



Nombre de la Materia:

Fundamentos de Telecomunicaciones

Aula

Nombre de la Licenciatura:

Ing. Sistemas Computacionales.

Nombre del Alumno(a):

Pool Ramírez Miguel Ángel.

Número de Control:

18530437.

Nombre de la Tarea:

Proyecto Sistema de Comunicación

Unidad #1

Nombre de la Unidad: Sistema de Comunicación.

Nombre del Profesor(a):

Ing. Ismael Jiménez Sánchez

Fecha: 28/10/20

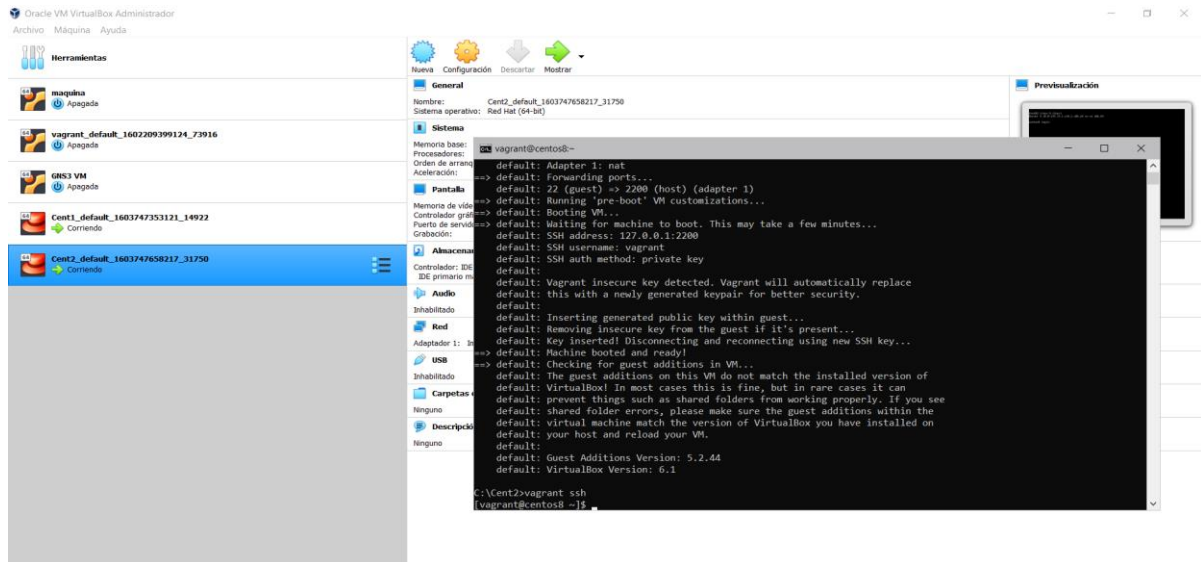


TECNOLÓGICO
NACIONAL DE MÉXICO

PROYECTO SISTEMA DE COMUNICACIÓN

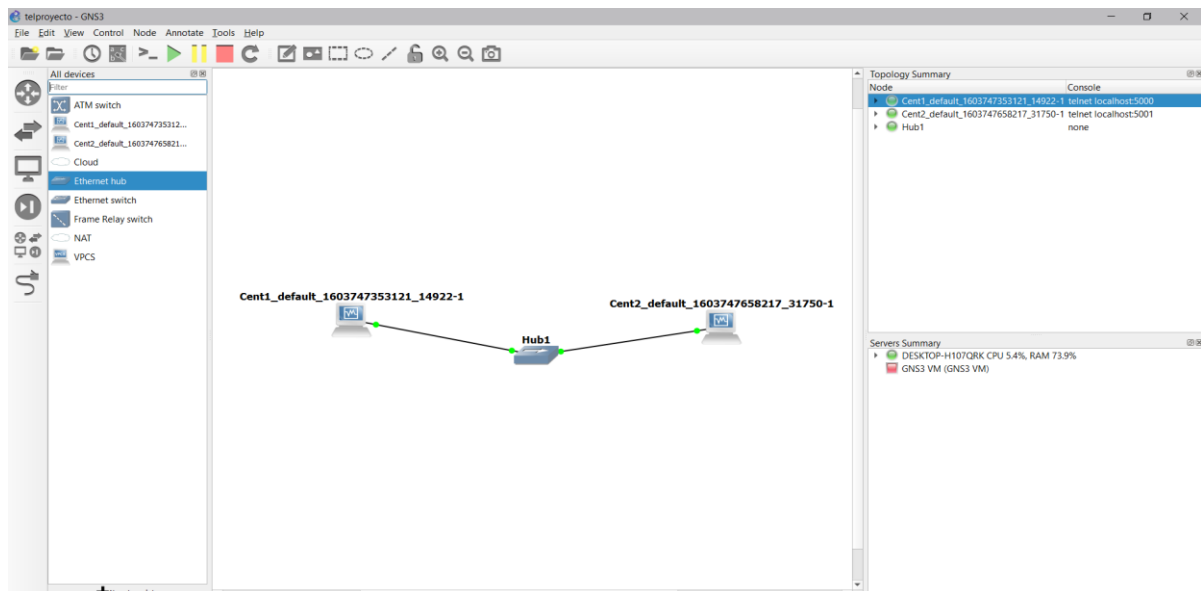
1 INSTALAR 2 MÁQUINAS VIRTUALES CON VAGRANT A PARTIR DE BOXES.

Se utilizará vagrant y boxes a través del símbolo de sistema para levantar dos máquinas virtuales (Centos 8) que son con las que trabajaremos.



2 INGRESAR LAS MÁQUINAS VIRTUALES A GNS3

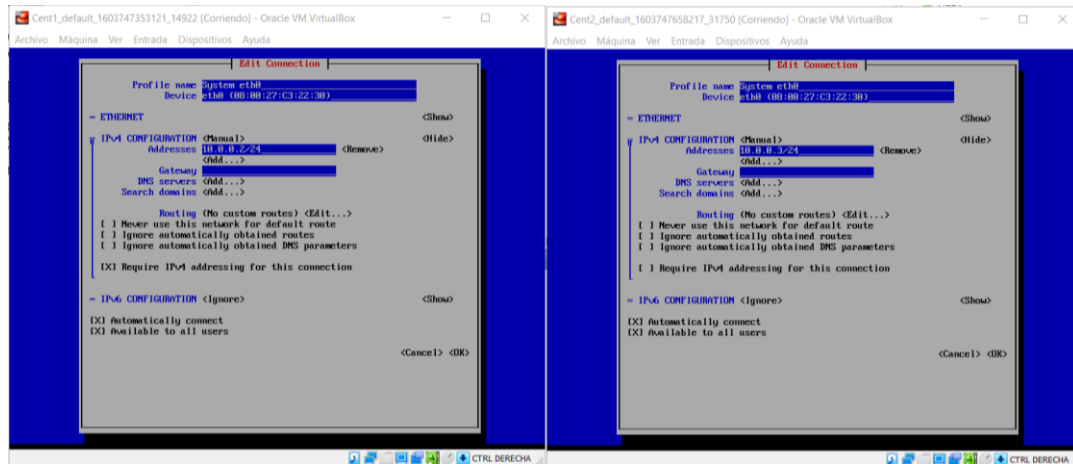
Añadiremos cada máquina virtual modificando su tipo de consola a telnet y conectándolas a un ethernet hub.



3 CONFIGURACIÓN DE DIRECCIONES IP

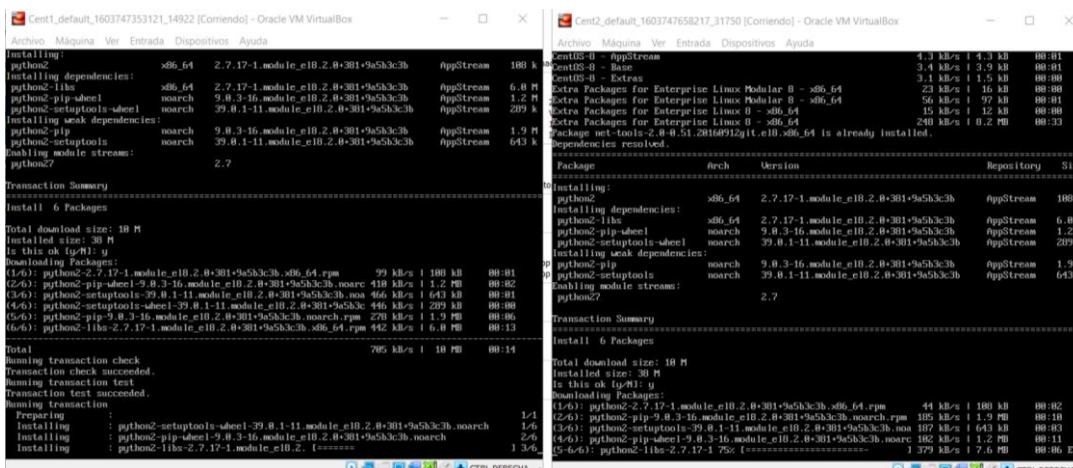
Se le asignara una dirección IP a cada máquina, una teniendo la función del servidor y la

otra el cliente.



4 INSTALACIÓN DE PYTHON Y USO DE SCRIPTS PARA COMUNICAR AMBAS MÁQUINAS VIRTUALES.

Se instalara python 2 dentro de ambas maquinas virtuales y seguidamente se implementaran los scripts de server y y cliente.



5 CONFIGURAR DIRECCIONES IP Y HACER PING PARA RECIBIR RESPUESTAS Y COMPROBAR COMUNICACIÓN.

El hacer ping nos proporcionara una respuesta de comunicación entre la maquina cliente y la maquina server.

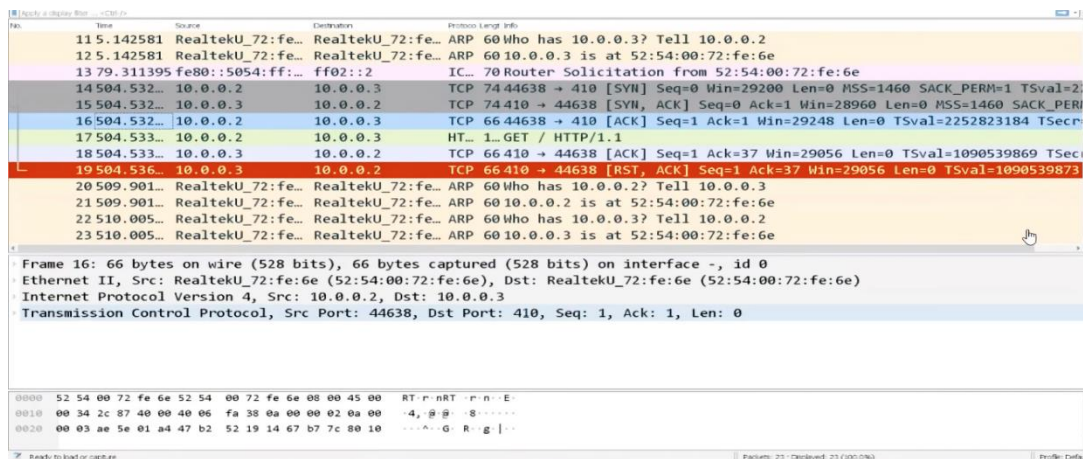
```

valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:72:fe:6e brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 brd 10.0.0.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe72:fe6e/64 scope link dadfailed tentative
        valid_lft forever preferred_lft forever
[root@localhost vagrant]# ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data:
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=1.51 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.736 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.736 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.652 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.605 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.696 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=0.726 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=0.725 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=0.667 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=0.647 ms
64 bytes from 10.0.0.3: icmp_seq=11 ttl=64 time=0.687 ms
64 bytes from 10.0.0.3: icmp_seq=12 ttl=64 time=0.659 ms
64 bytes from 10.0.0.3: icmp_seq=13 ttl=64 time=0.637 ms

```

6 VERIFICAR TRÁFICO DE RED A TRAVÉS DE WIRESHARK.

Esta herramienta nos permitirá ver el tráfico de comunicación a través de nuestras maquinas cliente – servidor, devolviendo una respuesta.



CONCLUSIONES.

Se observó el tráfico de red gracias a la herramienta wireshark en la cual nosotros tendremos un receptor y un emisor, al iniciar la comunicación iniciara con la bandera SYN esto nos indica que se ha enviado un paquete de datos, seguidamente podremos observar la bandera ACK por lo que podemos darnos cuenta que el paquete de datos ha sido recibido, para confirmar esto una de las maquinas volverá a reenviar la bandera ACK para confirmar que el paquete ha sido recibido, esta bandera tendría una función de confirmación