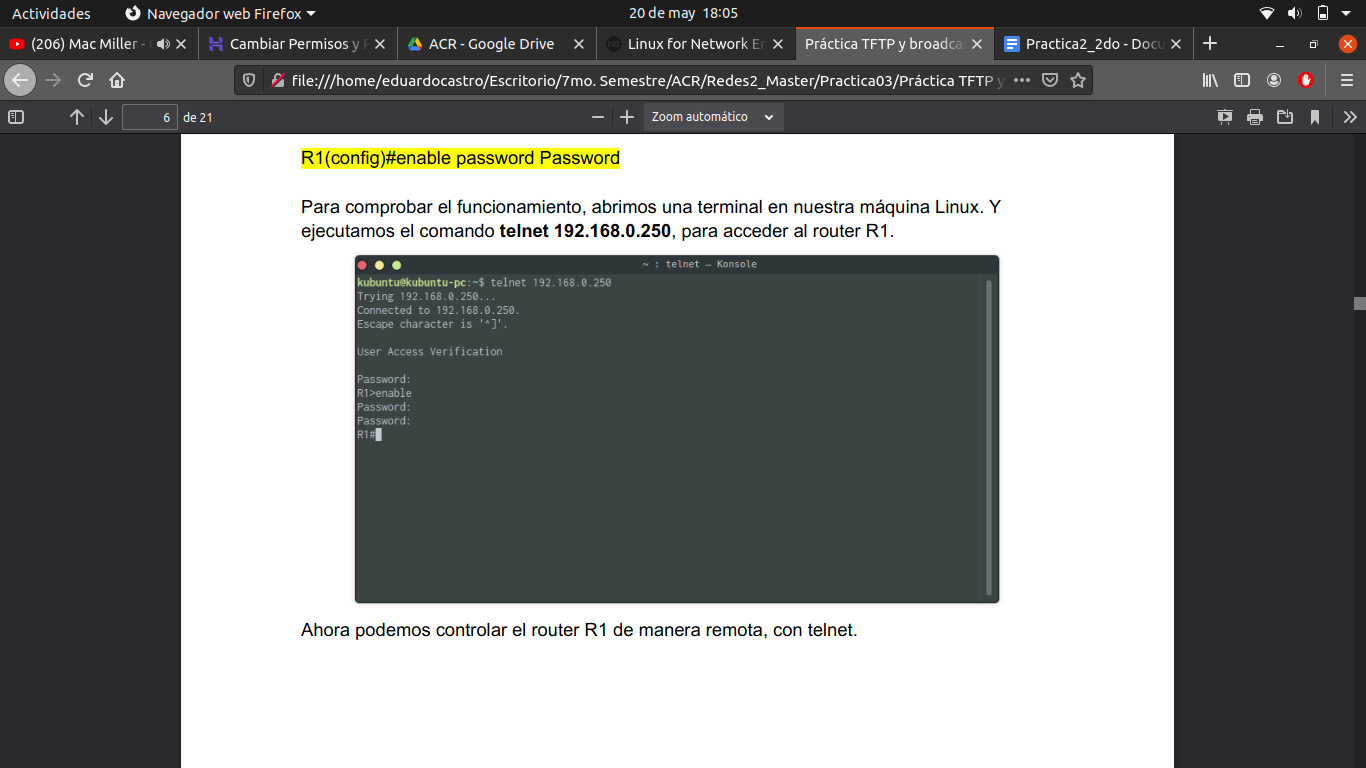
R1(config-line)# password 1234

R1(config-line)# login

R1(config-line)# exit

R1(config)# enable password Password

Para comprobar el funcionamiento, abrimos una terminal en nuestra máquina Linux. Y ejecutamos el comando ​telnet 192.168.0.11​, para acceder al router R1.

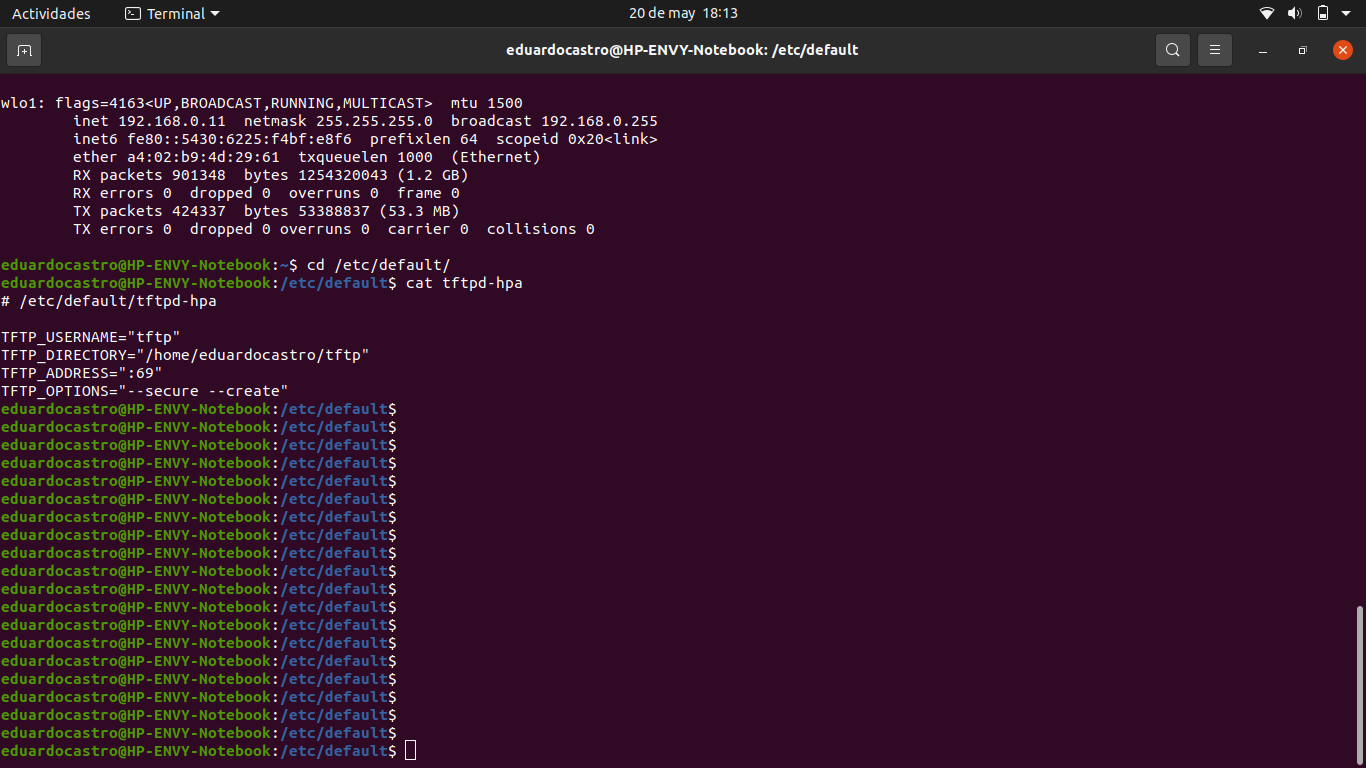


1. **Configuración del servidor TFTP en Linux**

Primero instalamos el paquete tftp-hpa :

$ sudo apt install tftpd-hpa

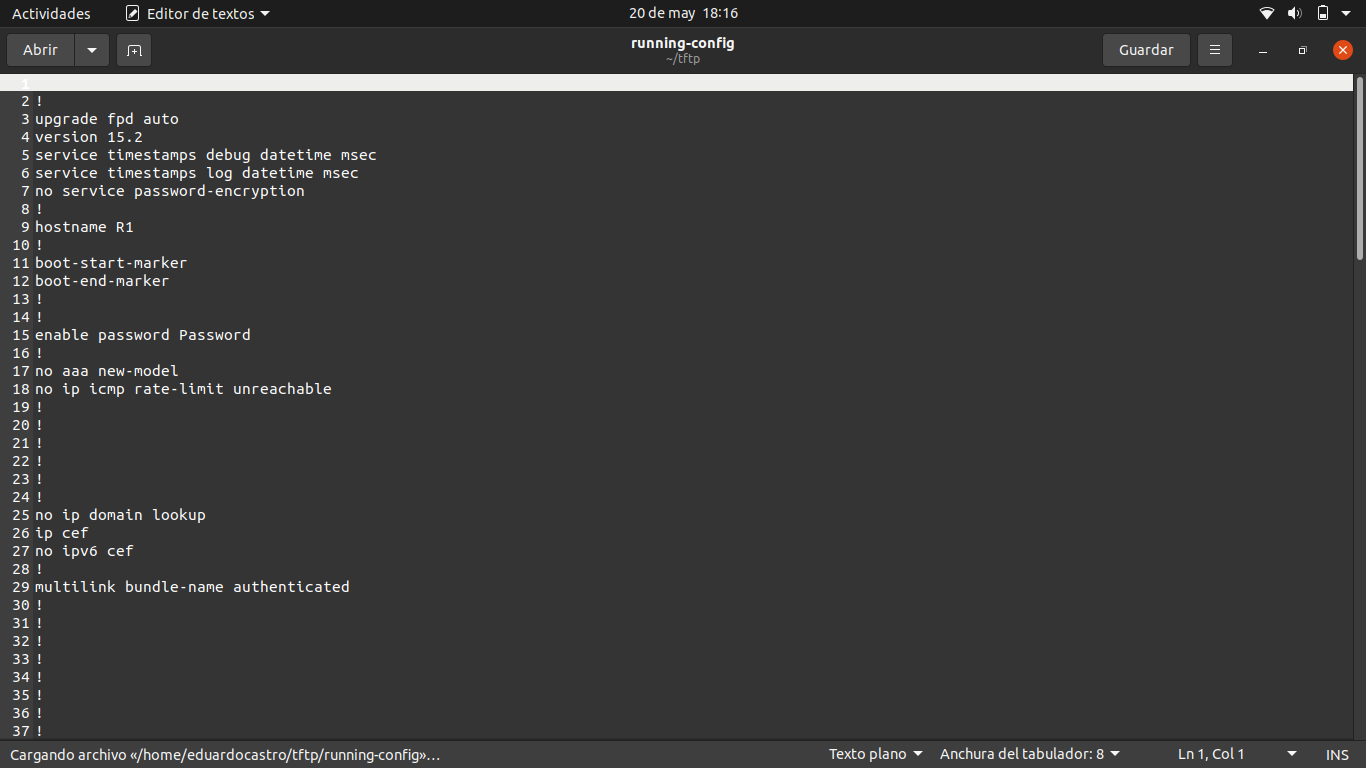
Abrimos el archivo de configuración **​/etc/default/tftp-hpa** del programa TFTP y cambiamos la ruta de carpeta TFTP, en mi caso será **​/home/eduardocastro/tftp:**



1. **Enviar y recibir running-config del router R1, a través de TFTP**

Ahora Probamos Enviando Archivo De Configuración ​“running-config” del router R1 al servidor TFTP en linux. Ejecutamos el siguiente comando en el router R1:

R1#copy running-config tftp



1. **Elaboración del Script en Python**

import telnetlib

import os

import time

TFTP\_DIR = '/home/eduardocastro/tftp'

HOST\_R1 = "192.168.0.254" # ip addr router r1

HOST\_R2 = "192.168.3.3" # ip addr router r2

HOST\_R3 = "192.168.6.3" # ip addr router r3

HOST\_R4 = "192.168.7.3" # ip addr router r4

IP\_TFTP = "192.168.0.11" # ip tfpt server

password = '1234' # password telnet

password\_enable = 'Password' # password for enable mode

'''

SEND AND RECIVE RUNNIG CONFIG FOR ROUTER R1

'''

tn\_r1 = telnetlib.Telnet(HOST\_R1) # connecto to r1

tn\_r1.read\_until(b"Password: ")

tn\_r1.write(password.encode('ascii') + b"\r") # write password telnet

print("You are connected to {}".format(HOST\_R1))

tn\_r1.read\_until(b"R1>")

tn\_r1.write('enable'.encode('ascii') + b"\r") # write enable command

tn\_r1.read\_until(b"Password: ")

tn\_r1.write(password\_enable.encode('ascii') + b"\r") # write password enable

print('R1: "enable" mode activated')

tn\_r1.read\_until(b"R1#")

tn\_r1.write('copy running-config tftp:'.encode('ascii') + b"\r") # send runnig config to tftp server in linux

tn\_r1.read\_until(b"Address or name of remote host []? ")

tn\_r1.write(IP\_TFTP.encode('ascii') + b"\r") # set server tftp address

tn\_r1.read\_until(b"Destination filename [r1-confg]? ")

tn\_r1.write('backup'.encode('ascii') + b"\r") # set file name

time.sleep(6) # wait to send file

print('R1: runnig-config send to tftp server')

path = os.path.join(TFTP\_DIR, 'backup')

file = open(path, 'r');

# Read operation

lines = []

for line in file.readlines():

if 'R1' in line:

lines.append(line.replace('R1', 'Remplazo1'))

else:

lines.append(line)

file.close()

# Write operation

new\_file = open(path, 'w+')

new\_file.writelines(lines)

new\_file.close()

print('R1: hostname R1 is update to Remplazo1')

tn\_r1.read\_until(b"R1#")

tn\_r1.write('copy tftp: running-config'.encode('ascii') + b"\r") # send runnig config to router in gns3

tn\_r1.read\_until(b"Address or name of remote host []? ")

tn\_r1.write(IP\_TFTP.encode('ascii') + b"\r") # set tftp server ip address

tn\_r1.read\_until(b"Source filename []? ")

tn\_r1.write('backup'.encode('ascii') + b"\r") # set filename of backup

tn\_r1.read\_until(b"Destination filename [running-config]? ")

tn\_r1.write(b"\r") # set return key to continue

print('R1: running-config modified send to router R1')

tn\_r1.read\_until(b"Remplazo1#")

tn\_r1.write('write'.encode('ascii') + b"\r") # save config

print('R1: configuration is saved')

tn\_r1.read\_until(b"Remplazo1#")

tn\_r1.close()

print('R1: telnet connection is close')

'''

SEND AND RECIVE RUNNIG CONFIG FOR ROUTER R2

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

tn\_r2 = telnetlib.Telnet(HOST\_R2) # connecto to r2

tn\_r2.read\_until(b"Password: ")

tn\_r2.write(password.encode('ascii') + b"\r") # write password telnet

print("You are connected to {}".format(HOST\_R2))

tn\_r2.read\_until(b"R2>")

tn\_r2.write('enable'.encode('ascii') + b"\r") # write enable command

tn\_r2.read\_until(b"Password: ")

tn\_r2.write(password\_enable.encode('ascii') + b"\r") # write password enable

print('R2: "enable" mode activated')

tn\_r2.read\_until(b"R2#")

tn\_r2.write('copy running-config tftp:'.encode('ascii') + b"\r") # send runnig config to tftp server in linux

tn\_r2.read\_until(b"Address or name of remote host []? ")

tn\_r2.write(IP\_TFTP.encode('ascii') + b"\r") # set server tftp address

tn\_r2.read\_until(b"Destination filename [r2-confg]? ")

tn\_r2.write('backup2'.encode('ascii') + b"\r") # set file name

time.sleep(6) # wait to send file

print('R2: runnig-config send to tftp server')

path = os.path.join(TFTP\_DIR, 'backup2')

file = open(path, 'r');

# Read operation

lines = []

for line in file.readlines():

if 'R2' in line:

lines.append(line.replace('R2', 'Remplazo2'))

else:

lines.append(line)

file.close()

# Write operation

new\_file = open(path, 'w+')

new\_file.writelines(lines)

new\_file.close()

print('R2: hostname R2 is update to Remplazo2')

tn\_r2.read\_until(b"R2#")

tn\_r2.write('copy tftp: running-config'.encode('ascii') + b"\r") # send runnig config to router in gns3

tn\_r2.read\_until(b"Address or name of remote host []? ")

tn\_r2.write(IP\_TFTP.encode('ascii') + b"\r") # set tftp server ip address

tn\_r2.read\_until(b"Source filename []? ")

tn\_r2.write('backup2'.encode('ascii') + b"\r") # set filename of backup

tn\_r2.read\_until(b"Destination filename [running-config]? ")

tn\_r2.write(b"\r") # set return key to continue

print('R2: running-config modified send to router R2')

tn\_r2.read\_until(b"Remplazo2#")

tn\_r2.write('write'.encode('ascii') + b"\r") # save config

print('R2: configuration is saved')

tn\_r2.read\_until(b"Remplazo2#")

tn\_r2.close()

print('R2: telnet connection is close')

'''

SEND AND RECIVE RUNNIG CONFIG FOR ROUTER R3

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

tn\_r3 = telnetlib.Telnet(HOST\_R3) # connecto to r3

tn\_r3.read\_until(b"Password: ")

tn\_r3.write(password.encode('ascii') + b"\r") # write password telnet

print("You are connected to {}".format(HOST\_R3))

tn\_r3.read\_until(b"R3>")

tn\_r3.write('enable'.encode('ascii') + b"\r") # write enable command

tn\_r3.read\_until(b"Password: ")

tn\_r3.write(password\_enable.encode('ascii') + b"\r") # write password enable

print('R3: "enable" mode activated')

tn\_r3.read\_until(b"R3#")

tn\_r3.write('copy running-config tftp:'.encode('ascii') + b"\r") # send runnig config to tftp server in linux

tn\_r3.read\_until(b"Address or name of remote host []? ")

tn\_r3.write(IP\_TFTP.encode('ascii') + b"\r") # set server tftp address

tn\_r3.read\_until(b"Destination filename [r3-confg]? ")

tn\_r3.write('backup3'.encode('ascii') + b"\r") # set file name

time.sleep(6) # wait to send file

print('R3: runnig-config send to tftp server')

path = os.path.join(TFTP\_DIR, 'backup3')

file = open(path, 'r');

# Read operation

lines = []

for line in file.readlines():

if 'R3' in line:

lines.append(line.replace('R3', 'Remplazo3'))

else:

lines.append(line)

file.close()

# Write operation

new\_file = open(path, 'w+')

new\_file.writelines(lines)

new\_file.close()

print('R3: hostname R3 is update to Remplazo3')

tn\_r3.read\_until(b"R3#")

tn\_r3.write('copy tftp: running-config'.encode('ascii') + b"\r") # send runnig config to router in gns3

tn\_r3.read\_until(b"Address or name of remote host []? ")

tn\_r3.write(IP\_TFTP.encode('ascii') + b"\r") # set tftp server ip address

tn\_r3.read\_until(b"Source filename []? ")

tn\_r3.write('backup3'.encode('ascii') + b"\r") # set filename of backup

tn\_r3.read\_until(b"Destination filename [running-config]? ")

tn\_r3.write(b"\r") # set return key to continue

print('R3: running-config modified send to router R3')

tn\_r3.read\_until(b"Remplazo3#")

tn\_r3.write('write'.encode('ascii') + b"\r") # save config

print('R3: configuration is saved')

tn\_r3.read\_until(b"Remplazo3#")

tn\_r3.close()

print('R3: telnet connection is close')

'''

SEND AND RECIVE RUNNIG CONFIG FOR ROUTER R4

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'''

tn\_r4 = telnetlib.Telnet(HOST\_R4) # connecto to r4

tn\_r4.read\_until(b"Password: ")

tn\_r4.write(password.encode('ascii') + b"\r") # write password telnet

print("You are connected to {}".format(HOST\_R4))

tn\_r4.read\_until(b"R4>")

tn\_r4.write('enable'.encode('ascii') + b"\r") # write enable command

tn\_r4.read\_until(b"Password: ")

tn\_r4.write(password\_enable.encode('ascii') + b"\r") # write password enable

print('R4: "enable" mode activated')

tn\_r4.read\_until(b"R4#")

tn\_r4.write('copy running-config tftp:'.encode('ascii') + b"\r") # send runnig config to tftp server in linux

tn\_r4.read\_until(b"Address or name of remote host []? ")

tn\_r4.write(IP\_TFTP.encode('ascii') + b"\r") # set server tftp address

tn\_r4.read\_until(b"Destination filename [r4-confg]? ")

tn\_r4.write('backup4'.encode('ascii') + b"\r") # set file name

time.sleep(6) # wait to send file

print('R4: runnig-config send to tftp server')

path = os.path.join(TFTP\_DIR, 'backup4')

file = open(path, 'r');

# Read operation

lines = []

for line in file.readlines():

if 'R4' in line:

lines.append(line.replace('R4', 'Remplazo4'))

else:

lines.append(line)

file.close()

# Write operation

new\_file = open(path, 'w+')

new\_file.writelines(lines)

new\_file.close()

print('R4: hostname R4 is update to Remplazo4')

tn\_r4.read\_until(b"R4#")

tn\_r4.write('copy tftp: running-config'.encode('ascii') + b"\r") # send runnig config to router in gns3

tn\_r4.read\_until(b"Address or name of remote host []? ")

tn\_r4.write(IP\_TFTP.encode('ascii') + b"\r") # set tftp server ip address

tn\_r4.read\_until(b"Source filename []? ")

tn\_r4.write('backup4'.encode('ascii') + b"\r") # set filename of backup

tn\_r4.read\_until(b"Destination filename [running-config]? ")

tn\_r4.write(b"\r") # set return key to continue

print('R4: running-config modified send to router R4')

tn\_r4.read\_until(b"Remplazo4#")

tn\_r4.write('write'.encode('ascii') + b"\r") # save config

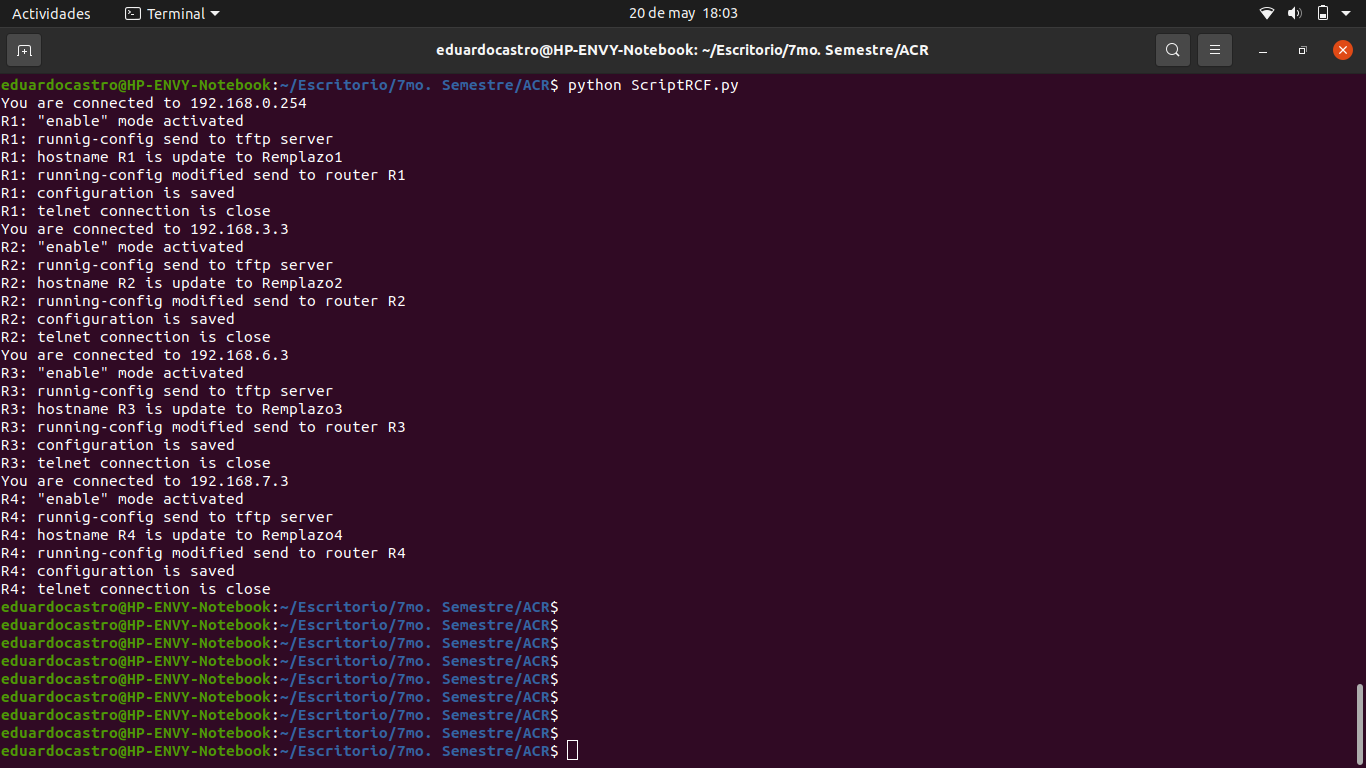
print('R4: configuration is saved')

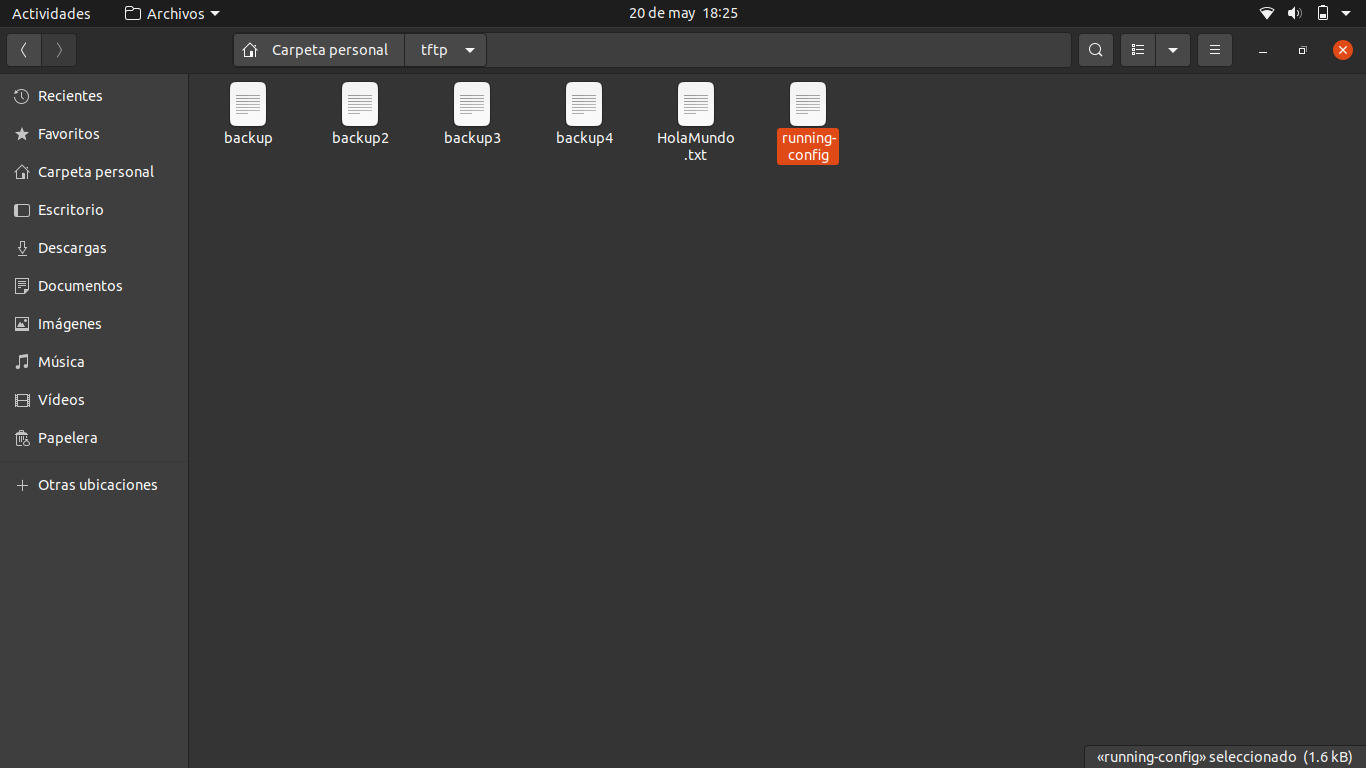
tn\_r4.read\_until(b"Remplazo4#")

tn\_r4.close()

print('R4: telnet connection is close')

1. **Capturas de pantalla de la ejecución del Script**





**Conclusiones**

* **Castro Cruces Jorge Eduardo**

En conclusión, puedo decir que el objetivo principal se cumplió, el cual era desarrollar un manual de usuario que muestre paso a paso el procedimiento que realizamos para concluir la práctica. Sin duda, la aplicación GNS3, es bastante parecida a Packet Tracer de CISCO, con la gran diferencia de que en GNS3 se pueden incluir Máquinas Virtuales. Tengo que reconocer que es bastante complicado aprender el correcto funcionamiento y manejo de las Máquinas Virtuales vinculadas a GNS3, porque un pequeño error que cometas, y el archivo de tu topología se arruina permanentemente sin manera de recuperarla, ni siquiera los Snapshots guardados. Tampoco deja guardar Snapshots si hay una Máquina Virtual conectada dentro de la topología. Quedé bastante decepcionado de esta aplicación, espero no tener que usarla de nuevo en otras materias.

* **Guzman Pérez Oscar**

Considero que en esta práctica pudimos apreciar el funcionamiento de los semáforos y de la importancia de equilibrar no sólo el número de productores y consumidores, sino también considerar para ello el número de secciones críticas puesto que también influye y como en este caso puede representar un cuello de botella puesto que no hay un número equivalente de recursos que nos permita aprovechar todo el potencial de recursos computacionales de forma eficiente.

* **Frausto Hernández Omar**

Fue una práctica bastante interesante ya que fue mi primera vez utilizando Gns3, anteriormente ya había utilizado packet tracer y desde mi punto de vista es mucho mejor y más completo Gns3, lo cual me ayudó mucho para aclarar pequeñas cosas que no sabía o ya se me había olvidado.