

Ingeniería en Sistemas Computacionales

Materia: Fundamentos de Telecomunicaciones

Tema: 5 Fases Proyecto Telecomunicaciones

Alumno: Vargas Rodríguez Javier Jesús

Maestro: Ismael Jiménez Sánchez

Fecha De Entrega: 26/Octubre/2020

Horario: 5:00 pm – 6:00 pm

Fase 1: nstalar 2 centos8 en virtualbox usando vagrant

En esta fase, nosotros debemos de instalar la Centos8 por medio de los comandos, deben de ser 2 maquinas virtuales en la cual empezaremos a realizar nuestro proyecto y comenzar a realizar esta practica.

```
Windows PowerShell
PS C:\Users\KOTTO-PC\Desktop\centos> vagrant init centos/8
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
PS C:\Users\KOTTO-PC\Desktop\centos> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/8' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'centos/8'
default: URL: https://vagrantcloud.com/centos/8
==> default: Adding box 'centos/8' (v1905.1) for provider: virtualbox
default: Downloading: https://vagrantcloud.com/centos/boxes/8/versions/1905.1/providers/virtualbox.box
Download redirected to host: cloud.centos.org
default: Calculating and comparing box checksum...
default: Successfully added box 'centos/8' (v1905.1) for 'virtualbox'!
default: Importing base box 'centos/8'...
default: Matching MAC address for NAT networking...
default: Checking if box 'centos/8' version '1905.1' is up to date...
default: Setting the name of the VM: centos_default_1603675188909_11451
default: Fixed port collision for 22 => 2222. Now on port 2200.
default: Clearing any previously set network interfaces...
default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: Forwarding ports...
default: 22 (guest) => 2200 (host) (adapter 1)
default: Booting VM...
default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2200
default: SSH username: vagrant
default: SSH auth method: private key
default: Vagrant insecure key detected. Vagrant will automatically replace
this with a newly generated keypair for better security.
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
```

The screenshot shows a Windows desktop with a Windows PowerShell terminal window and the Oracle VM VirtualBox Administrator window. The PowerShell window shows the commands and output for installing two CentOS 8 VMs using Vagrant. The VirtualBox Administrator window shows the configuration for the two VMs, including the name, system operating system, memory, and storage.

Windows PowerShell

```
PS C:\Users\KOTTO-PC\Desktop\centos2> vagrant init centos/8
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
PS C:\Users\KOTTO-PC\Desktop\centos2> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/8' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'centos/8'
default: URL: https://vagrantcloud.com/centos/8
==> default: Adding box 'centos/8' (v1905.1) for provider: virtualbox
default: Downloading: https://vagrantcloud.com/centos/boxes/8/versions/1905.1/providers/virtualbox.box
Download redirected to host: cloud.centos.org
default: Calculating and comparing box checksum...
default: Successfully added box 'centos/8' (v1905.1) for 'virtualbox'!
default: Importing base box 'centos/8'...
default: Matching MAC address for NAT networking...
default: Checking if box 'centos/8' version '1905.1' is up to date...
default: Setting the name of the VM: centos2_default_1603675188909_11451
default: Fixed port collision for 22 => 2222. Now on port 2200.
default: Clearing any previously set network interfaces...
default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: Forwarding ports...
default: 22 (guest) => 2222 (host) (adapter 1)
default: Booting VM...
default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default: Vagrant insecure key detected. Vagrant will automatically replace
this with a newly generated keypair for better security.
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: This is not an error message; everything may continue to work properly,
default: in which case you may ignore this message.
default: Rsyncing folder: /cygdrive/c/Users/KOTTO-PC/Desktop/centos2/ => /vagrant
PS C:\Users\KOTTO-PC\Desktop\centos2>
```

Oracle VM VirtualBox Administrator

Archivo Máquina Ayuda

Herramientas

centos_default_1603675... Corriendo

centos2_default_1... Corriendo

General

Nombre: centos2_default_1603675188909_11451

Sistema operativo: Red Hat (64-bit)

Sistema

Memoria base: 512 MB

Orden de arranque: Disquete, Óptica, Disco duro

Aceleración: VT-x/AMD-V, Paginación anidada, PAE/Nx, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB

Controlador gráfico: VBoxVGA

Servidor de escritorio remoto: Inhabilitado

Grabación: Inhabilitado

Almacenamiento

Controlador: IDE

IDE primario maestro: CentOS-8-Vagrant-8.0.1905-1.x86_64.vmdk (Normal, 10.00 GB)

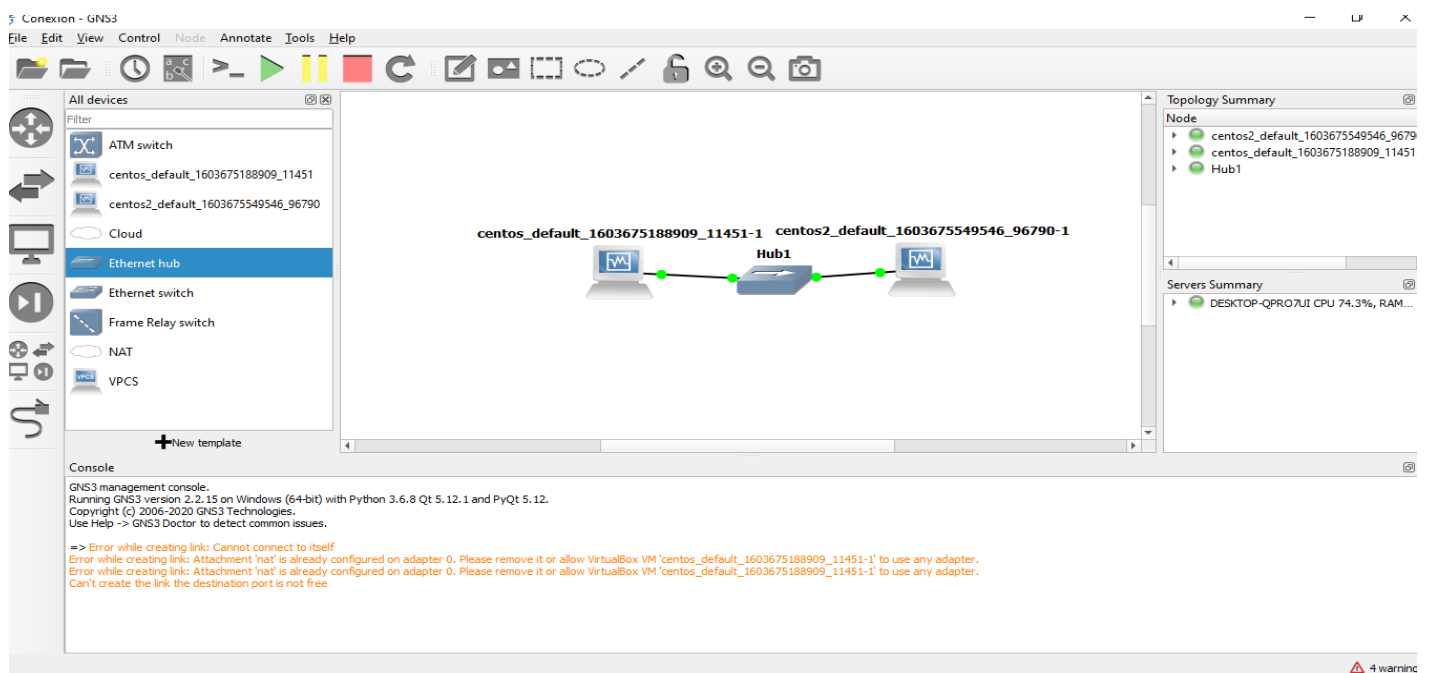
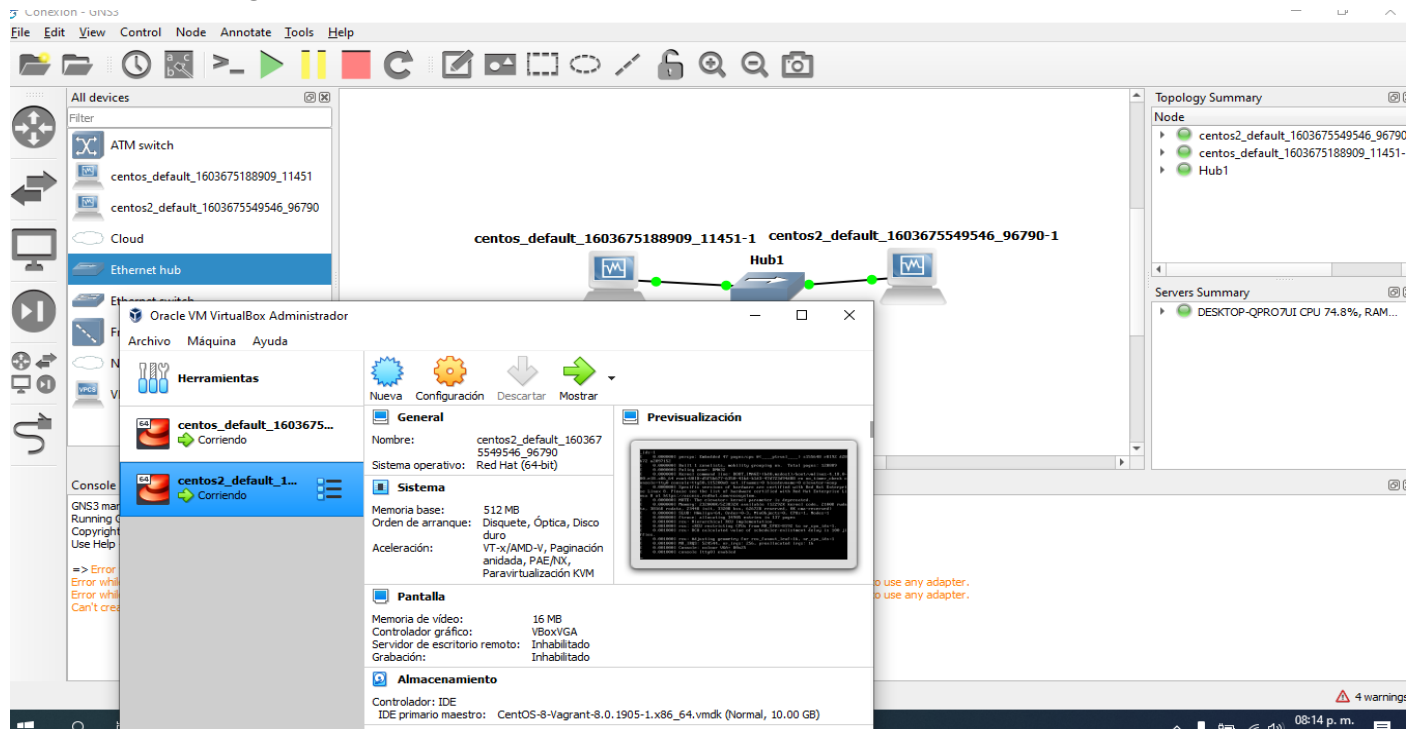
Audio

Controlador de anfitrión: Windows DirectSound

Controlador: ICH AC97

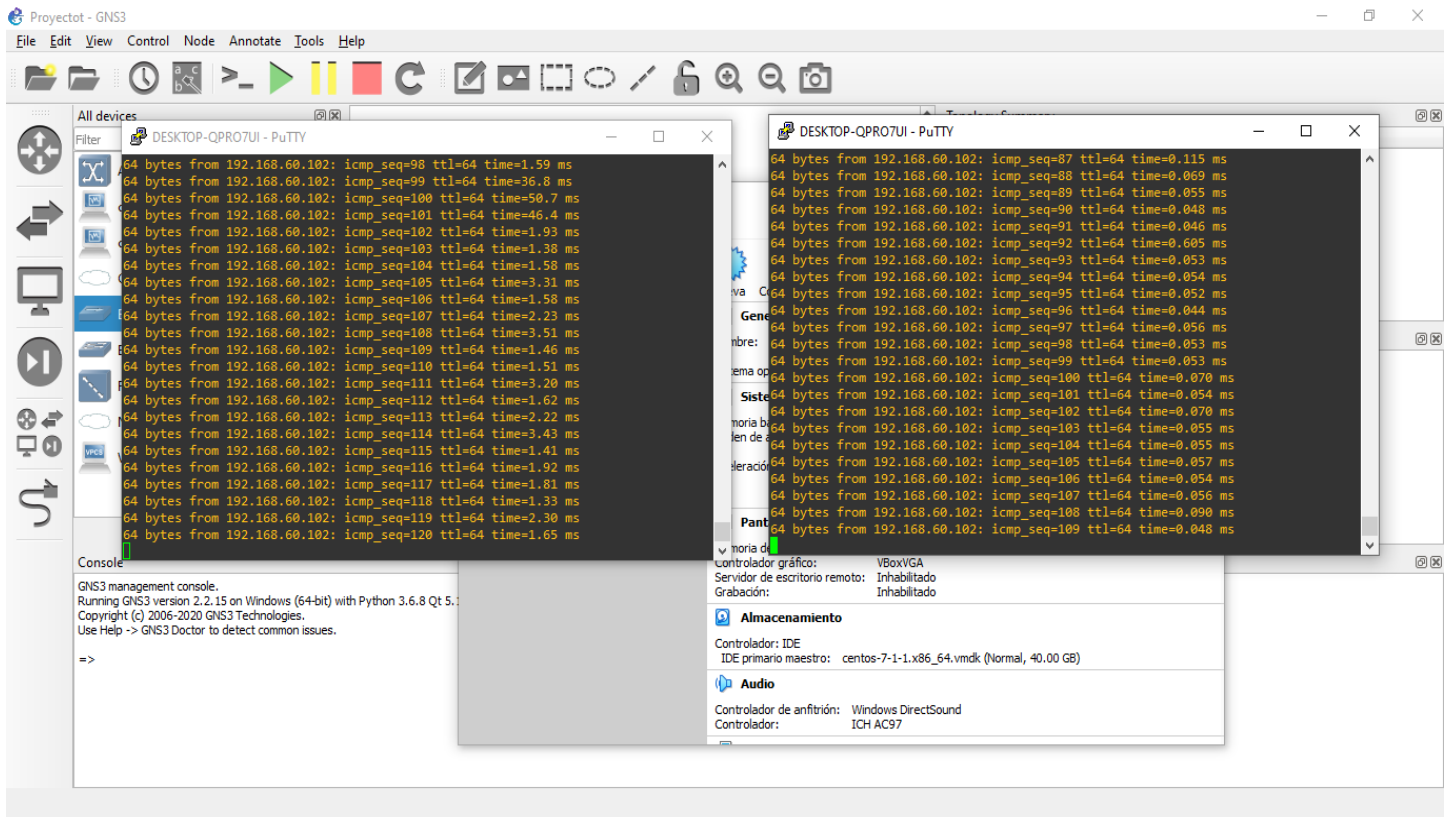
Fase 2: Conectar en GNS3, las dos VMs de CentOS con un switch ethernet.

Ahora mediante el programa GNS3 vamos a conectar esas 2 maquinas virtuales que ya creamos anteriormente mediante el Centos, entonces las conectamos y nos debe de agarrar sin ningun problema.

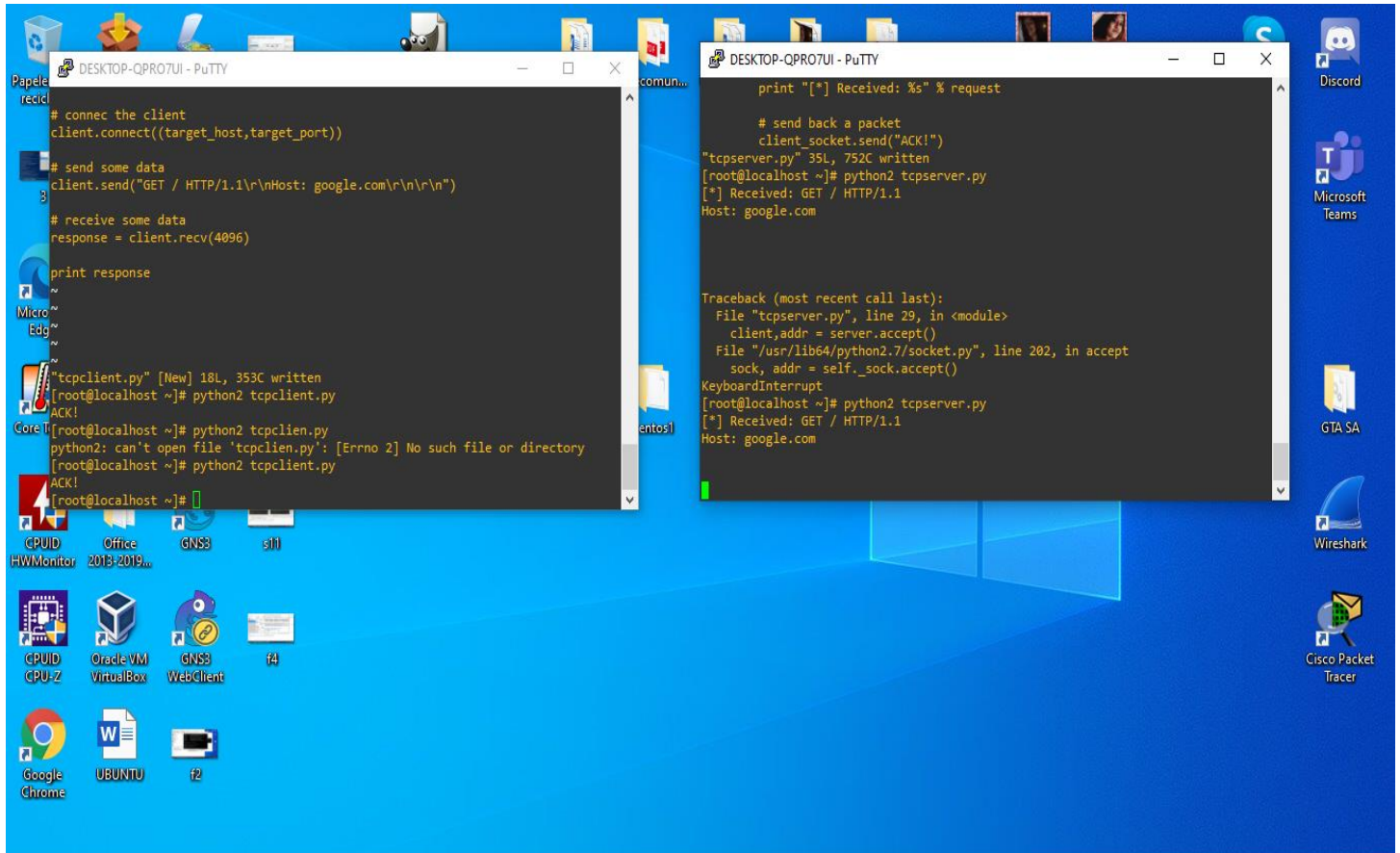


Fase 3: Usar los scripts de python para conectar las dos VMs usando sockets.

En esta fase vamos a comprobar que nuestras IP de Cliente y Servidor esten funcionando de manera correcta, entonces ya con el python instalado iremos comprobando cada uno de estos para ver que si esten conectadas las maquinas virtuales.



Ahora vamos a poner los codigos que nos puso el profesor en el Discord que viene siendo el codigo de Python para que nuestro Cliente-Servidor muestre la comunicación que tiene estas en las maquinas virtuales.



The screenshot shows a Windows desktop with a blue background. Two PuTTY windows are open, both titled "DESKTOP-QPRO7UI - PuTTY".

The left window shows the following code and output:

```
# connect the client
client.connect((target_host,target_port))

# send some data
client.send("GET / HTTP/1.1\r\nHost: google.com\r\n\r\n")

# receive some data
response = client.recv(4096)

print response
```

The output shows the file "tcpclient.py" being written and then an error message:

```
[root@localhost ~]# python2 tcpclient.py
python2: can't open file 'tcpclient.py': [Errno 2] No such file or directory
[root@localhost ~]# python2 tcpclient.py
ACK!
[root@localhost ~]#
```

The right window shows the following code and output:

```
print "[*] Received: %s" % request

# send back a packet
client_socket.send("ACK!")

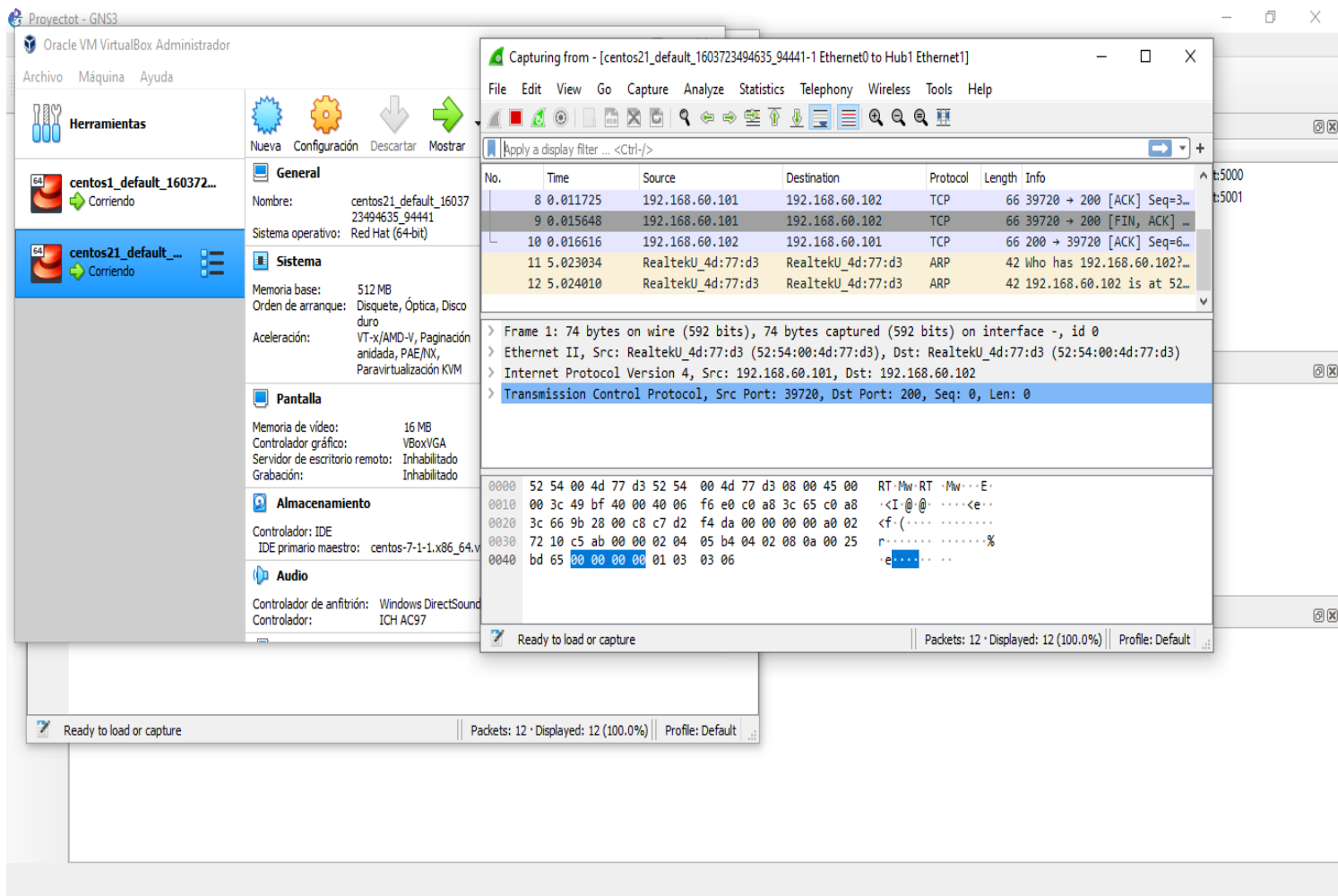
"tcpserver.py" 35L, 752C written
[root@localhost ~]# python2 tcpserver.py
[*] Received: GET / HTTP/1.1
Host: google.com
```

The output shows a traceback error:

```
Traceback (most recent call last):
  File "tcpserver.py", line 29, in <module>
    client,addr = server.accept()
  File "/usr/lib64/python2.7/socket.py", line 202, in accept
    sock, addr = self._sock.accept()
KeyboardInterrupt
[root@localhost ~]# python2 tcpserver.py
[*] Received: GET / HTTP/1.1
Host: google.com
```

Fase 4 - Capturar el tráfico de la comunicación entre las dos VMs al momento de utilizar los scripts.

Ya en esta fase solo es captura el trafico de datos de información que contiene ya nuestras maquinas conectadas, por lo cual gracias al Wiresharck se podrá apreciar de manera correcta si es que estas al final si se pudieron realizar lo esperado.



Fase 5 – Reporte de conclusiones:

Ya como podemos finalizar de este trabajo, es que viene siendo un poco complejo realizar este tipo de actividades en diversas funciones, pero también es demasiado interesante como podemos observar el tráfico de los datos y además de conectar las dos máquinas virtuales para trabajar en conjunto con Cliente-Servidor.

Gracias a este proyecto aprendimos a como usar Vagrant, instalar las Centos8 para poder levantar nuestras máquinas virtuales, también como se menciono anteriormente gracias al Wireshark supimos como es el tráfico de datos que se envían y reciben, en parte si es complicado al principio porque hay que estar super atentos a cada comando que estemos escribiendo, también de saber identificar cual será nuestro Cliente y cual será el Servidor y así poder estar trabajando en conjunto con estos dos.

Para finalizar esta conclusión se aprendió mucho de esta práctica, nos reforzo los conocimientos que hemos estado viendo en clases con el profesor y así seguir trabajando de mejor manera para poder desenvolvernos mejor en cualquier practica o trabajo que nos deje.