



Instituto Politécnico Nacional Escuela Superior de Cómputo

Tarea 10. Replicación de un servidor en la nube

PRESENTA

Vladimir Azpeitia Hernández

PROFESOR

Carlos Pineda Guerrero

ASGINATURA

Desarrollo de Sistemas Distribuidos - 4CV2

25 de mayo de 2021

1. Introducción

1.1. Planteamiento del problema

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.

Como vimos en clase, 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.

Vamos a utilizar el programa SimpleProxyServer.java el cual es un proxy simple escrito en Java, que he modificado para que funcione como un administrador de tráfico.

2. Desarrollo

2.1. Creación de la primera máquina

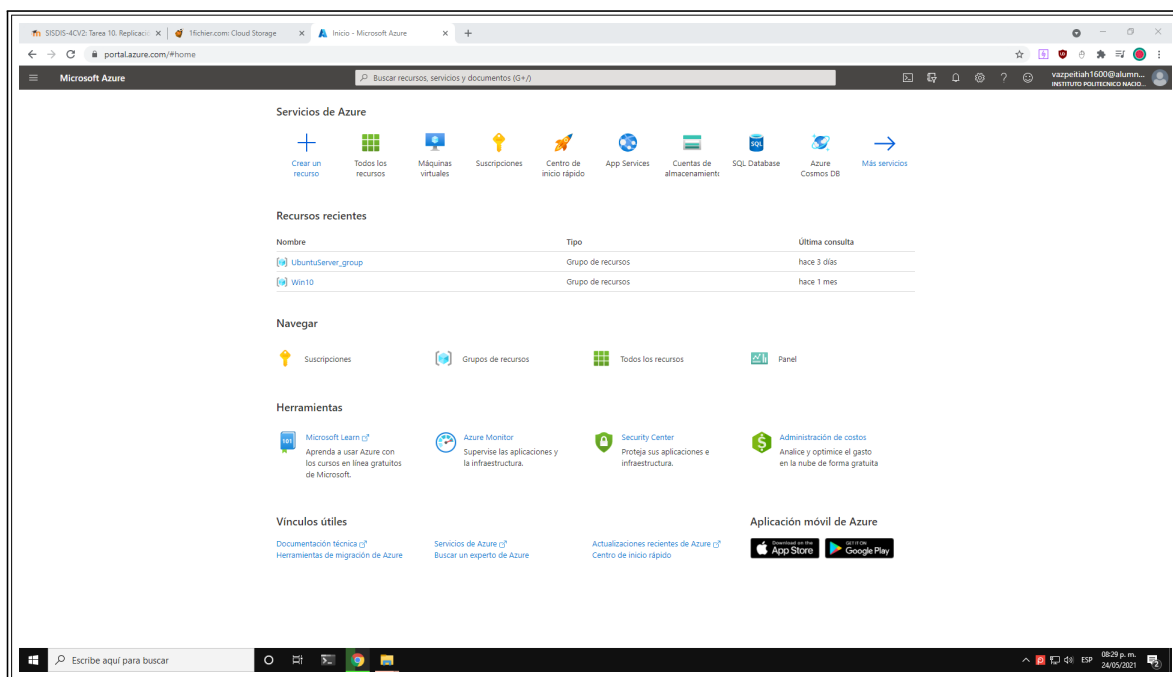


Figura 1: Creación de la máquina 1: Portal de azure

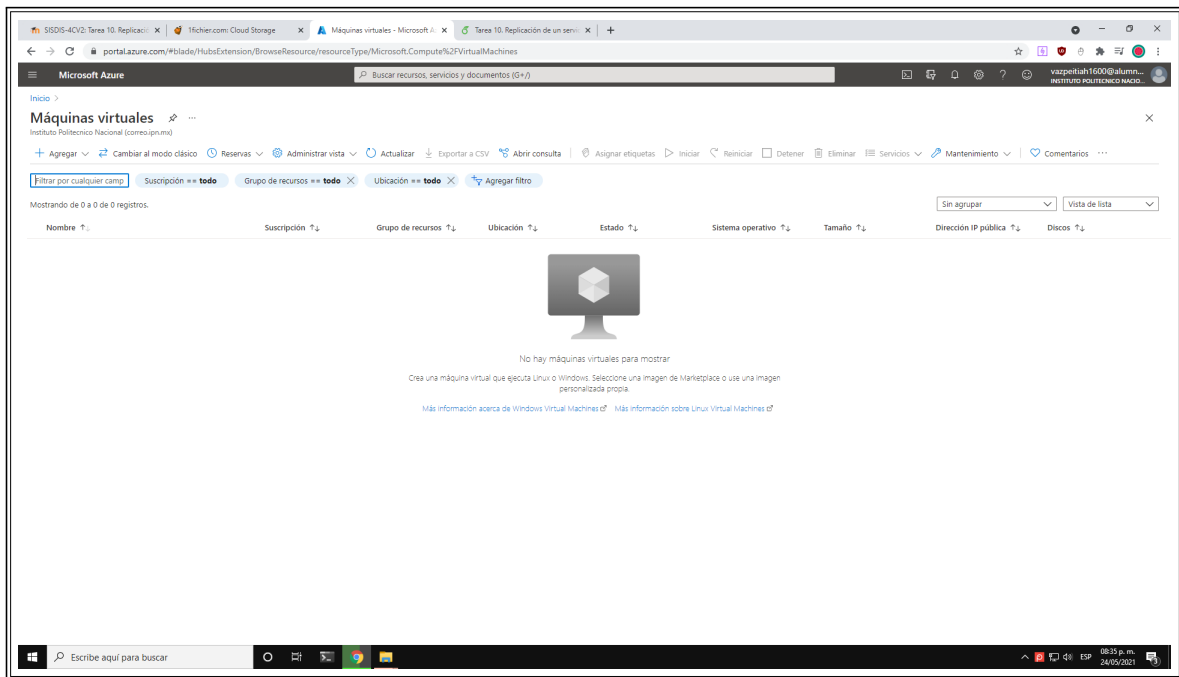


Figura 2: Creación de la máquina 1: Máquinas virtuales

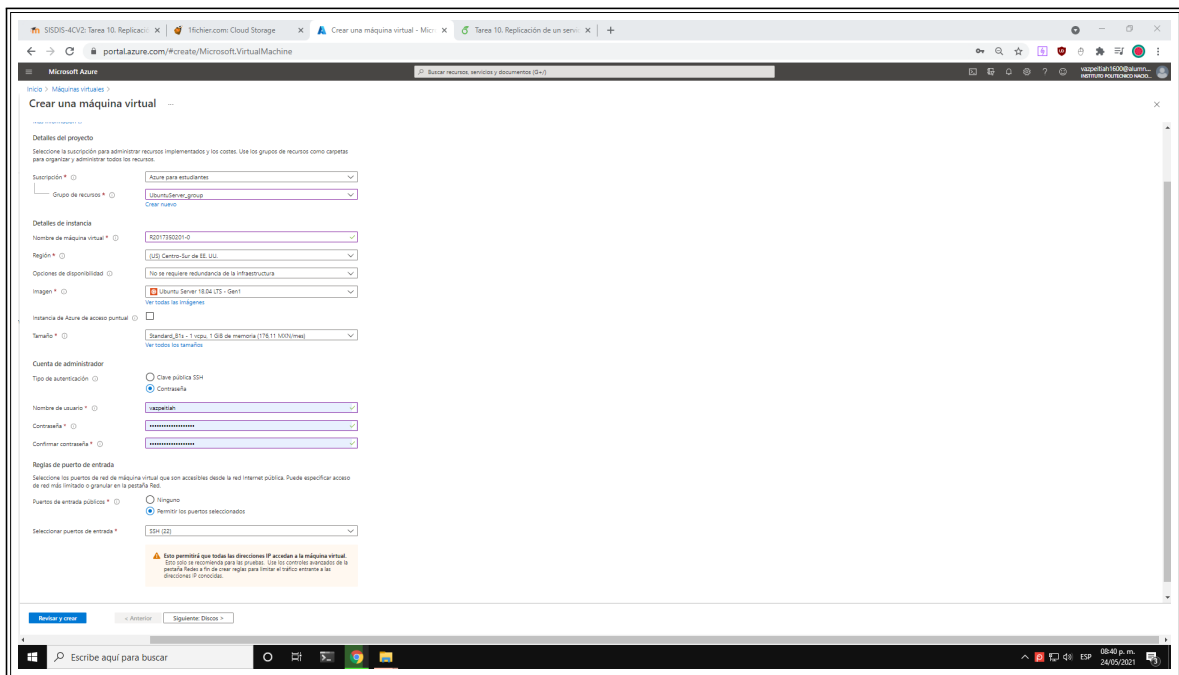


Figura 3: Creación de la máquina 1: Datos básicos

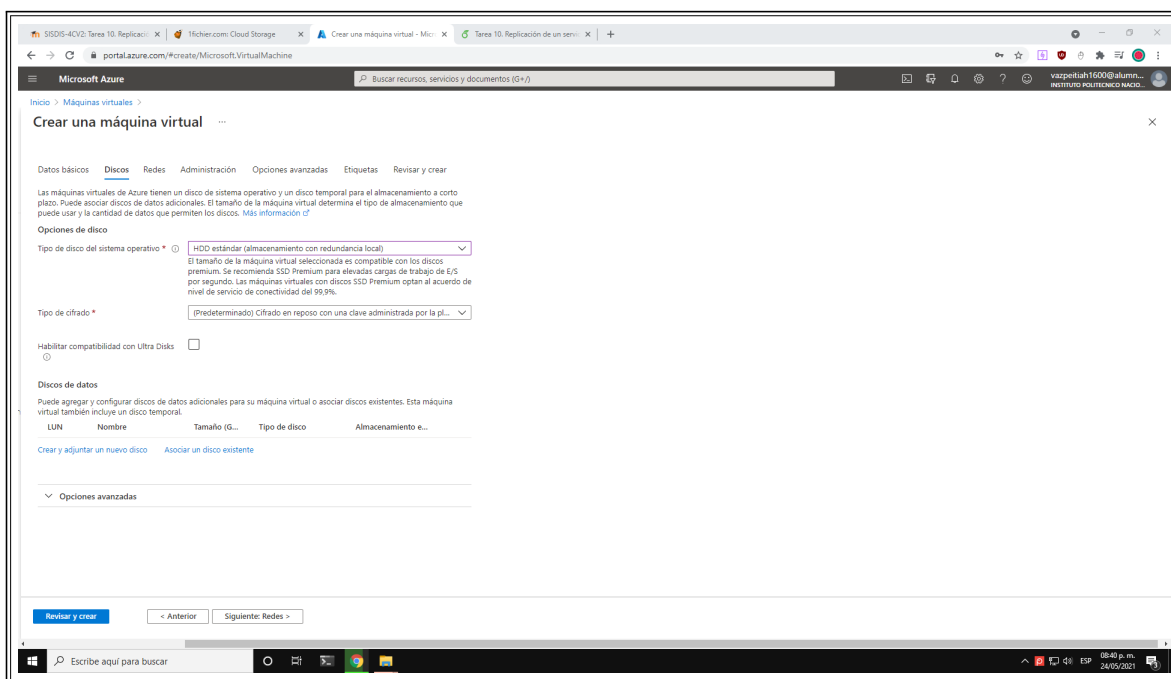


Figura 4: Creación de la máquina 1: Discos

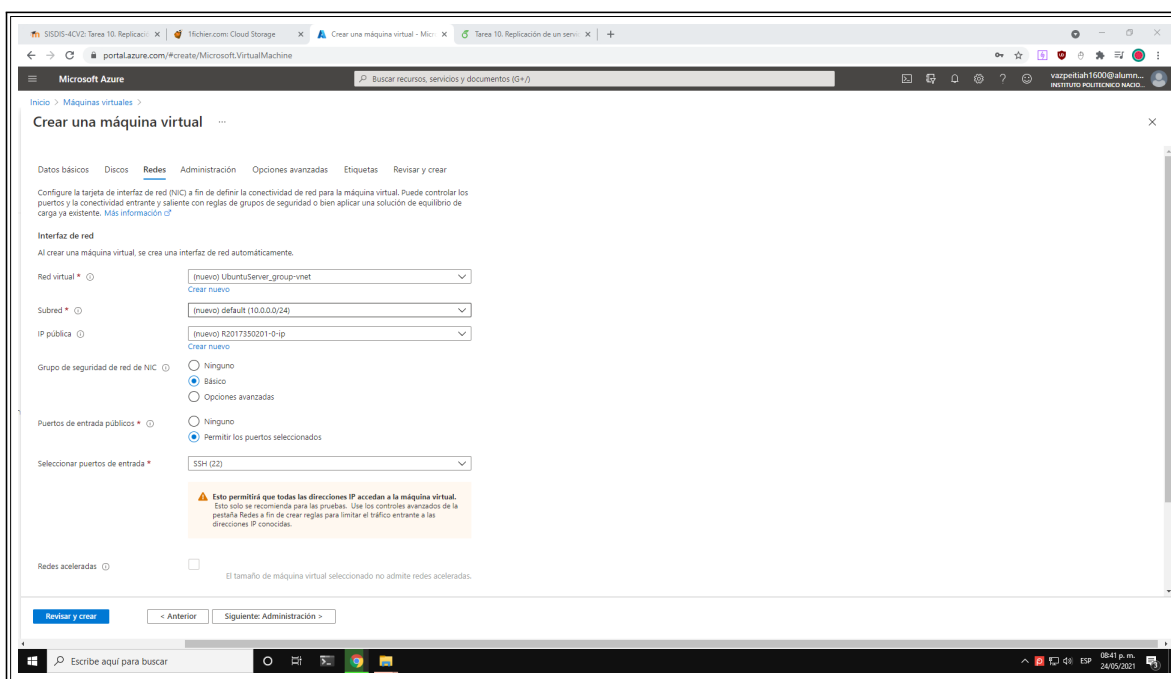


Figura 5: Creación de la máquina 1: Redes

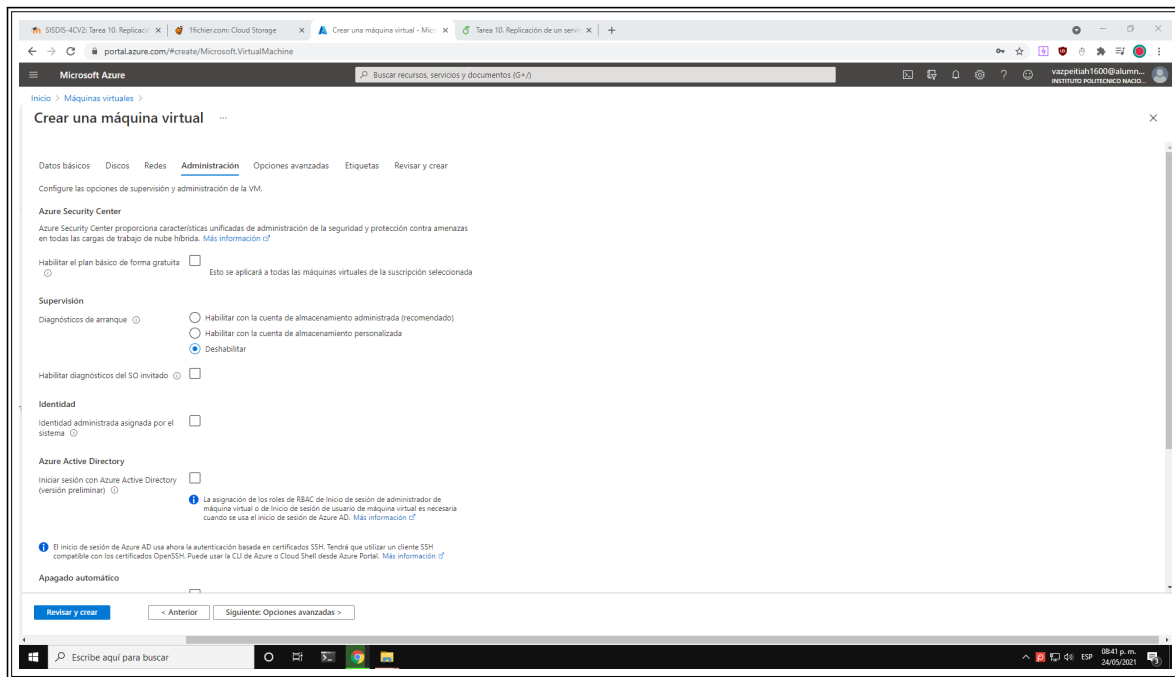


Figura 6: Creación de la máquina 1: Administración

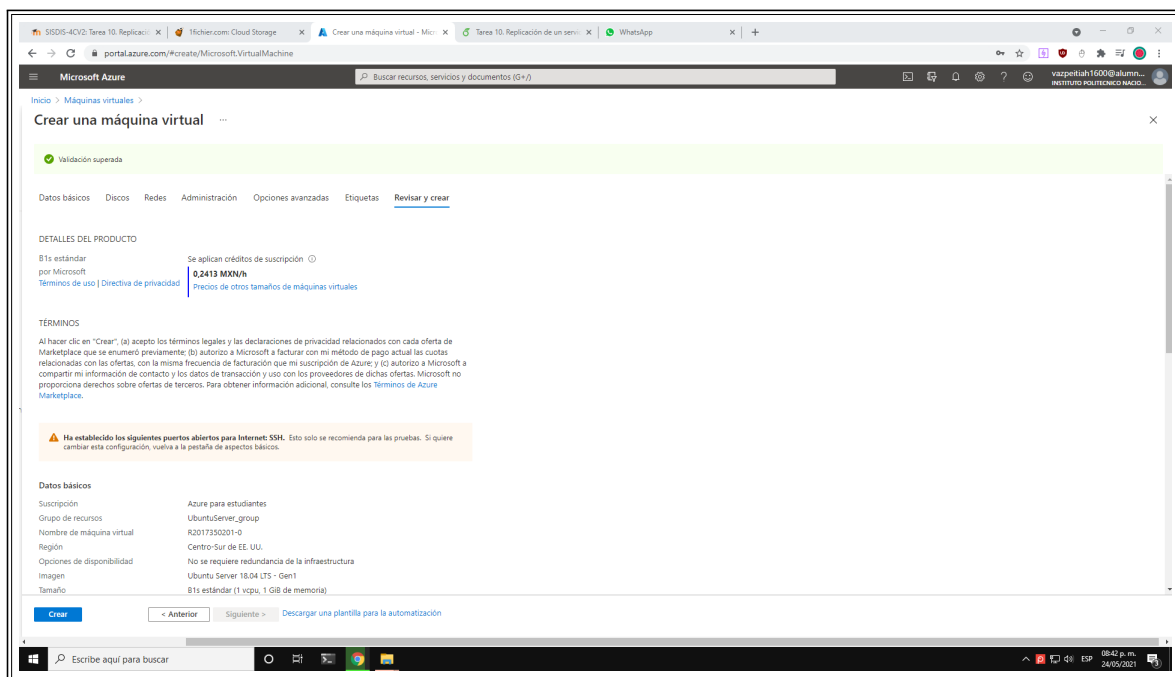


Figura 7: Creación de la máquina 1: Validación superada

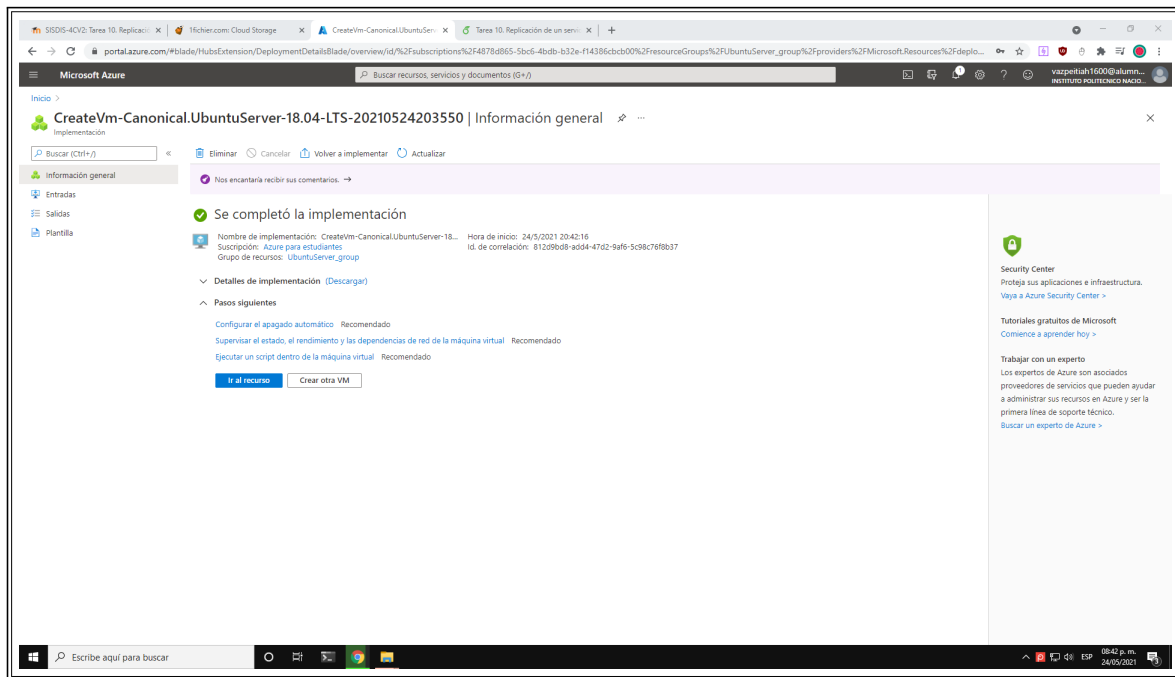


Figura 8: Creación de la máquina 1: Implementación completada

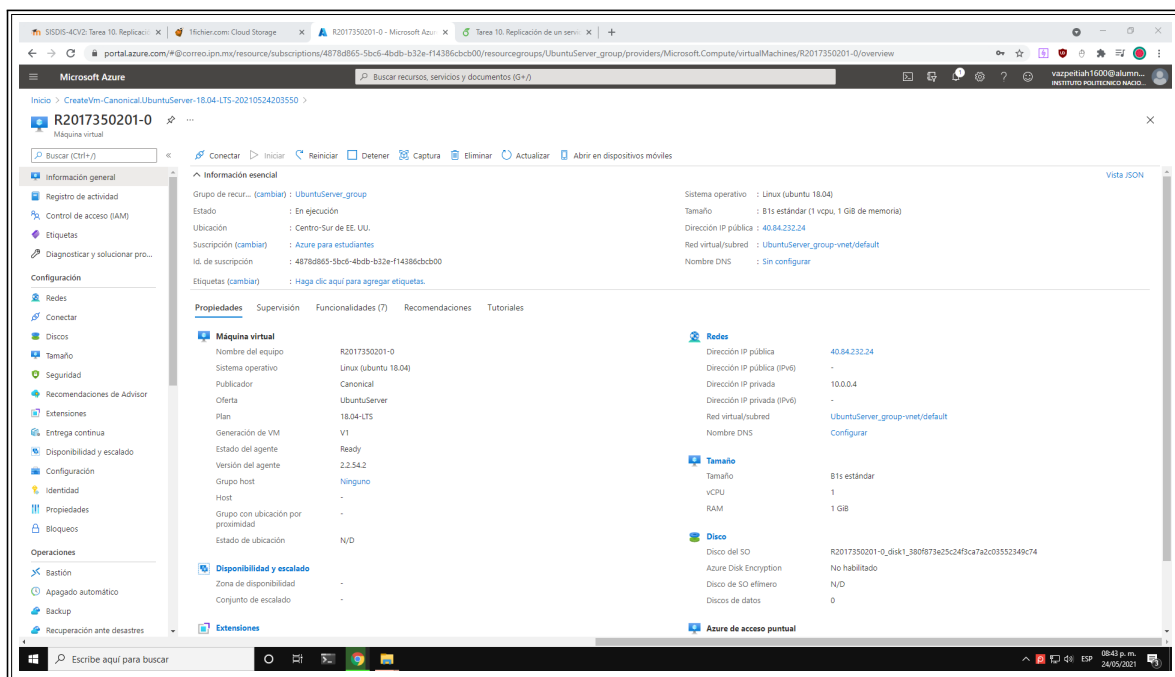


Figura 9: Creación de la máquina 1: Panel de control de la máquina virtual

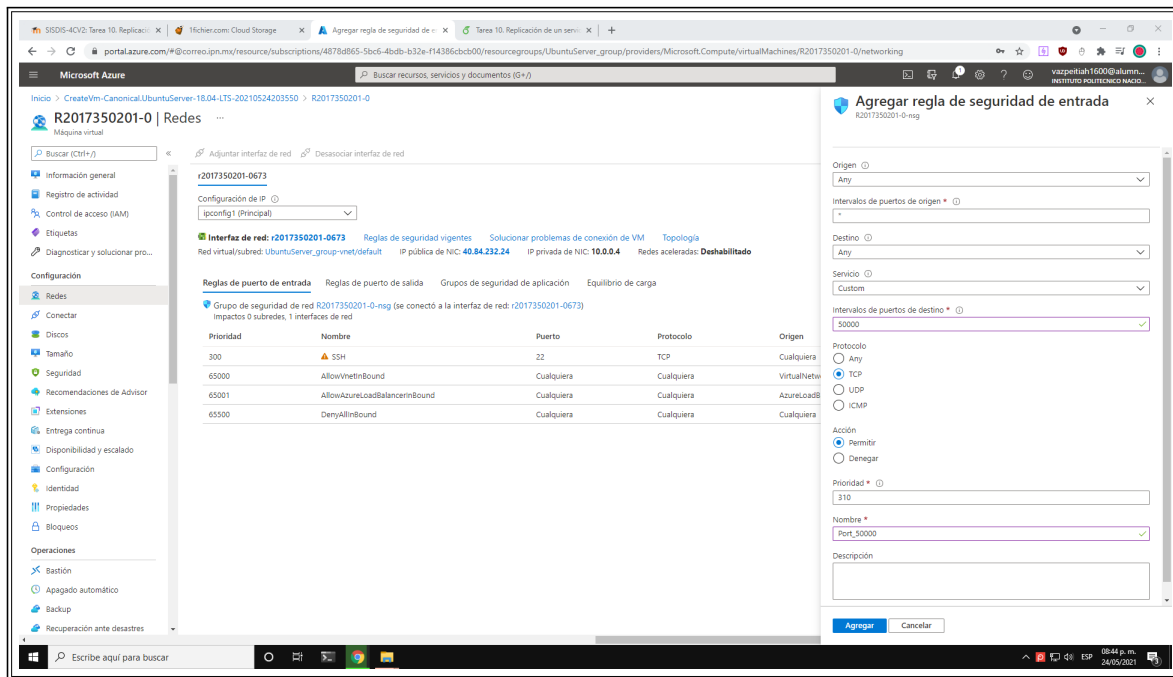


Figura 10: Creación de la máquina 1: Configurando puerto 50000

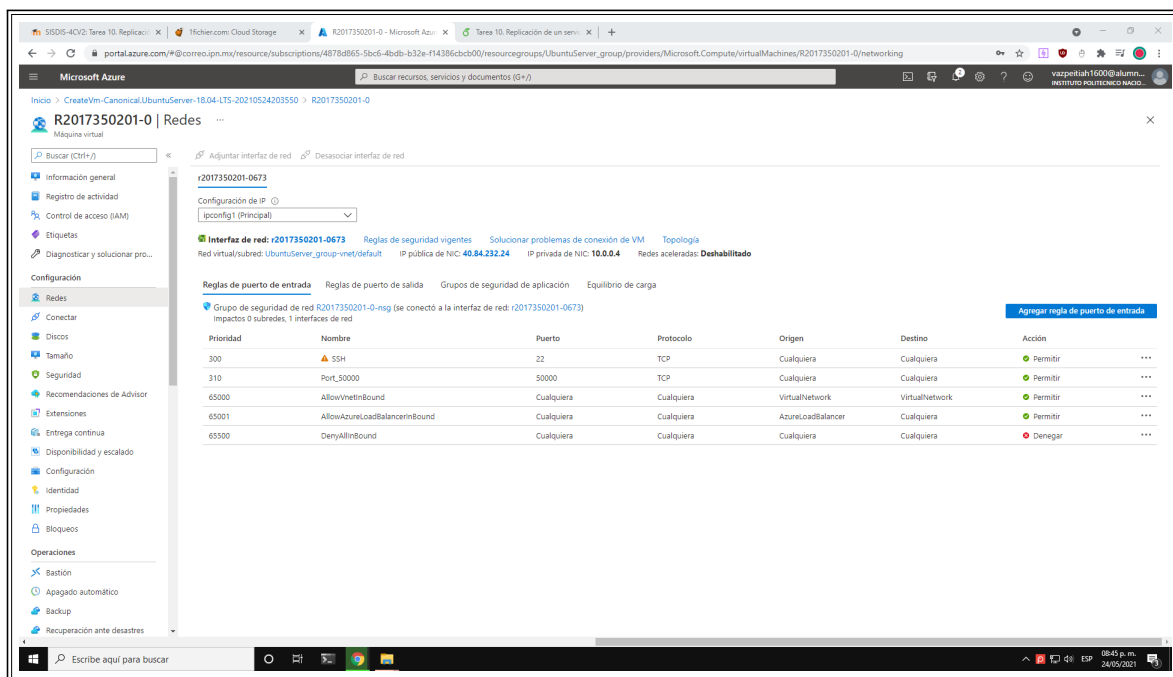


Figura 11: Creación de la máquina 1: Puerto 50000 abierto para TCP

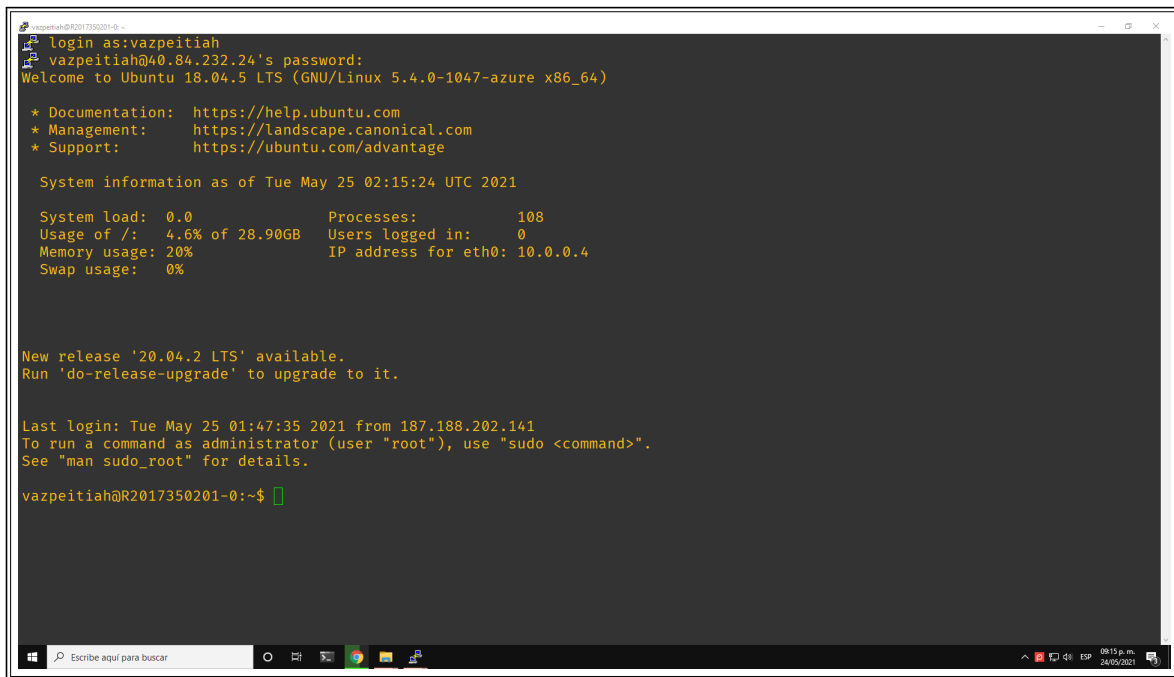


Figura 12: Creación de la máquina 1: Conexión a través de ssh a la máquina virtual, usando PuTTY

2.2. Creación de la segunda máquina

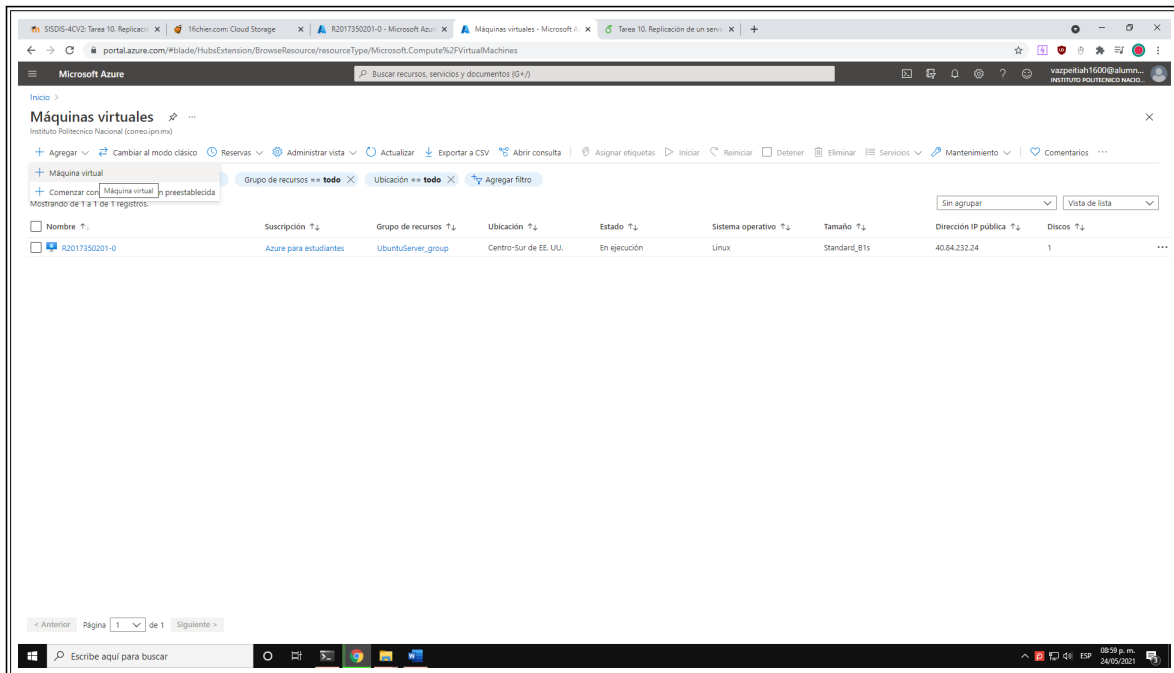


Figura 13: Creación de la máquina 2: Máquinas virtuales

Figura 14: Creación de la máquina 2: Datos básicos

Figura 15: Creación de la máquina 2: Discos

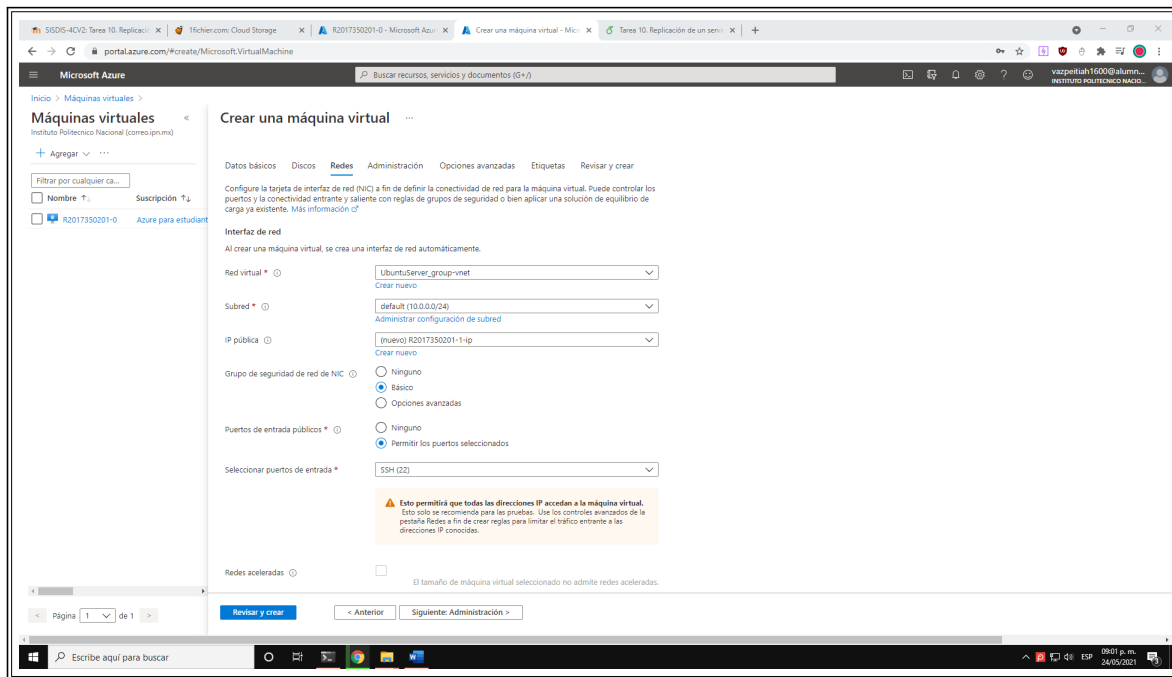


Figura 16: Creación de la máquina 2: Redes

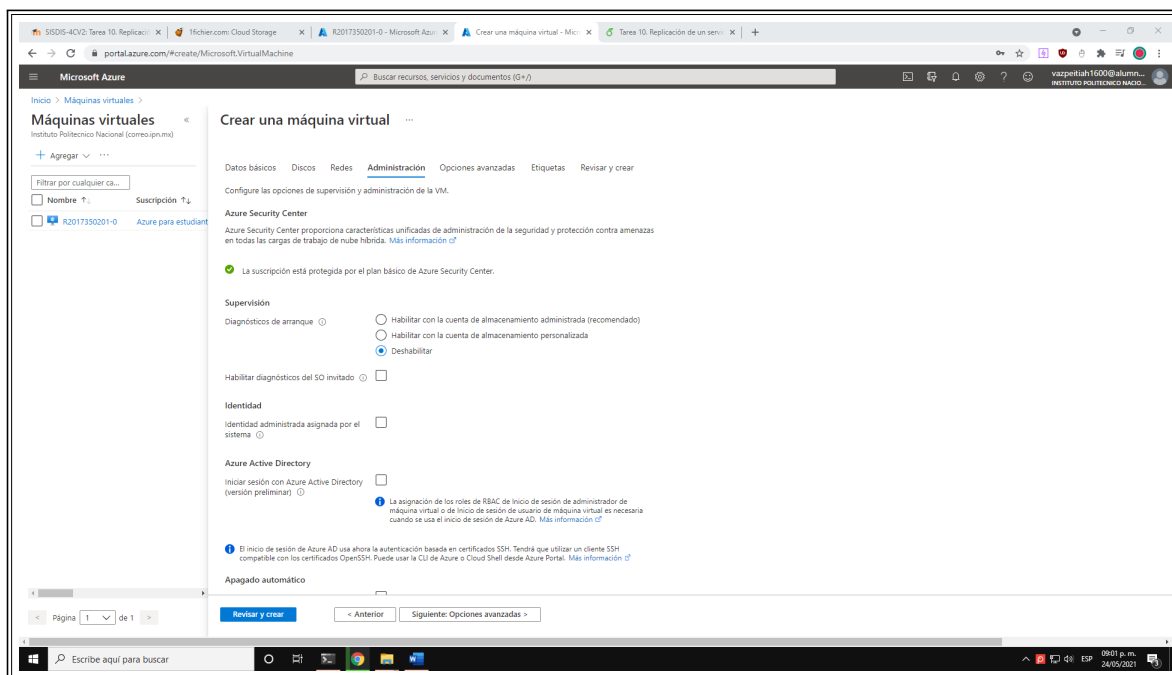


Figura 17: Creación de la máquina 2: Administración

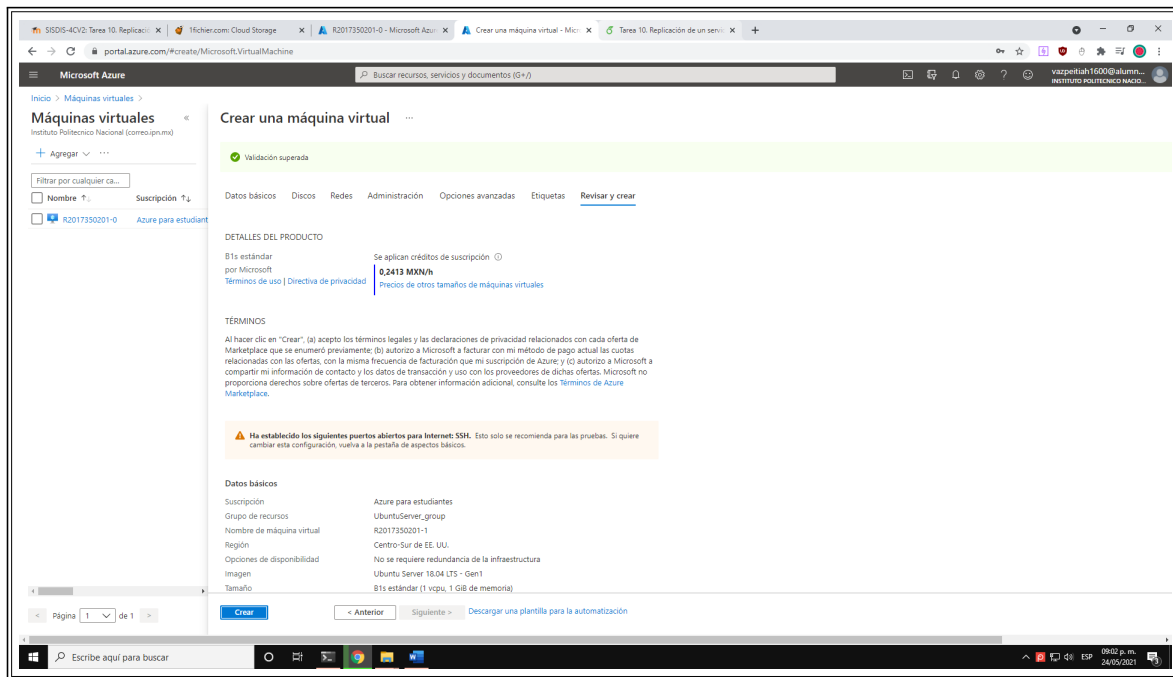


Figura 18: Creación de la máquina 2: Validación superada

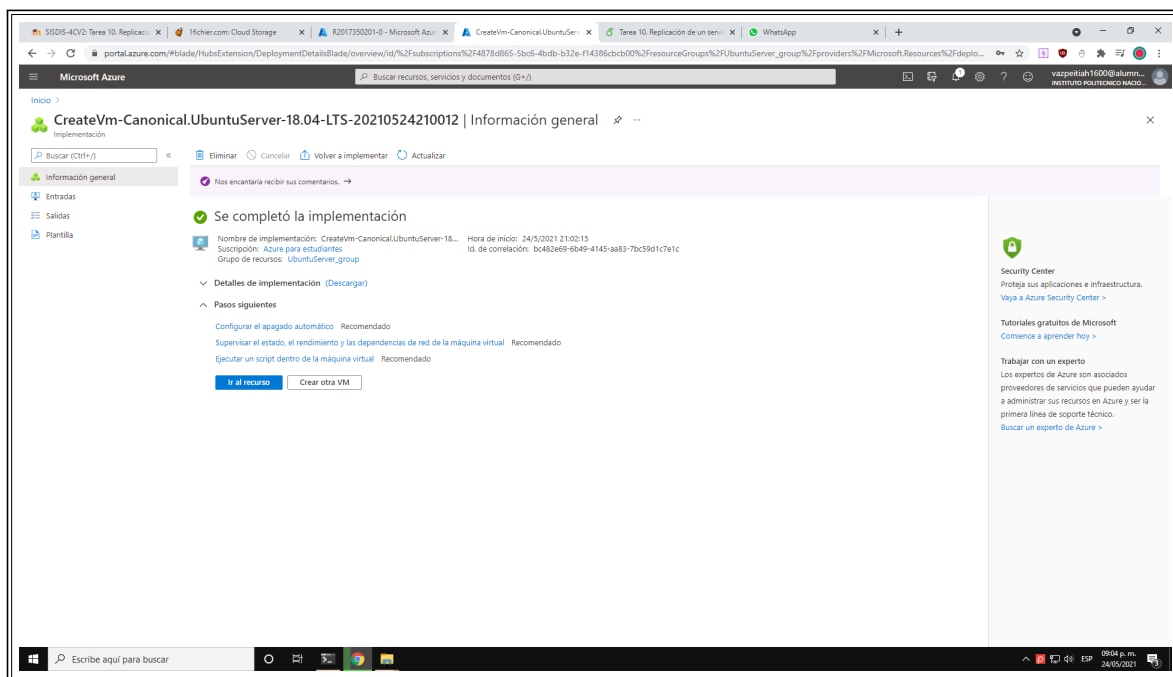


Figura 19: Creación de la máquina 2: Implementación completada

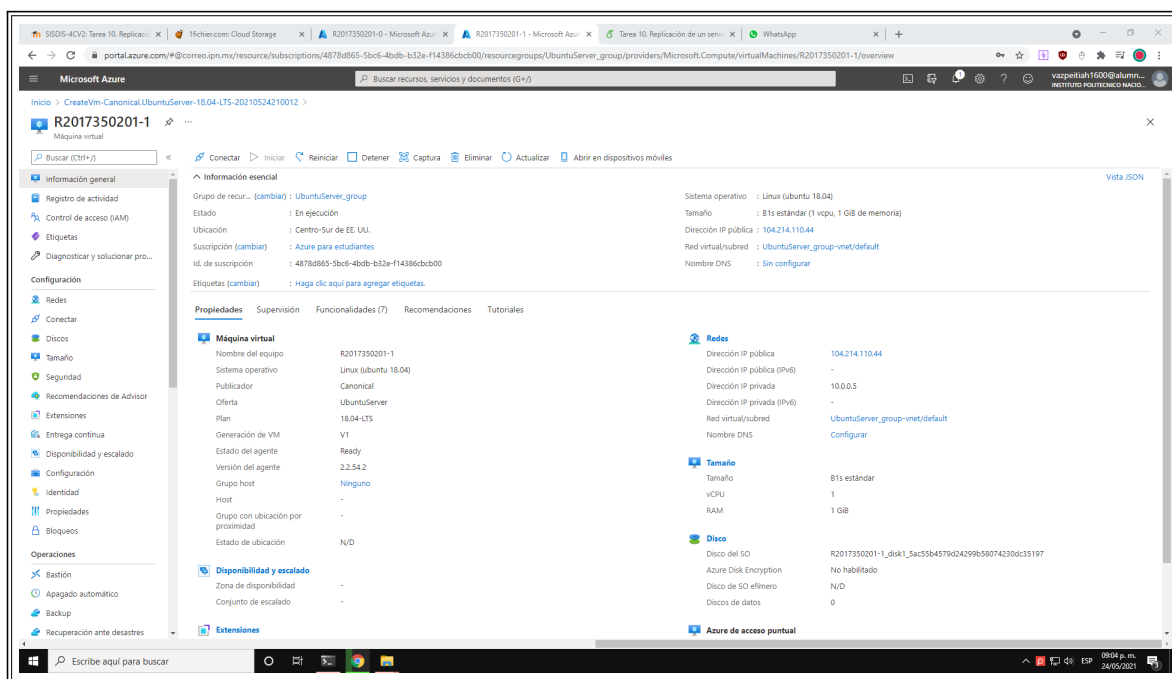


Figura 20: Creación de la máquina 2: Panel de control de la máquina virtual

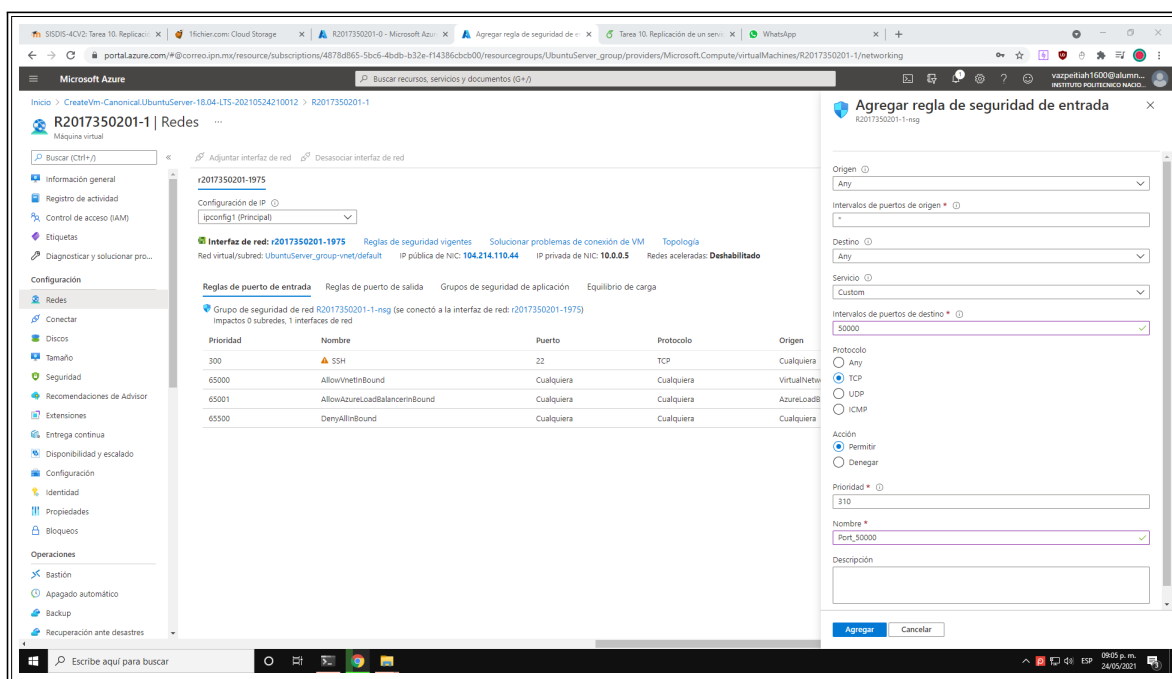


Figura 21: Creación de la máquina 2: Configurando puerto 50000

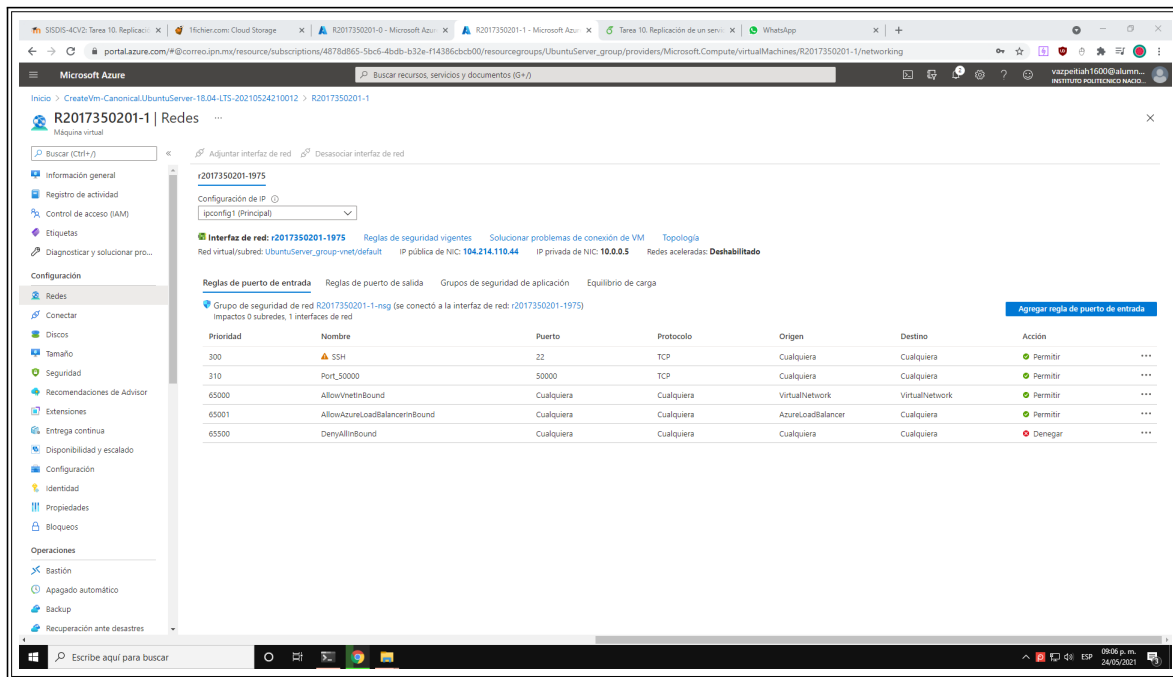


Figura 22: Creación de la máquina 2: Puerto 50000 abierto para TCP

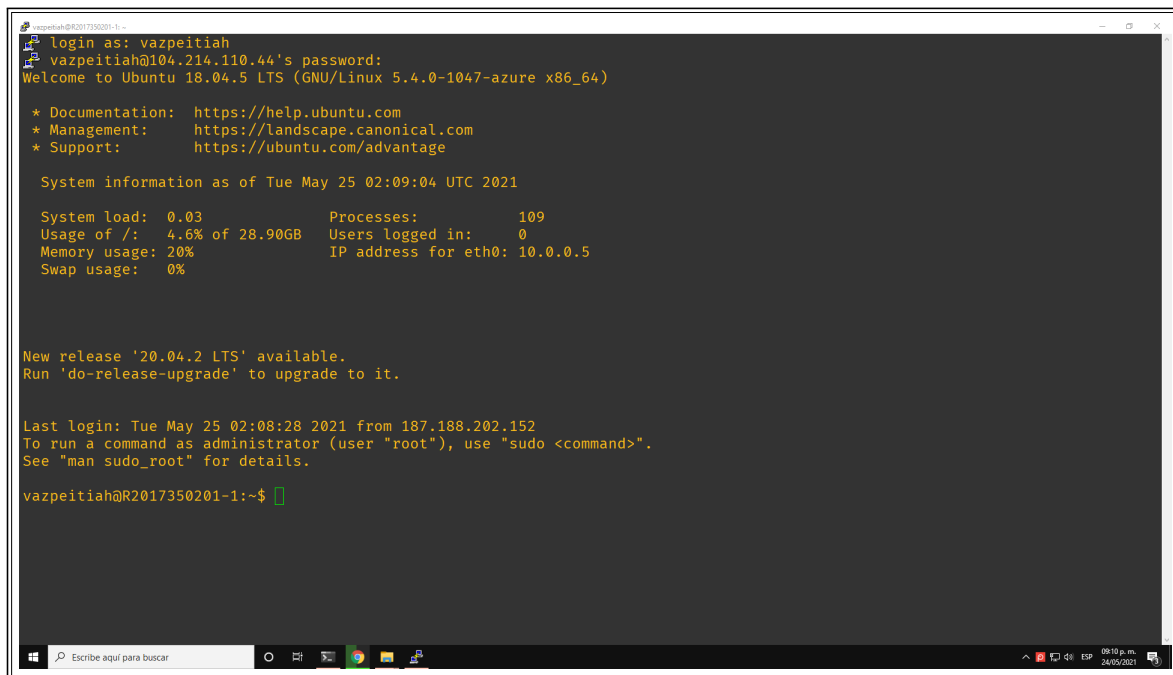
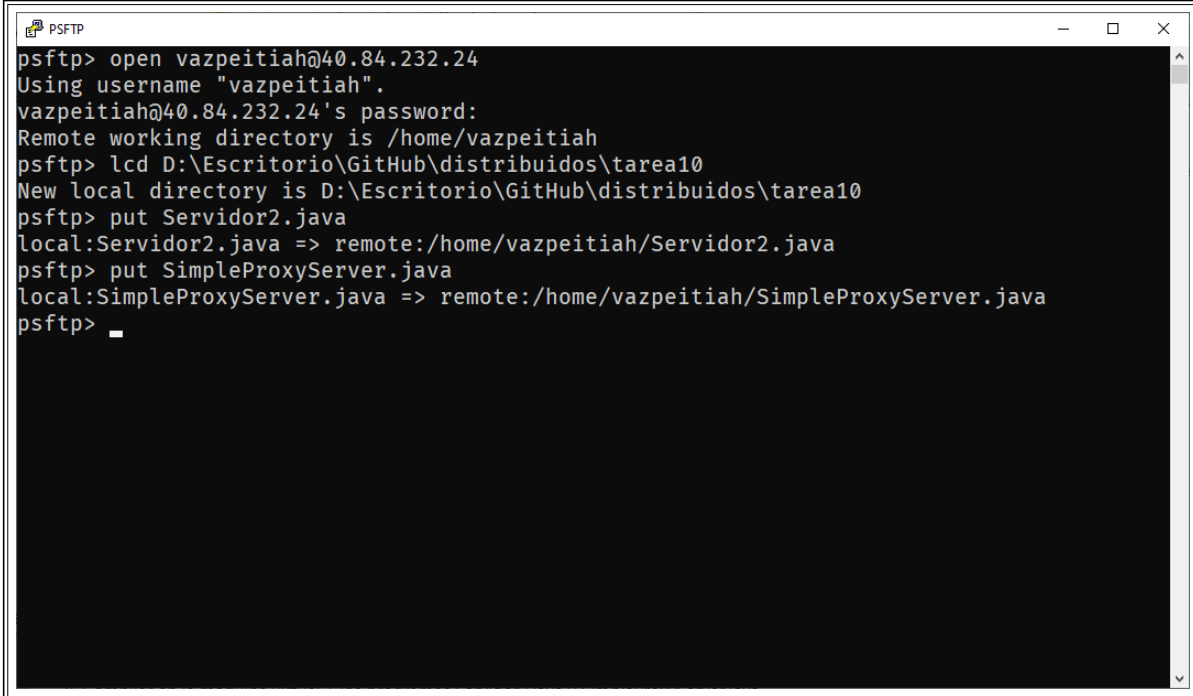


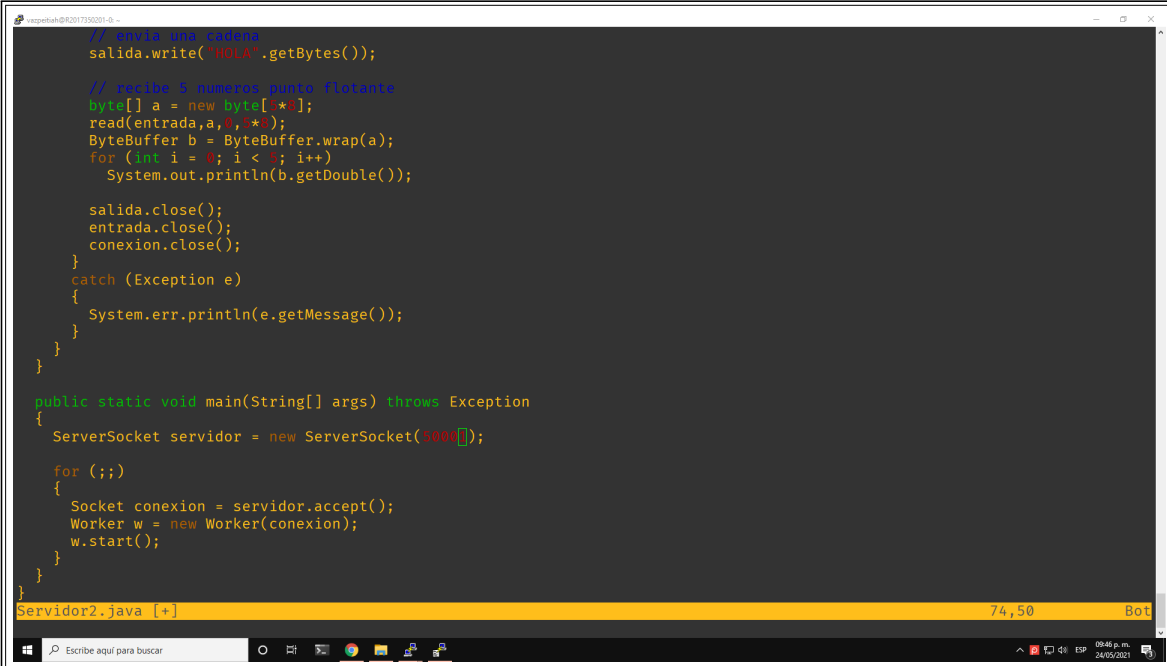
Figura 23: Creación de la máquina 2: Conexión a través de ssh a la máquina virtual, usando PuTTY

3. Realizando las actividades



```
psftp> open vazpeitiah@40.84.232.24
Using username "vazpeitiah".
vazpeitiah@40.84.232.24's password:
Remote working directory is /home/vazpeitiah
psftp> lcd D:\Escritorio\GitHub\distribuidos\tarea10
New local directory is D:\Escritorio\GitHub\distribuidos\tarea10
psftp> put Servidor2.java
local:Servidor2.java => remote:/home/vazpeitiah/Servidor2.java
psftp> put SimpleProxyServer.java
local:SimpleProxyServer.java => remote:/home/vazpeitiah/SimpleProxyServer.java
psftp> _
```

Figura 24: Utilizando el programa psftp.exe para enviar a la máquina virtual 1 los archivos: Servidor2.java y SimpleProxyServer.java



```
// recibe una cadena
salida.write("HOLA".getBytes());

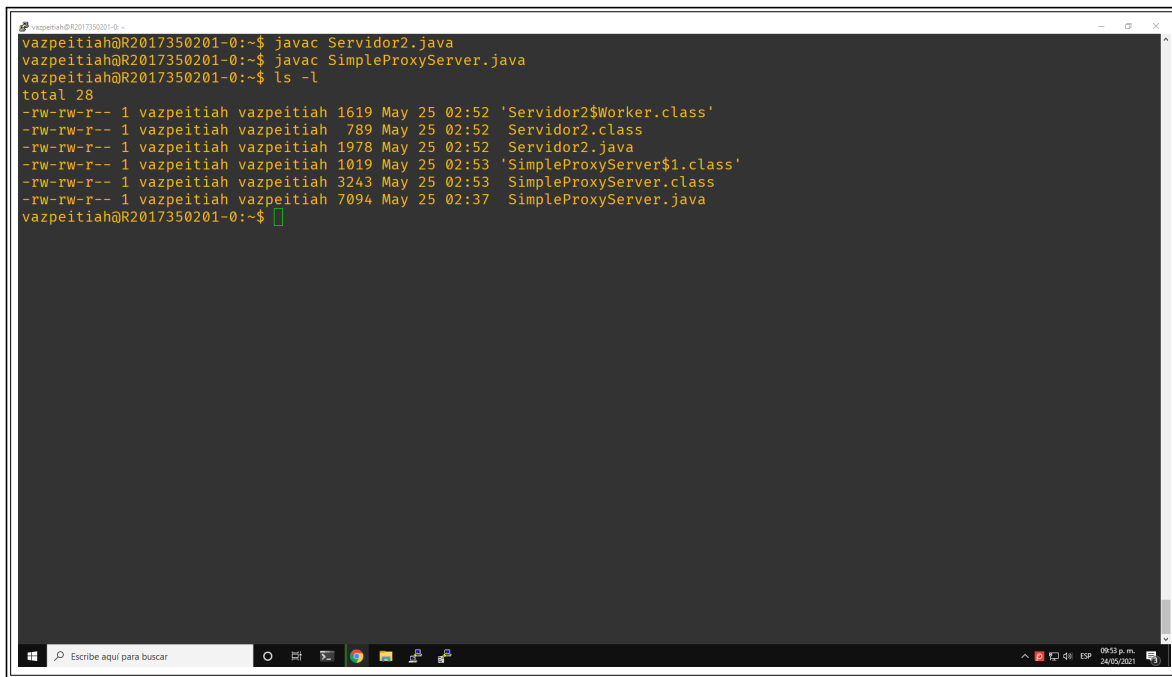
// recibe 5 numeros punto flotante
byte[] a = new byte[5*4];
read(entrada,a,0,5*4);
ByteBuffer b = ByteBuffer.wrap(a);
for (int i = 0; i < 5; i++)
    System.out.println(b.getDouble());

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(8080);

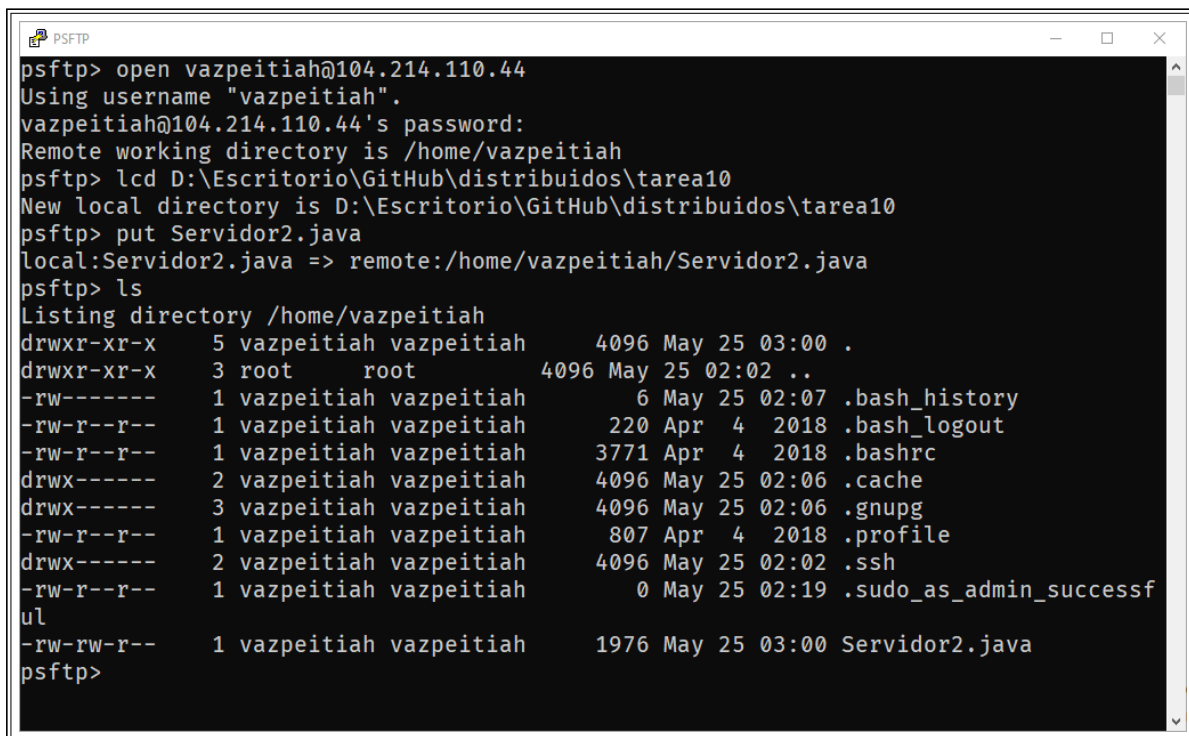
    for (;;)
    {
        Socket conexion = servidor.accept();
        Worker w = new Worker(conexion);
        w.start();
    }
}
Servidor2.java [74,50 Bot]
```

Figura 25: Editando el método *main*, en la máquina virtual 1, de la clase *Servidor2.java*



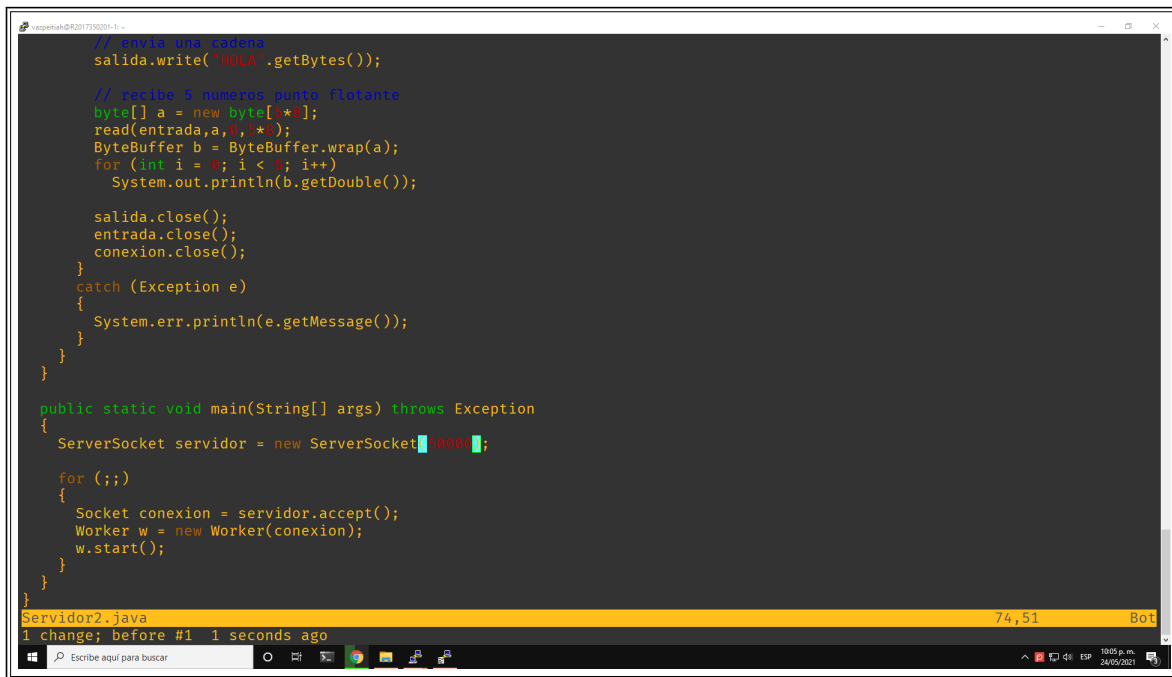
```
vazpeitiah@R2017350201-0:~$ javac Servidor2.java
vazpeitiah@R2017350201-0:~$ javac SimpleProxyServer.java
vazpeitiah@R2017350201-0:~$ ls -l
total 28
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1619 May 25 02:52 'Servidor2$Worker.class'
-rw-rw-r-- 1 vazpeitiah vazpeitiah 789 May 25 02:52 Servidor2.class
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1978 May 25 02:52 Servidor2.java
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1019 May 25 02:53 'SimpleProxyServer$1.class'
-rw-rw-r-- 1 vazpeitiah vazpeitiah 3243 May 25 02:53 SimpleProxyServer.class
-rw-rw-r-- 1 vazpeitiah vazpeitiah 7094 May 25 02:37 SimpleProxyServer.java
vazpeitiah@R2017350201-0:~$
```

Figura 26: Compilando los archivos: Servidor2.java y SimpleProxyServer.java



```
psftp> open vazpeitiah@104.214.110.44
Using username "vazpeitiah".
vazpeitiah@104.214.110.44's password:
Remote working directory is /home/vazpeitiah
psftp> lcd D:\Escritorio\GitHub\distribuidos\tarea10
New local directory is D:\Escritorio\GitHub\distribuidos\tarea10
psftp> put Servidor2.java
local:Servidor2.java => remote:/home/vazpeitiah/Servidor2.java
psftp> ls
Listing directory /home/vazpeitiah
drwxr-xr-x  5 vazpeitiah vazpeitiah    4096 May 25 03:00 .
drwxr-xr-x  3 root      root          4096 May 25 02:02 ..
-rw-r----- 1 vazpeitiah vazpeitiah     6 May 25 02:07 .bash_history
-rw-r--r--  1 vazpeitiah vazpeitiah    220 Apr  4 2018 .bash_logout
-rw-r--r--  1 vazpeitiah vazpeitiah   3771 Apr  4 2018 .bashrc
drwx----- 2 vazpeitiah vazpeitiah   4096 May 25 02:06 .cache
drwx----- 2 vazpeitiah vazpeitiah   4096 May 25 02:06 .gnupg
-rw-r--r--  1 vazpeitiah vazpeitiah    807 Apr  4 2018 .profile
drwx----- 2 vazpeitiah vazpeitiah   4096 May 25 02:02 .ssh
-rw-r--r--  1 vazpeitiah vazpeitiah     0 May 25 02:19 .sudo_as_admin_successf
ul
-rw-rw-r--  1 vazpeitiah vazpeitiah   1976 May 25 03:00 Servidor2.java
psftp>
```

Figura 27: Utilizando el programa psftp.exe para enviar a la máquina virtual 2 el archivo Servidor2.java



```

// envia una cadena
salida.write("HOLA".getBytes());

// recibe 5 numeros punto flotante
byte[] a = new byte[5*4];
read(entrada,a,0,5*4);
ByteBuffer b = ByteBuffer.wrap(a);
for (int i = 0; i < 5; i++)
    System.out.println(b.getDouble());

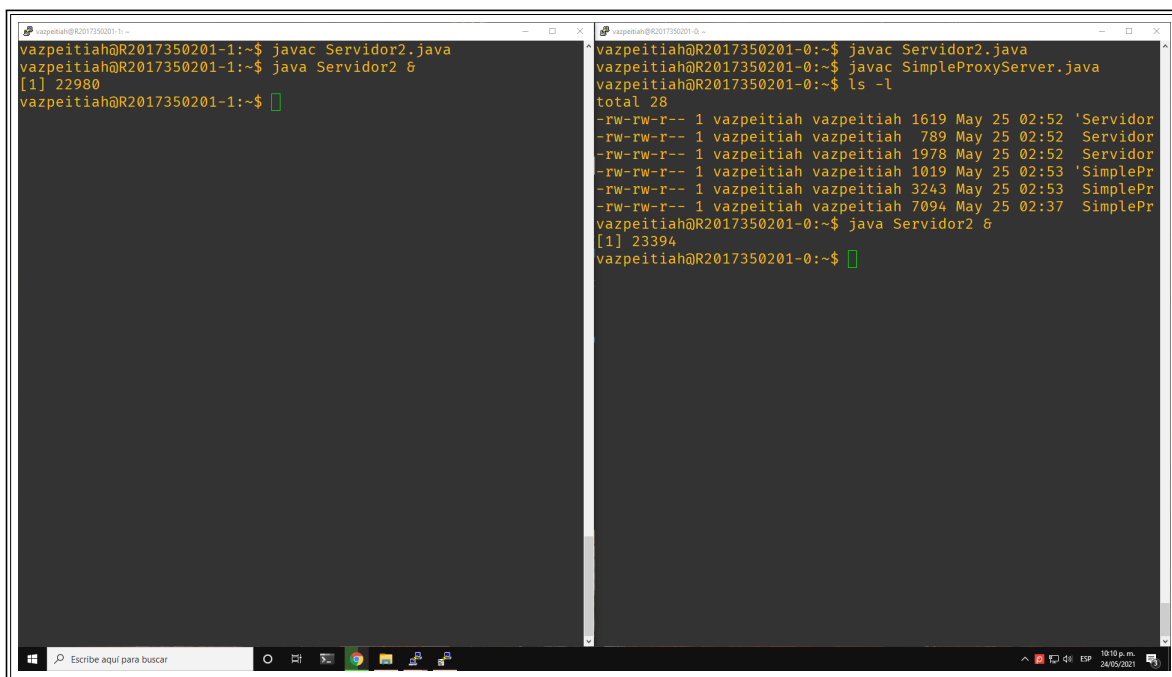
salida.close();
entrada.close();
conexion.close();
}
catch (Exception e)
{
    System.err.println(e.getMessage());
}
}

public static void main(String[] args) throws Exception
{
    ServerSocket servidor = new ServerSocket(8080);

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

```

Figura 28: Editando el método *main*, en la máquina virtual 2, de la clase *Servidor2.java*



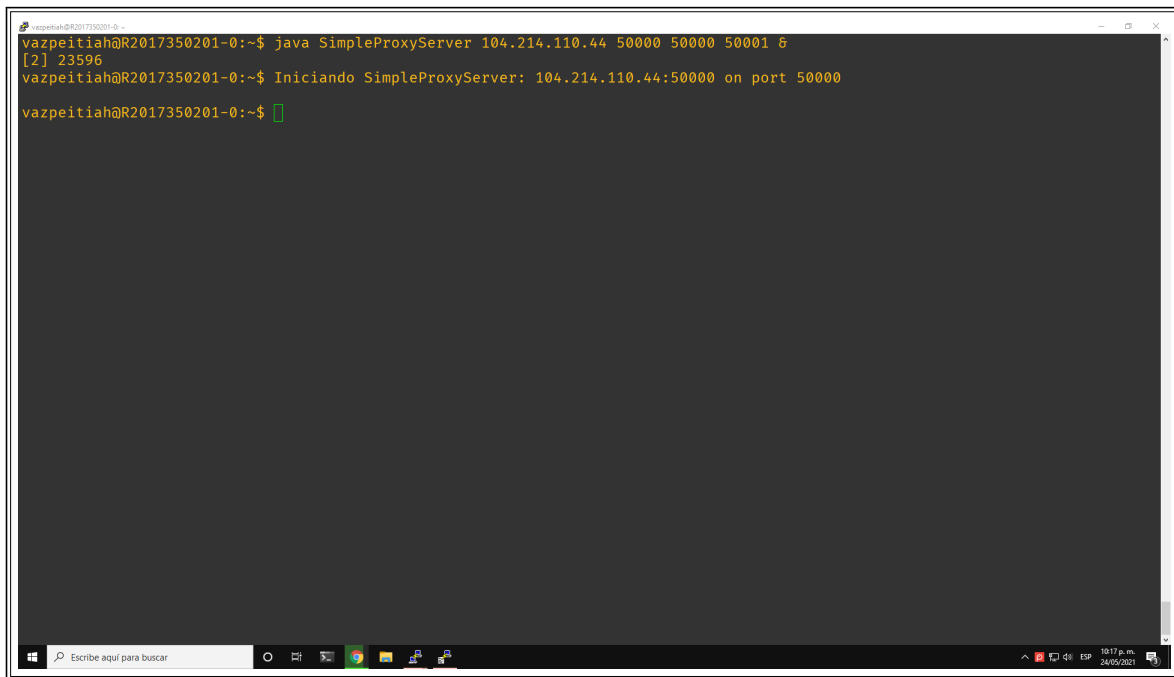
```

vazpeitiah@R2017350201-1:~$ javac Servidor2.java
vazpeitiah@R2017350201-1:~$ java Servidor2 6
[1] 22980
vazpeitiah@R2017350201-1:~$

vazpeitiah@R2017350201-0:~$ javac Servidor2.java
vazpeitiah@R2017350201-0:~$ javac SimpleProxyServer.java
vazpeitiah@R2017350201-0:~$ ls -l
total 28
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1619 May 25 02:52 'Servidor
-rw-rw-r-- 1 vazpeitiah vazpeitiah 789 May 25 02:52 'Servidor
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1978 May 25 02:52 'Servidor
-rw-rw-r-- 1 vazpeitiah vazpeitiah 1019 May 25 02:53 'SimplePr
-rw-rw-r-- 1 vazpeitiah vazpeitiah 3243 May 25 02:53 'SimplePr
-rw-rw-r-- 1 vazpeitiah vazpeitiah 7094 May 25 02:37 'SimplePr
vazpeitiah@R2017350201-0:~$ java Servidor2 6
[1] 23394
vazpeitiah@R2017350201-0:~$

```

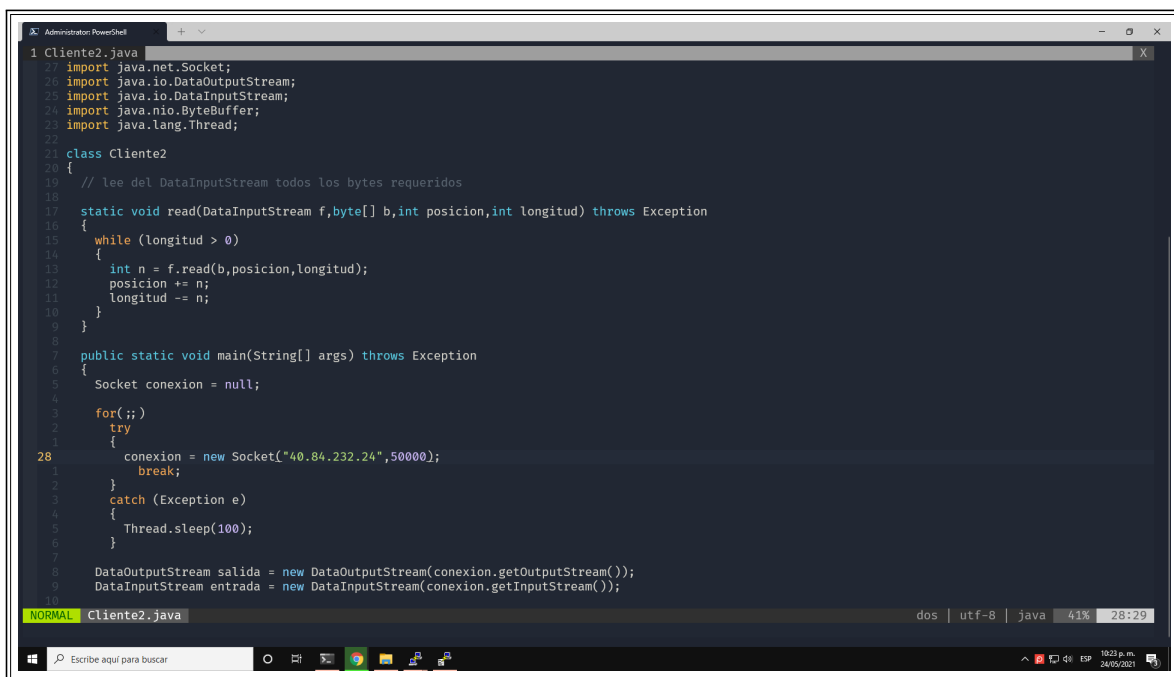
Figura 29: Compilando la clase *Servidor2.java*, en la máquina 2, y ejecutandola en segundo plano en ambas máquinas



```
vazpeitiah@R2017350201-0:~$ java SimpleProxyServer 104.214.110.44 50000 50000 50001 8
[2] 23596
vazpeitiah@R2017350201-0:~$ Iniciando SimpleProxyServer: 104.214.110.44:50000 on port 50000

vazpeitiah@R2017350201-0:~$
```

Figura 30: Ejecutando, en la máquina virtual 1, el proxy



```
1 Cliente2.java
2 import java.net.Socket;
3 import java.io.DataOutputStream;
4 import java.io.DataInputStream;
5 import java.nio.ByteBuffer;
6 import java.lang.Thread;
7
8 class Cliente2
9 {
10     // lee del DataInputStream todos los bytes requeridos
11     static void read(DataInputStream f,byte[] b,int posicion,int longitud) throws Exception
12     {
13         while (longitud > 0)
14         {
15             int n = f.read(b,posicion,longitud);
16             posicion += n;
17             longitud -= n;
18         }
19     }
20
21     public static void main(String[] args) throws Exception
22     {
23         Socket conexion = null;
24         for(;;)
25         {
26             try
27             {
28                 conexion = new Socket("40.84.232.24",50000);
29                 break;
30             }
31             catch (Exception e)
32             {
33                 Thread.sleep(100);
34             }
35         }
36
37         DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());
38         DataInputStream entrada = new DataInputStream(conexion.getInputStream());
39     }
40 }
```

Figura 31: Editar el programa Cliente2.java para que se conecte a la máquina virtual 1.

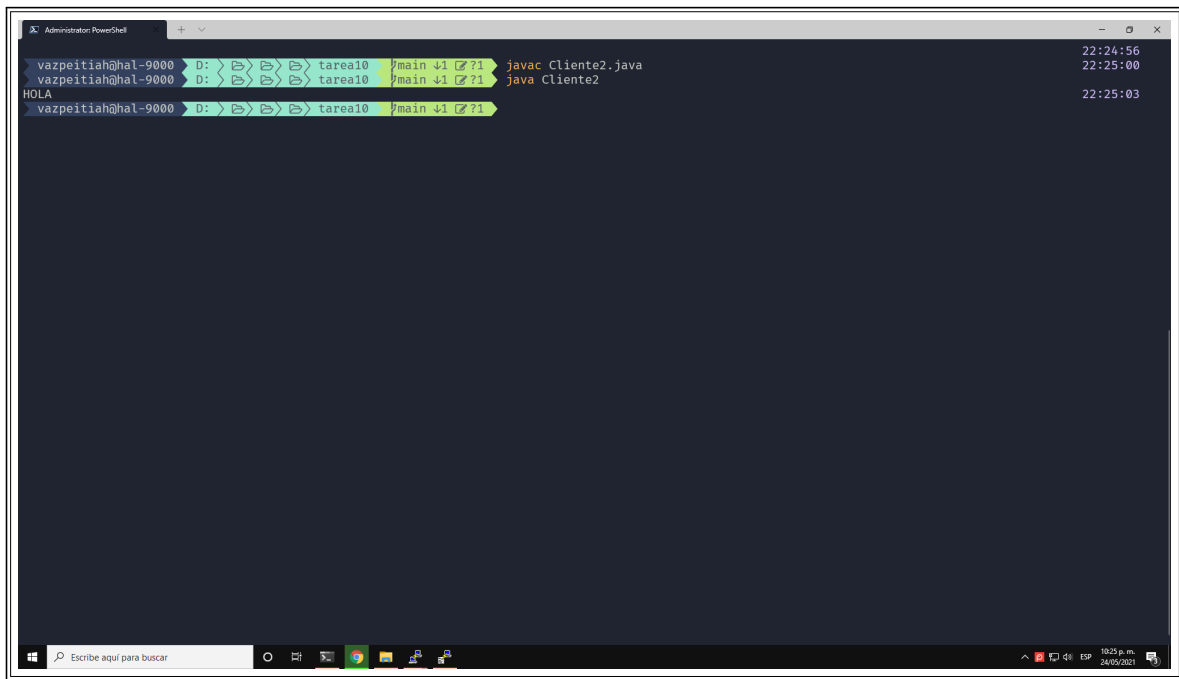


Figura 32: Compilando y ejecutando el programa Cliente2.java, desde Windows 10

4. Conclusiones

Como pudimos observar en esta tarea. Se puede replicar un sistema y se puede acceder a este sistema junto con sus funcionalidades sin problema alguno. Es bueno considerar ciertos beneficios como la seguridad, ya que al tener la replicación de datos pueden caer en malas manos, por lo que tener guardados los datos y archivos en un backup o algún sistema de base de datos para la recuperación de estos es bastante factible.