



# INSTITUTO POLITÉCNICO NACIONAL

## ESCUELA SUPERIOR DE CÓMPUTO

### DESARROLLO DE SISTEMAS DISTRIBUIDOS

---

## Tarea 10. Replicación de un servidor en la nube

---

*Alumno:*

Daniela Cortés Castillo

*Profesor:*

Carlos Pineda Guerrero

4CV1

11 de enero de 2021

# Índice

<b>1. Descripción</b>	<b>4</b>
<b>2. Desarrollo</b>	<b>4</b>
2.1. Creación máquinas virtuales . . . . .	4
2.2. Abrir puerto 50000 . . . . .	7
2.3. Conectar máquina virtual 1 . . . . .	9
2.4. Instalar jdk8 en máquina virtual 1 . . . . .	10
2.5. Enviar a la máquina virtual 1 archivos . . . . .	10
2.6. Editar main de archivo Servidor2.java en máquina virtual 1 . . . . .	11
2.7. Compilar programas en máquina 1 . . . . .	12
2.8. Conectar máquina virtual 2 . . . . .	12
2.9. Instalar jdk en la máquina virtual 2 . . . . .	13
2.10. Enviar a la máquina virtual 2 archivos . . . . .	14
2.11. Editar main de archivo Servidor2.java en máquina virtual 2 . . . . .	14
2.12. Compilar programa en máquina 2 . . . . .	15
2.13. Ejecutar el programa en máquina virtual 2 . . . . .	15
2.14. Ejecutar los programas en máquina virtual 1 . . . . .	15
2.15. En Ubuntu . . . . .	16
2.15.1. Editar el programa cliente . . . . .	16
2.15.2. Compilar programa . . . . .	17
2.15.3. Ejecutar el programa . . . . .	17

## Índice de figuras

1.	Datos básicos de máquina virtual . . . . .	5
2.	Datos básicos de máquina virtual . . . . .	5
3.	Administración de máquina virtual . . . . .	6
4.	Crear máquina virtual . . . . .	6
5.	Máquina virtual uno, resaltando la IP pública que usaremos después . . . . .	7
6.	Máquina virtual dos, resaltando la IP pública que usaremos después . . . . .	7
7.	Botón de redes en Portal de azure . . . . .	8
8.	Botón de regla de puerto de entrada . . . . .	8
9.	Regla de puerto de entrada. . . . .	9
10.	Putty para abrir la máquina virtual 1 . . . . .	9
11.	Comando update en la máquina virtual 1 . . . . .	10
12.	Comando de instalación de JDK en la máquina virtual 1 . . . . .	10
13.	Comando open para ingresar a máquina virtual 1 con psftp.exe . . . . .	11
14.	Comando put para pasar archivos a máquina virtual 1 con psftp.exe . . . . .	11
15.	Comando ls para ver archivos en máquina virtual 1 . . . . .	11
16.	Cambio en main de programa en máquina virtual 1 . . . . .	12
17.	Compilar archivos en máquina virtual 1 . . . . .	12
18.	Putty para abrir la máquina virtual 2 . . . . .	13
19.	Comando update en la máquina virtual 2 . . . . .	13
20.	Comando de instalación de JDK en la máquina virtual 2 . . . . .	14
21.	Comando open y put para ingresar a máquina virtual 2 y mandar archivos con psftp.exe . . . . .	14
22.	Cambio en main de programa en máquina virtual 2 . . . . .	15
23.	Compilar archivo en máquina virtual 2 . . . . .	15
24.	Ejecutar archivo en máquina virtual 2 . . . . .	15
25.	Ejecutar archivos en máquina virtual 1 . . . . .	16
26.	Editar programa Cliente2.java . . . . .	16
27.	Correr y compilar programa Cliente2.java . . . . .	17
28.	Resultado en máquinas virtuales tras ejecutar Cliente2.java . . . . .	17

## 1. Descripción

En esta tarea vamos a realizar un ejercicio de replicación de un sistema completo, en este caso la replicación de un servidor TCP, tal como podría ser un servidor HTTP, un servidor de servicios web, un manejador de bases de datos, etc.

Para replicar un sistema, podemos crear una máquina virtual en la nube (réplica) que procese todas las peticiones que realizan los clientes, en paralelo al proceso de las mismas peticiones que realiza el sistema principal.

A continuación se muestra paso a paso el procedimiento descrito en la tarea.

## 2. Desarrollo

Los pasos a seguir para el desarrollo de la tarea son los siguientes:

### 2.1. Creación máquinas virtuales

Crear dos máquinas virtuales en la nube con Ubuntu 18 con 1 GB de RAM y disco HDD estándar. Para esto seguiremos los siguientes pasos:

- Iremos al portal de azure, y daremos click en "máquinas virtuales", después a agregar y comenzaremos la configuración de máquina virtual.
- En datos básicos, crearemos un nuevo grupo de recursos. Y pondremos el nombre de la máquina y llenaremos los campos con las especificaciones anteriormente mencionadas. Ver imágenes 1 y 2.
- En la parte de discos seleccionamos el HDD estándar.
- En administración deshabilitamos el diagnostico de arranque. Ver imagen 3.
- Revisamos la máquina y creamos. Ver imagen 4.

El procedimiento se tiene que repetir para ambas máquinas virtuales. Al final, en mi caso quedaron de la siguiente manera.(Figuras 5 y 6)

Figura 1: Datos básicos de máquina virtual

Figura 2: Datos básicos de máquina virtual

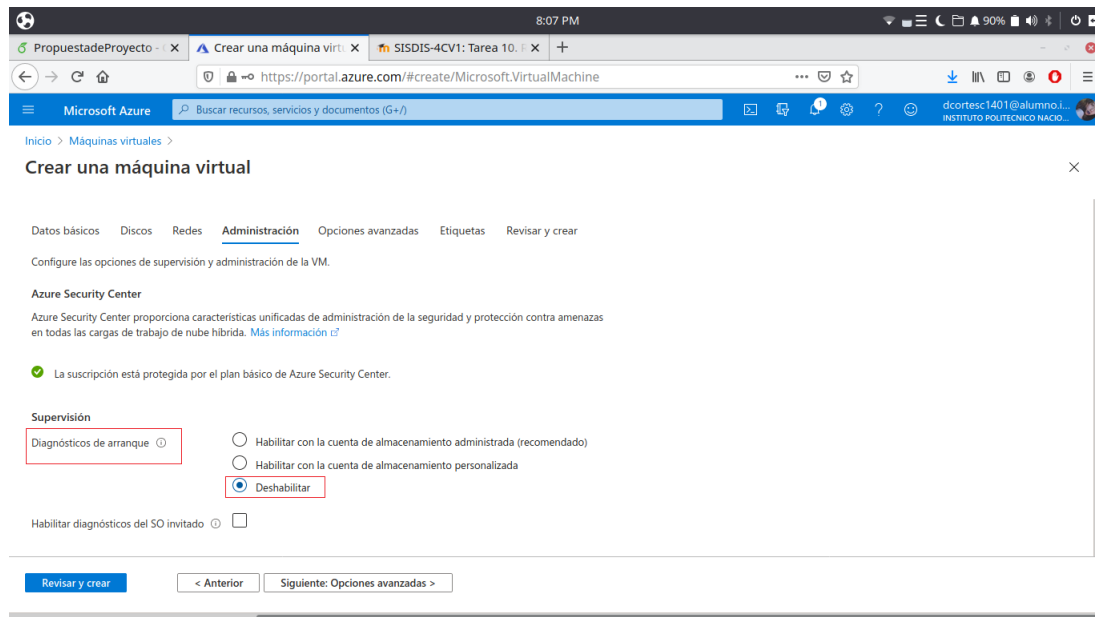


Figura 3: Administración de máquina virtual

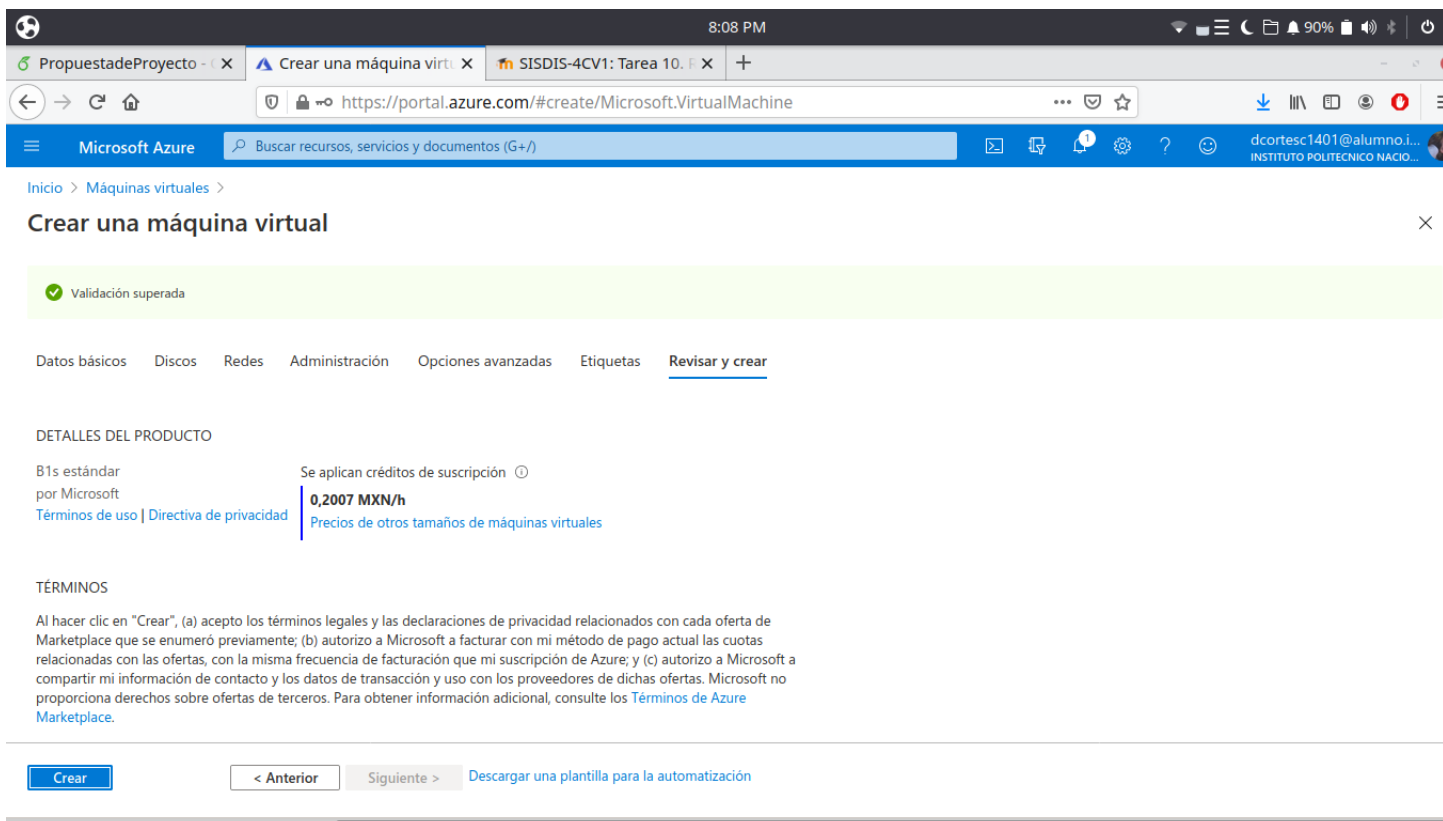


Figura 4: Crear máquina virtual

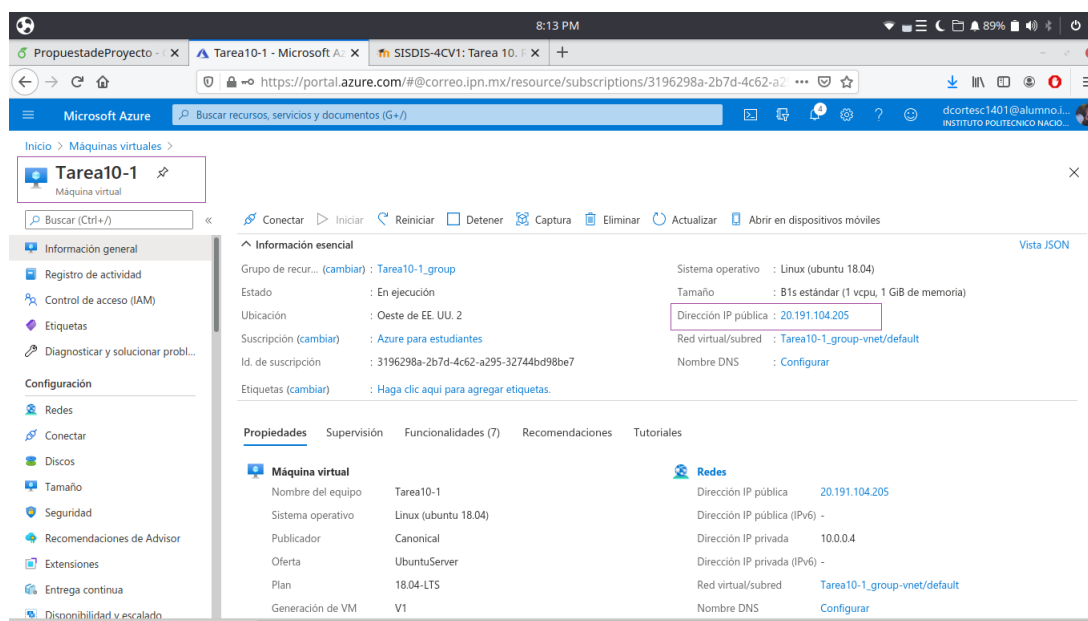


Figura 5: Máquina virtual uno, resaltando la IP pública que usaremos después

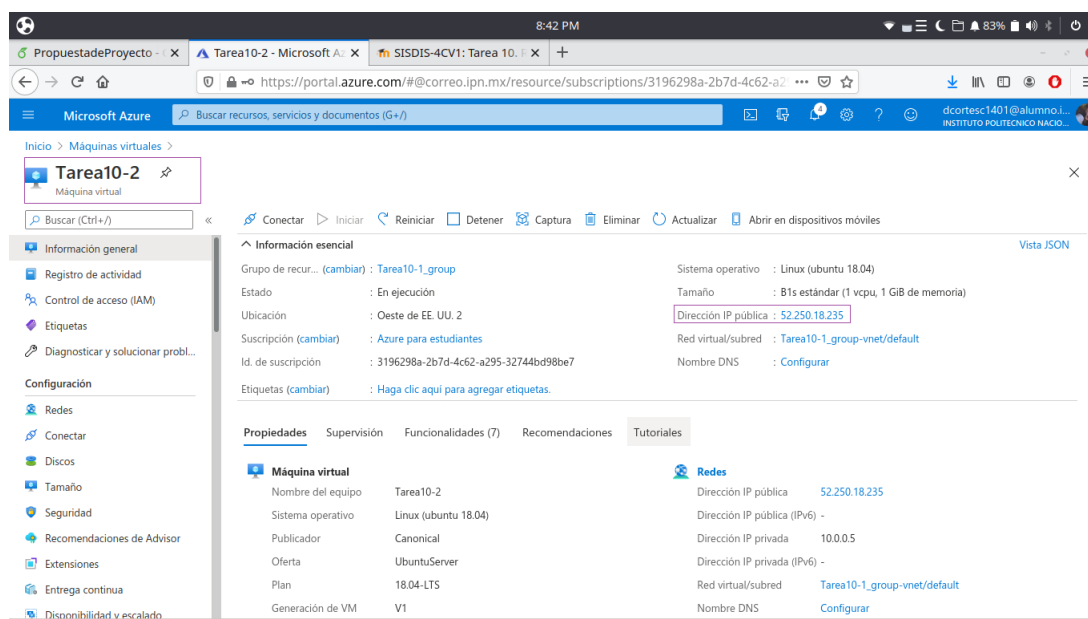


Figura 6: Máquina virtual dos, resaltando la IP pública que usaremos después

## 2.2. Abrir puerto 50000

Ahora en ambas máquinas abriremos el puerto 50000 para TCP, el procedimiento es dar clic en la sección de redes (figura 7), después al botón de agregar regla de puerto d entrada (figura 8), y llenar las casillas correspondientes (figura 9).

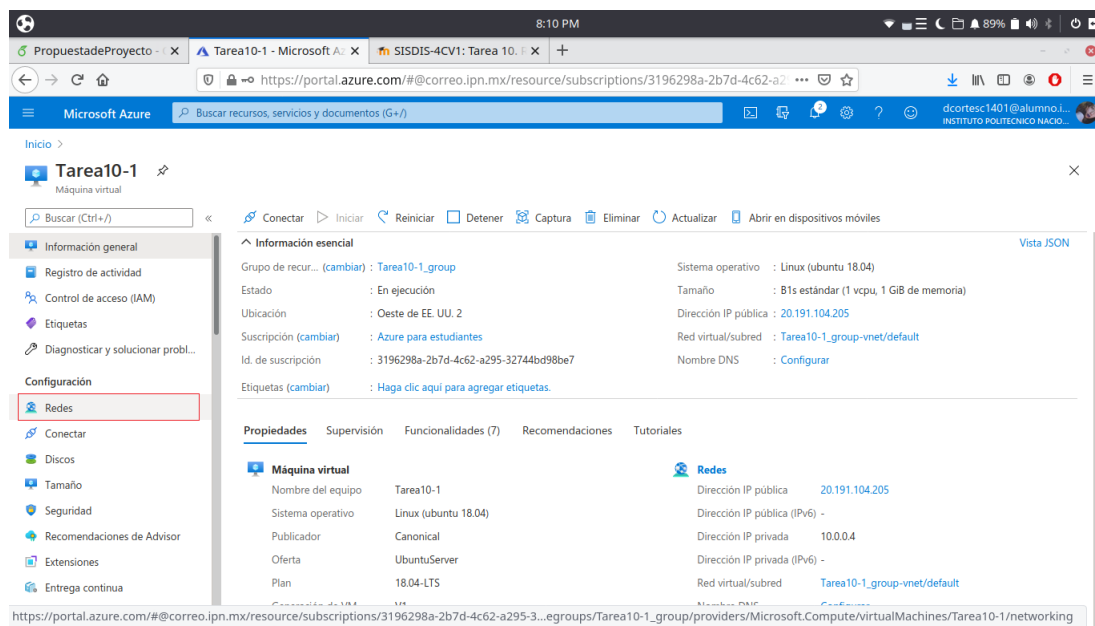


Figura 7: Botón de redes en Portal de azure

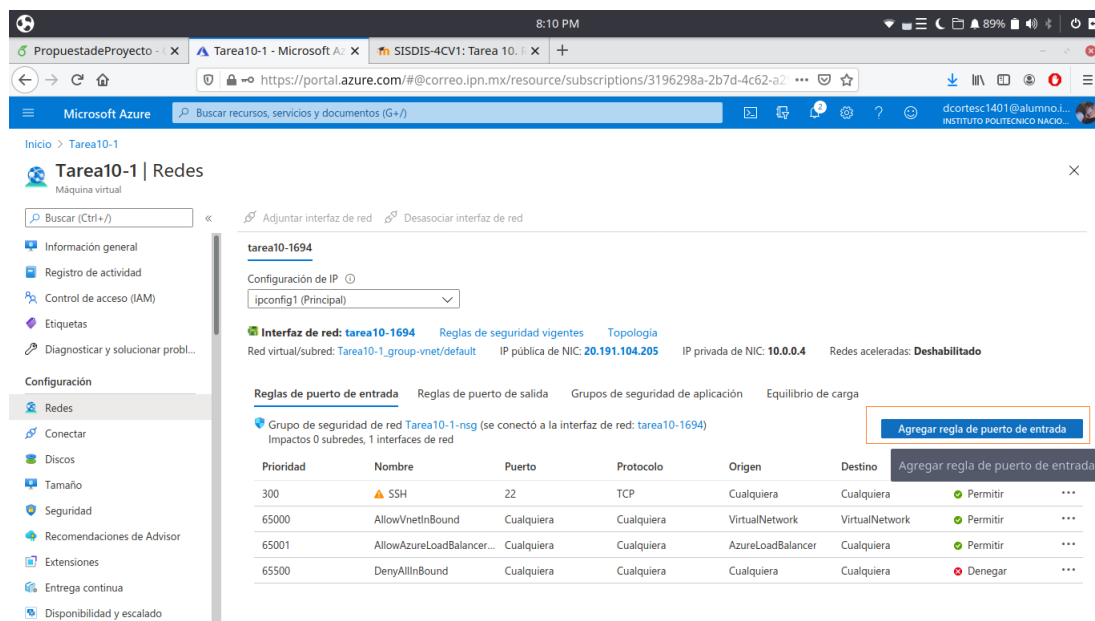


Figura 8: Botón de regla de puerto de entrada



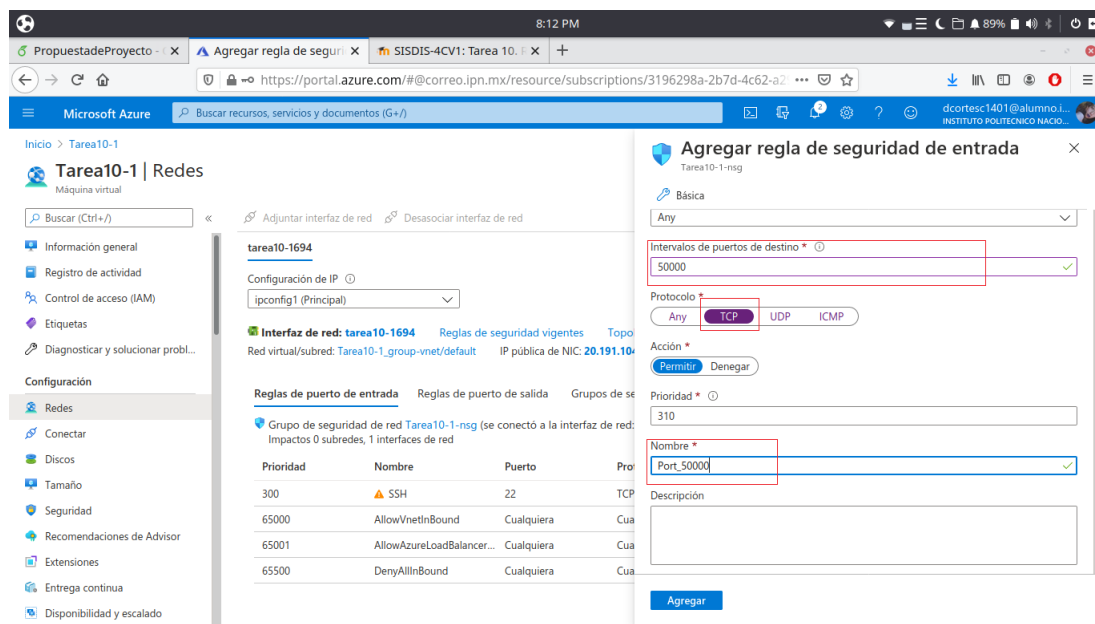


Figura 9: Regla de puerto de entrada.

## 2.3. Conectar máquina virtual 1

La máquina virtual 1 es el sistema principal. Para abrirla utilizo el programa putty.exe de la siguiente manera:

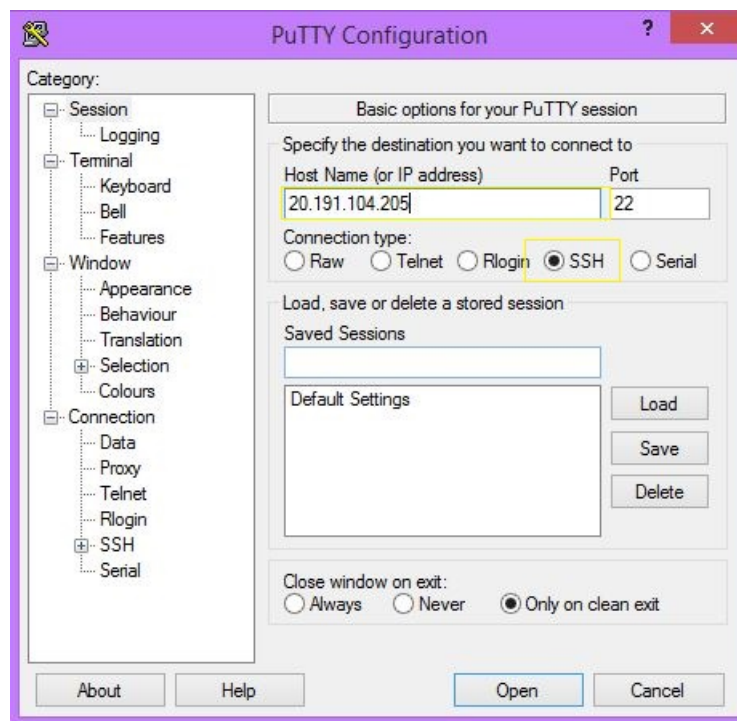


Figura 10: Putty para abrir la máquina virtual 1

Después de ingresar los datos, damos click en *open*, nos pedirá el usuario y contraseña,

después de esto, ya podemos usar la máquina virtual.

## 2.4. Instalar jdk8 en máquina virtual 1

Para instalar el jdk primero debemos actualizar el sistema (figura 11) y e instalaremos con el comando que se muestra en la figura 12

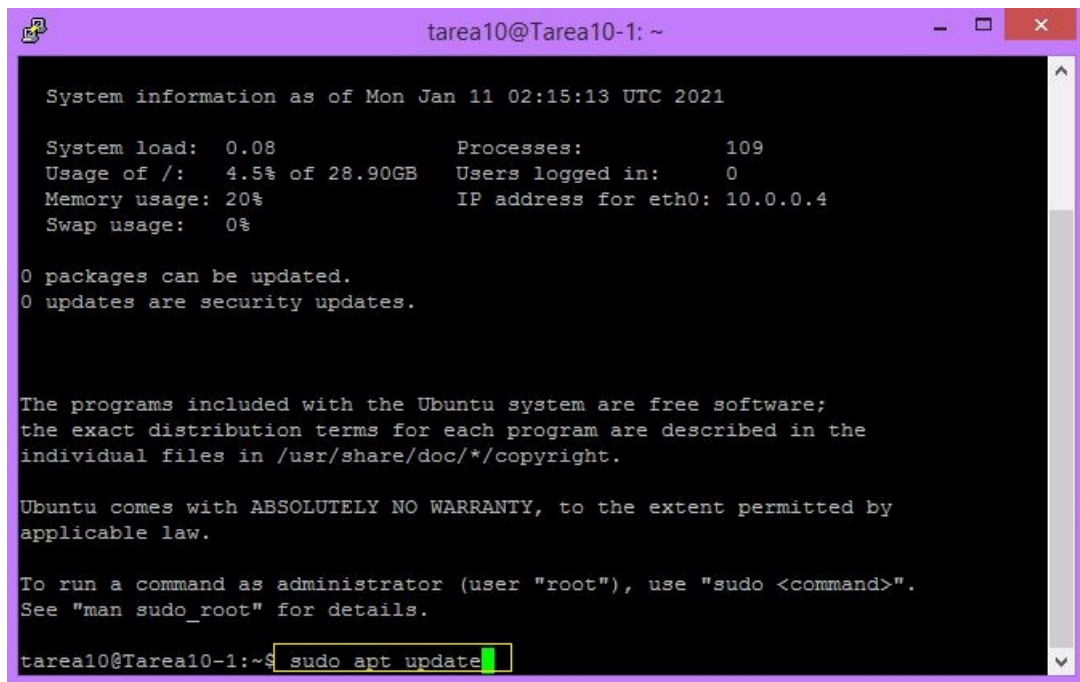
A terminal window titled 'tarea10@Tarea10-1: ~' with a purple title bar. The terminal output shows system information as of Mon Jan 11 02:15:13 UTC 2021. It lists system load (0.08), usage of / (4.5% of 28.90GB), memory usage (20%), swap usage (0%), processes (109), users logged in (0), and IP address for eth0 (10.0.0.4). It also states that 0 packages can be updated and 0 updates are security updates. A disclaimer about Ubuntu's warranty is shown. The prompt 'tarea10@Tarea10-1:~\$' is followed by the command 'sudo apt update' which is highlighted with a yellow box and a green cursor.

Figura 11: Comando update en la máquina virtual 1

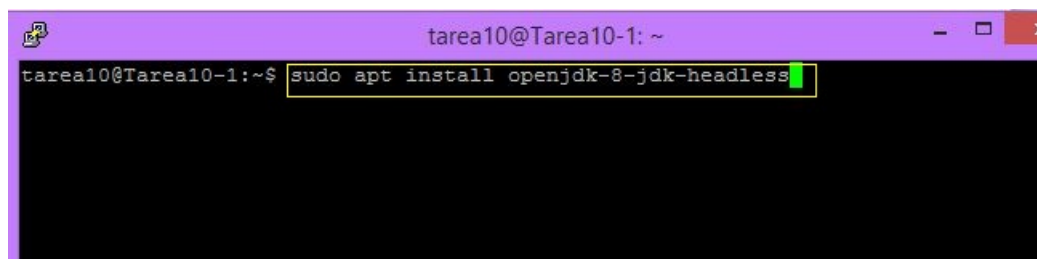
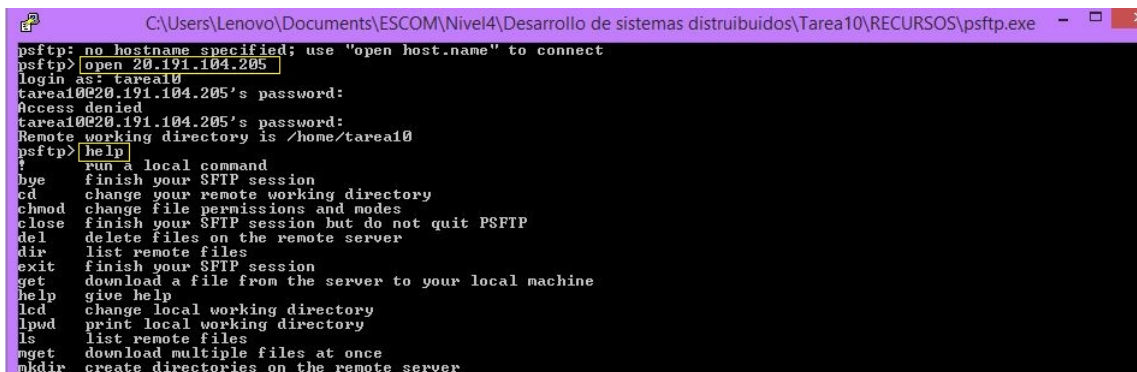
A terminal window titled 'tarea10@Tarea10-1: ~' with a purple title bar. The prompt 'tarea10@Tarea10-1:~\$' is followed by the command 'sudo apt install openjdk-8-jdk-headless' which is highlighted with a yellow box and a green cursor.

Figura 12: Comando de instalación de JDK en la máquina virtual 1

## 2.5. Enviar a la máquina virtual 1 archivos

Utilizando el programa **psftp.exe** enviar a la máquina virtual 1 los archivos: Servidor2.java y SimpleProxyServer.java Después de haber descargado el programa, lo abrimos e introducimos el comando de la figura 13, este lleva la dirección IP de la máquina que vamos a usar. Después de eso podemos ver que utilicé el comando **help**, este nos muestra las diferentes opciones de comandos. Después use el comando **put** para pasar el archivo da la máquina en la que trabajé,

como se muestra en la imagen 14. Nota, el comando `lpwd` nos muestra la dirección actual en la máquina.

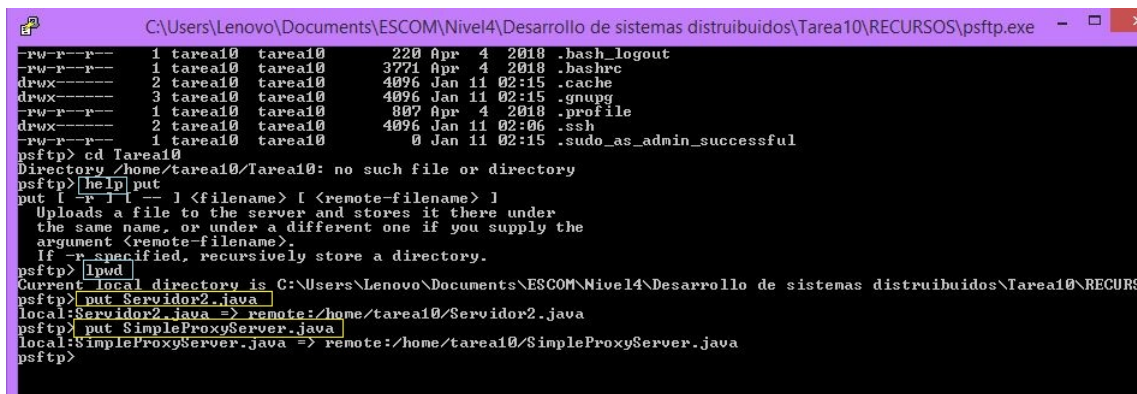


```

psftp> no hostname specified; use "open host.name" to connect
psftp> open 20.191.104.205
login as: tarea10
tarea10@20.191.104.205's password:
Access denied
tarea10@20.191.104.205's password:
Remote working directory is /home/tarea10
psftp> help
!      run a local command
bye    finish your SFTP session
cd     change your remote working directory
chmod  change file permissions and modes
close  finish your SFTP session but do not quit PSFTP
del    delete files on the remote server
dir    list remote files
exit   finish your SFTP session
get    download a file from the server to your local machine
help   give help
lcd    change local working directory
lpwd   print local working directory
ls     list remote files
mget   download multiple files at once
mkdir  create directories on the remote server

```

Figura 13: Comando `open` para ingresar a máquina virtual 1 con `psftp.exe`



```

-rw-r--r-- 1 tarea10 tarea10 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 tarea10 tarea10 3771 Apr 4 2018 .bashrc
drwx----- 2 tarea10 tarea10 4096 Jan 11 02:15 .cache
drwx----- 3 tarea10 tarea10 4096 Jan 11 02:15 .gnupg
-rw-r--r-- 1 tarea10 tarea10 307 Apr 4 2018 .profile
drwx----- 2 tarea10 tarea10 4096 Jan 11 02:06 .sch
-rw-r--r-- 1 tarea10 tarea10 0 Jan 11 02:15 .sudo_as_admin_successful
psftp> cd Tarea10
Directory /home/tarea10/Tarea10: no such file or directory
psftp> help put
put [ -R ] [ - ] [ <filename> ] [ <remote-filename> ]
Uploads a file to the server and stores it there under
the same name, or under a different one if you supply the
argument <remote-filename>.
If -R specified, recursively store a directory.
psftp> lpwd
Current local directory is C:\Users\Lenovo\Documents\ESCOM\Nivel4\Desarrollo de sistemas distribuidos\Tarea10\RECURS
psftp> put Servidor2.java
local:Servidor2.java => remote:/home/tarea10/Servidor2.java
psftp> put SimpleProxyServer.java
local:SimpleProxyServer.java => remote:/home/tarea10/SimpleProxyServer.java
psftp>

```

Figura 14: Comando `put` para pasar archivos a máquina virtual 1 con `psftp.exe`

Ahora ya tenemos los archivos en la máquina virtual 1. (Ver figura 15)



```

tarea10@Tarea10-1:~$ ls
Servidor2.java SimpleProxyServer.java
tarea10@Tarea10-1:~$

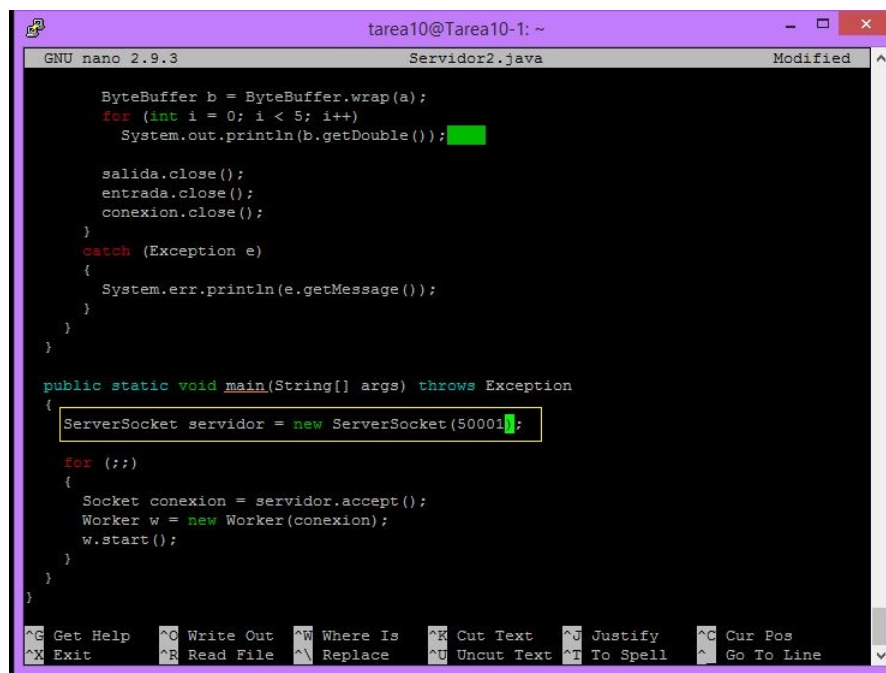
```

Figura 15: Comando `ls` para ver archivos en máquina virtual 1

## 2.6. Editar main de archivo `Servidor2.java` en máquina virtual 1

En la máquina virtual 1 editar el método *main* en el archivo `Servidor2.java`:

```
ServerSocket servidor = new ServerSocket(50001);
```



```
GNU nano 2.9.3 Servidor2.java Modified
ByteBuffer b = ByteBuffer.wrap(a);
for (int i = 0; i < 5; i++)
    System.out.println(b.getDouble());

salida.close();
entrada.close();
conexion.close();
}
catch (Exception e)
{
    System.err.println(e.getMessage());
}
}

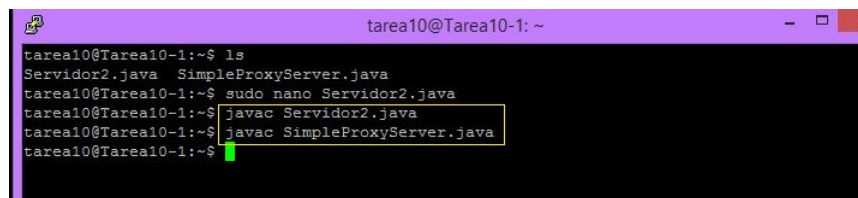
public static void main(String[] args) throws Exception
{
    ServerSocket servidor = new ServerSocket(50001);

    for (;;)
    {
        Socket conexion = servidor.accept();
        Worker w = new Worker(conexion);
        w.start();
    }
}
```

Figura 16: Cambio en main de programa en máquina virtual 1

## 2.7. Compilar programas en máquina 1

Compilar en la máquina virtual 1 los programas Servidor2.java y SimpleProxyServer.java. Los comandos se muestran en la figura



```
tarea10@Tarea10-1: ~
tarea10@Tarea10-1:~$ ls
Servidor2.java SimpleProxyServer.java
tarea10@Tarea10-1:~$ sudo nano Servidor2.java
tarea10@Tarea10-1:~$ javac Servidor2.java
tarea10@Tarea10-1:~$ javac SimpleProxyServer.java
tarea10@Tarea10-1:~$
```

Figura 17: Compilar archivos en máquina virtual 1

## 2.8. Conectar máquina virtual 2

Esta máquina virtual es la replica de la pasada. Primero abriremos la máquina con **putty.exe**, igual que con la máquina pasada. (Ve figura 18)

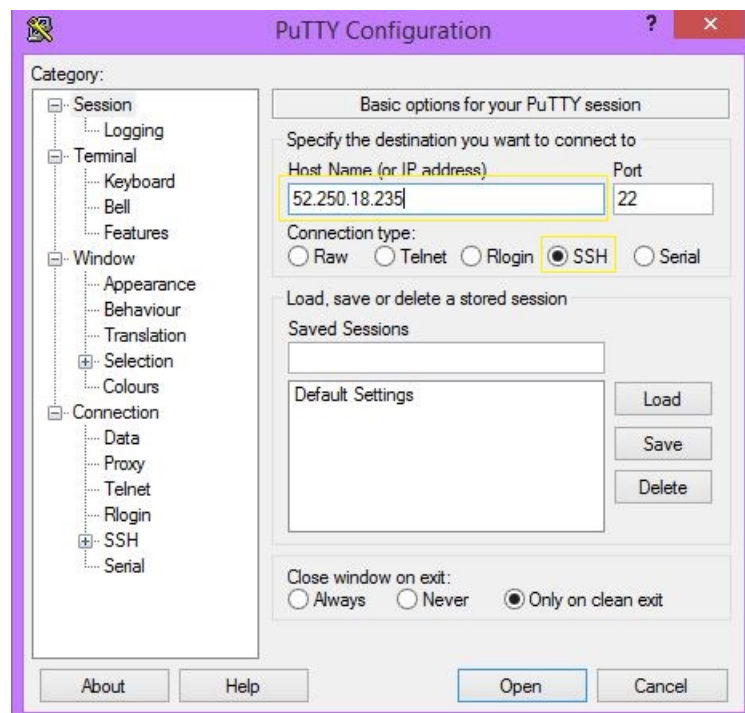


Figura 18: Putty para abrir la máquina virtual 2

## 2.9. Instalar jdk en la máquina virtual 2

Para este paso usaremos los mismos comandos que en la máquina pasada. (Ver figuras 19 y 20)

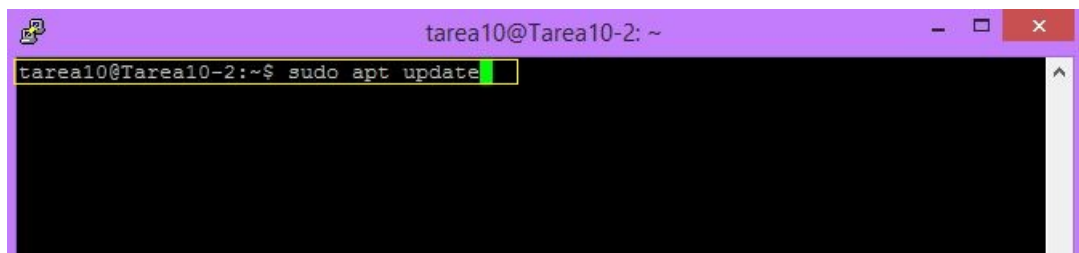
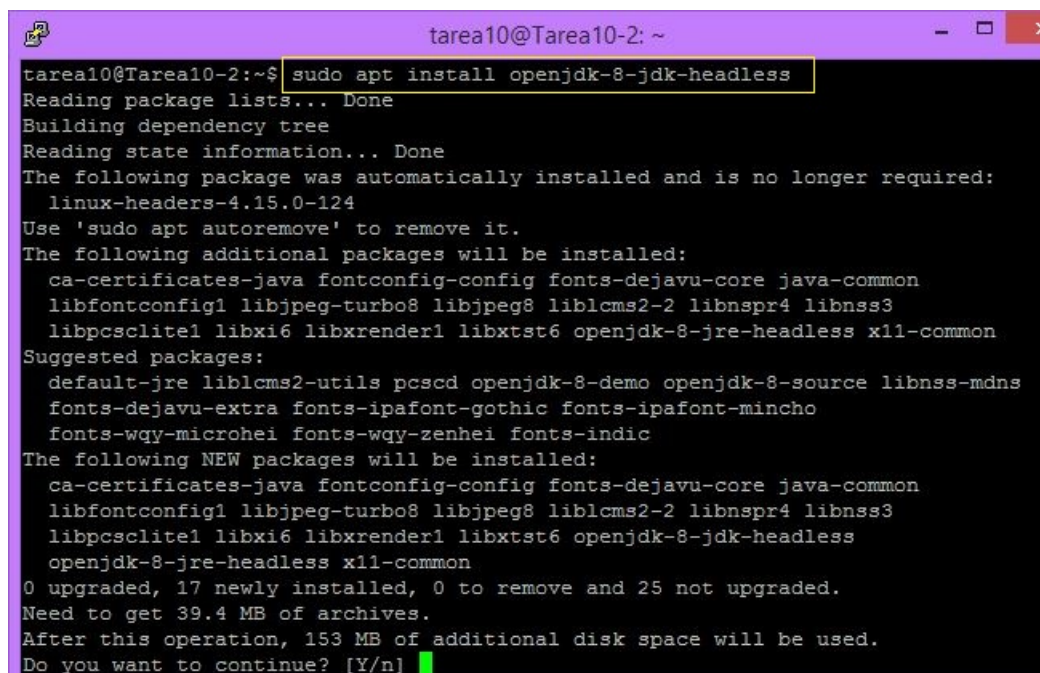


Figura 19: Comando update en la máquina virtual 2

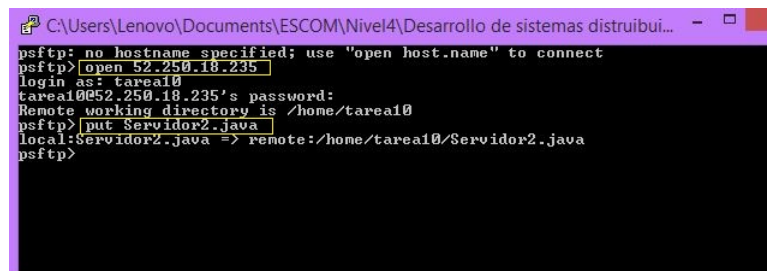


```
tarea10@Tarea10-2: ~  
tarea10@Tarea10-2:~$ sudo apt install openjdk-8-jdk-headless  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  linux-headers-4.15.0-124  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common  
  libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libnspr4 libnss3  
  libpcsclite1 libxi6 libxrender1 libxtst6 openjdk-8-jre-headless x11-common  
Suggested packages:  
  default-jre liblcms2-utils pscd openjdk-8-demo openjdk-8-source libnss-mdns  
  fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho  
  fonts-wqy-microhei fonts-wqy-zenhei fonts-indic  
The following NEW packages will be installed:  
  ca-certificates-java fontconfig-config fonts-dejavu-core java-common  
  libfontconfig1 libjpeg-turbo8 libjpeg8 liblcms2-2 libnspr4 libnss3  
  libpcsclite1 libxi6 libxrender1 libxtst6 openjdk-8-jdk-headless  
  openjdk-8-jre-headless x11-common  
0 upgraded, 17 newly installed, 0 to remove and 25 not upgraded.  
Need to get 39.4 MB of archives.  
After this operation, 153 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Figura 20: Comando de instalación de JDK en la máquina virtual 2

## 2.10. Enviar a la máquina virtual 2 archivos

Utilizando el programa **psftp.exe** enviar a la máquina virtual 2 el archivo **Servidor2.java**. El procedimiento es el mismo, solo que tenemos que usar la IP de esta máquina, como se muestra en la figura 21.



```
C:\Users\Lenovo\Documents\ESCOM\Nivel4\Desarrollo de sistemas distribu...  
psftp: no hostname specified; use "open host.name" to connect  
psftp> open 52.250.18.235  
login as: tarea10  
tarea10@52.250.18.235's password:  
Remote working directory is /home/tarea10  
psftp> put Servidor2.java  
local:Servidor2.java => remote:/home/tarea10/Servidor2.java  
psftp>
```

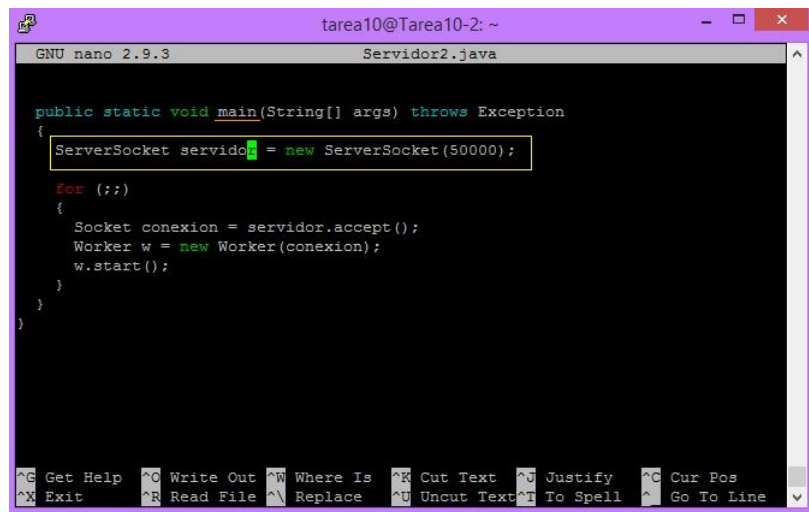
Figura 21: Comando open y put para ingresar a máquina virtual 2 y mandar archivos con psftp.exe

## 2.11. Editar main de archivo Servidor2.java en máquina virtual 2

En la máquina virtual 1 editar el método *main* en el archivo **Servidor2.java**:

```
ServerSocket servidor = new ServerSocket(50000);
```





```
GNU nano 2.9.3 Servidor2.java

public static void main(String[] args) throws Exception
{
    ServerSocket servidor = new ServerSocket(50000);

    for (;;)
    {
        Socket conexion = servidor.accept();
        Worker w = new Worker(conexion);
        w.start();
    }
}
```

Figura 22: Cambio en main de programa en máquina virtual 2

Este paso se puede saltar, ya que el código ya está como se pide.

## 2.12. Compilar programa en máquina 2

Compilar el programa Servidor2.java, esto se muestra en la figura 23.



```
tarea10@Tarea10-2: ~$ ls
Servidor2.java
tarea10@Tarea10-2: ~$ sudo nano Servidor2.java
tarea10@Tarea10-2: ~$ javac Servidor2.java
tarea10@Tarea10-2: ~$
```

Figura 23: Compilar archivo en máquina virtual 2

## 2.13. Ejecutar el programa en máquina virtual 2

Ejecutar el programa Servidor2.java en la máquina virtual 2, para esto usamos el siguiente comando que se muestra en la figura 24.



```
tarea10@Tarea10-2: ~$ java Servidor2&
[1] 16081
tarea10@Tarea10-2: ~$
```

Figura 24: Ejecutar archivo en máquina virtual 2

## 2.14. Ejecutar los programas en máquina virtual 1

Para ejecutar el programa Servidor2.java en la máquina virtual 1 usamos el comando que se muestra en la figura 25, es igual que con la otra máquina.



```
tarea10@Tarea10-1: ~  
tarea10@Tarea10-1:~$ ls  
Servidor2.java SimpleProxyServer.java  
tarea10@Tarea10-1:~$ sudo nano Servidor2.java  
tarea10@Tarea10-1:~$ javac Servidor2.java  
tarea10@Tarea10-1:~$ javac SimpleProxyServer.java  
tarea10@Tarea10-1:~$ java Servidor2 &  
[1] 4857  
tarea10@Tarea10-1:~$ java SimpleProxyServer 52.250.18.235 50000 50000 50001 &  
[2] 4899  
tarea10@Tarea10-1:~$ Starting proxy for 52.250.18.235:50000 on port 50000
```

Figura 25: Ejecutar archivos en máquina virtual 1

Después ejecutar el máquina virtual 1 el proxy, de la manera que se muestra en la figura 25. De tal manera que pasamos la IP de la réplica, el puerto abierto en la réplica, el puerto abierto en el sistema principal y el puerto en la máquina virtual 1 donde el programa Servidor2.java recibe las peticiones.

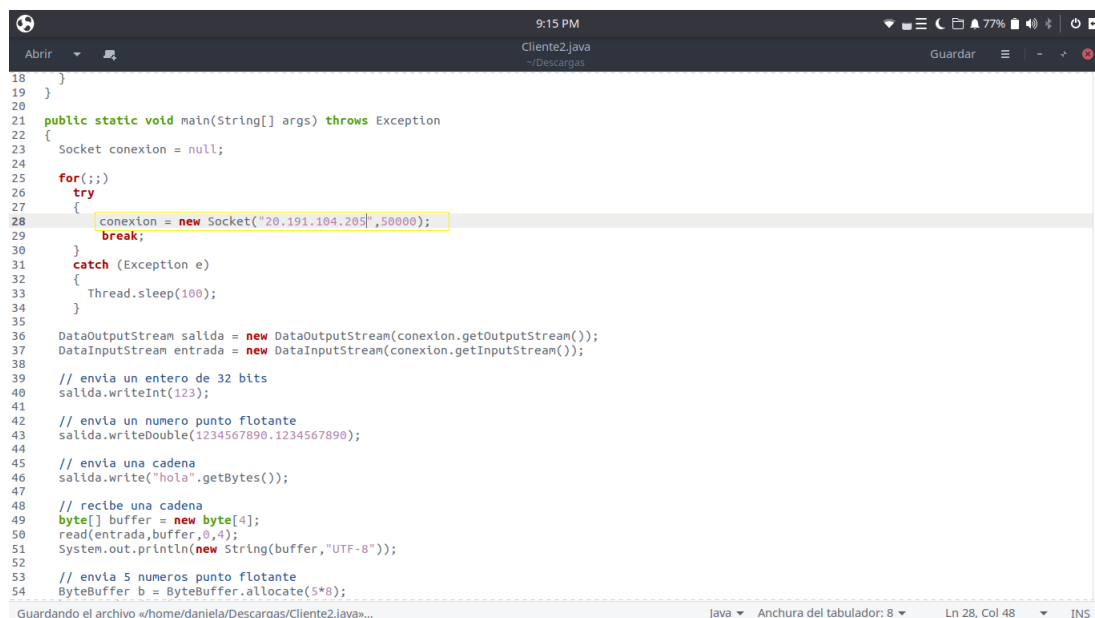
Nota: El puerto 50001 no se debe abrir en la máquina virtual 1, ya que el proxy y Servidor2.java se comunican mediante loopback.

## 2.15. En Ubuntu

Esta sección también se puede realizar en Windows, y es exactamente los mismo.

### 2.15.1. Editar el programa cliente

Editar el programa Cliente2.java para que se conecte a la máquina virtual 1. Basta con cambiar localhost por la IP de la máquina virtual 1. (Ver figura 26)



```
18 }  
19 }  
20 }  
21 public static void main(String[] args) throws Exception  
22 {  
23     Socket conexion = null;  
24  
25     for(;;)  
26     {  
27         try  
28         {  
29             conexion = new Socket("20.191.104.205", 50000);  
30             break;  
31         }  
32         catch (Exception e)  
33         {  
34             Thread.sleep(100);  
35         }  
36  
37         DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());  
38         DataInputStream entrada = new DataInputStream(conexion.getInputStream());  
39  
40         // envia un entero de 32 bits  
41         salida.writeInt(123);  
42  
43         // envia un numero punto flotante  
44         salida.writeDouble(1234567890.1234567890);  
45  
46         // envia una cadena  
47         salida.write("hola".getBytes());  
48  
49         // recibe una cadena  
50         byte[] buffer = new byte[4];  
51         read(entrada, buffer, 0, 4);  
52         System.out.println(new String(buffer, "UTF-8"));  
53  
54         // envia 5 numeros punto flotante  
55         ByteBuffer b = ByteBuffer.allocate(5*8);
```

Figura 26: Editar programa Cliente2.java



### 2.15.2. Compilar programa

Compilar el programa Cliente2.java, el comando se muestra en la figura 27

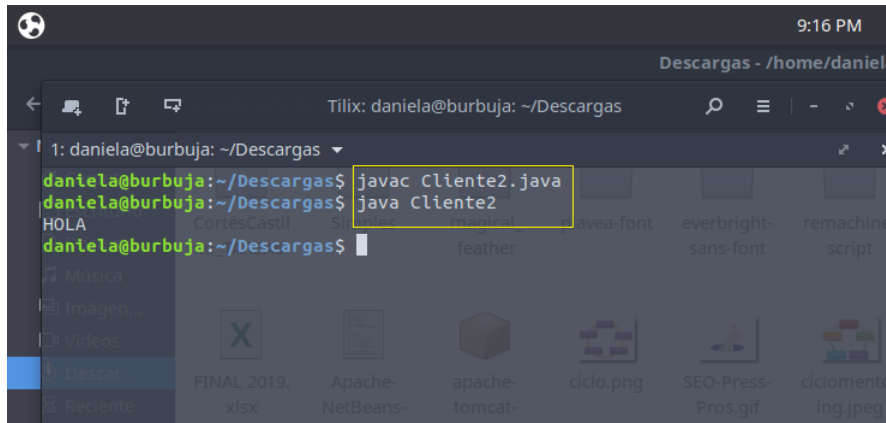


Figura 27: Correr y compilar programa Cliente2.java

### 2.15.3. Ejecutar el programa

Ejecutar el programa Cliente2.java, el comando se muestra en la figura 27. Después de ejecutarlo, el resultado que obtenemos en las máquinas virtuales se puede observar en la figura 28.

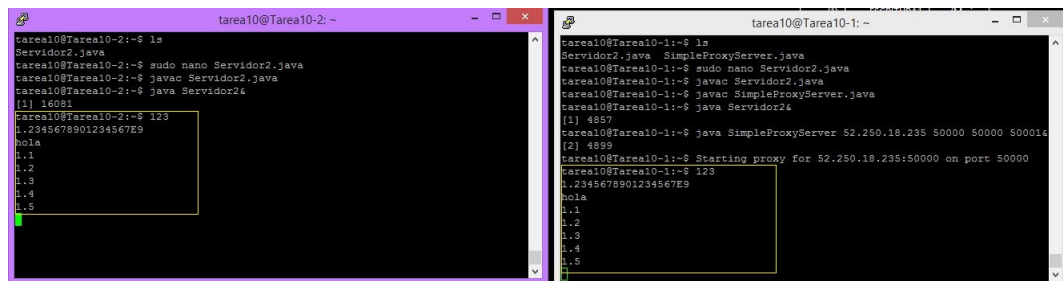


Figura 28: Resultado en máquinas virtuales tras ejecutar Cliente2.java