



***Instituto Politécnico Nacional***  
***Escuela Superior de Cómputo***



## ***Desarrollo de Sistemas Distribuidos***

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

***Nombre:*** Sampayo Hernández Mauro

***Grupo:*** 4CV1

***Profesor:*** Pineda Guerrero Carlos

## Creación de las máquinas virtuales:

Se realiza la creación de dos máquinas virtuales las cuales estarán configuradas con una imagen del S.O. Ubuntu Server 18.04 LTS, un tamaño de memoria de 1 GB de RAM, una contraseña como tipo de autenticación y con un disco HDD estándar como disco del sistema operativo.

Se inicia configurando la sección de “Datos Básicos” de las máquinas virtuales, donde ingresaremos los nombres tanto de usuarios como de estas, y sus respectivas contraseñas; seleccionaremos la región y el tamaño de las máquinas virtuales y dejaremos abierto en las "Reglas de puerto de entrada" el puerto 22 para utilizar SSH (la terminal de secure shell).

- **Máquina Virtual 1**

Microsoft Azure

Crear una máquina virtual

Datos básicos | Discos | Redes | Administración | Opciones avanzadas | Etiquetas | Revisar y crear

Creo una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración. [Más información](#)

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \*

Grupo de recursos \*  [Crear nuevo](#)

**Detalles de instancia**

Nombre de máquina virtual \*

Región \*

Opciones de disponibilidad \*

Imagen \*  [Ver todas las imágenes](#)

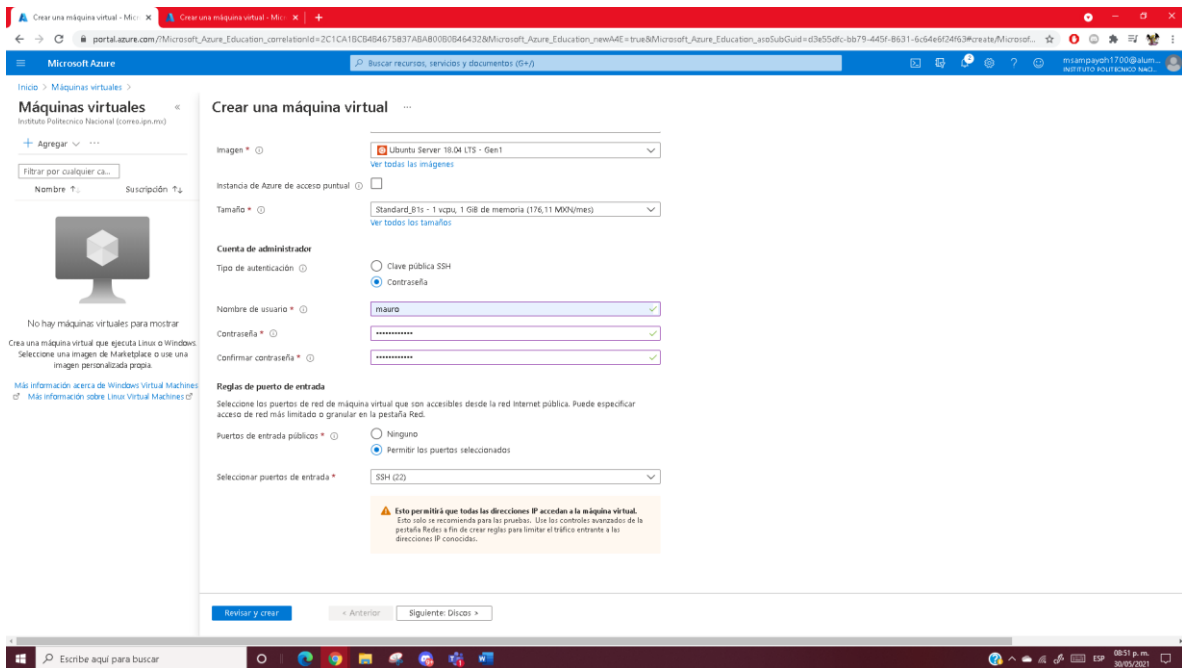
Instancia de Azure de acceso puntual ☐

Tamaño \*  [Ver todos los tamaños](#)

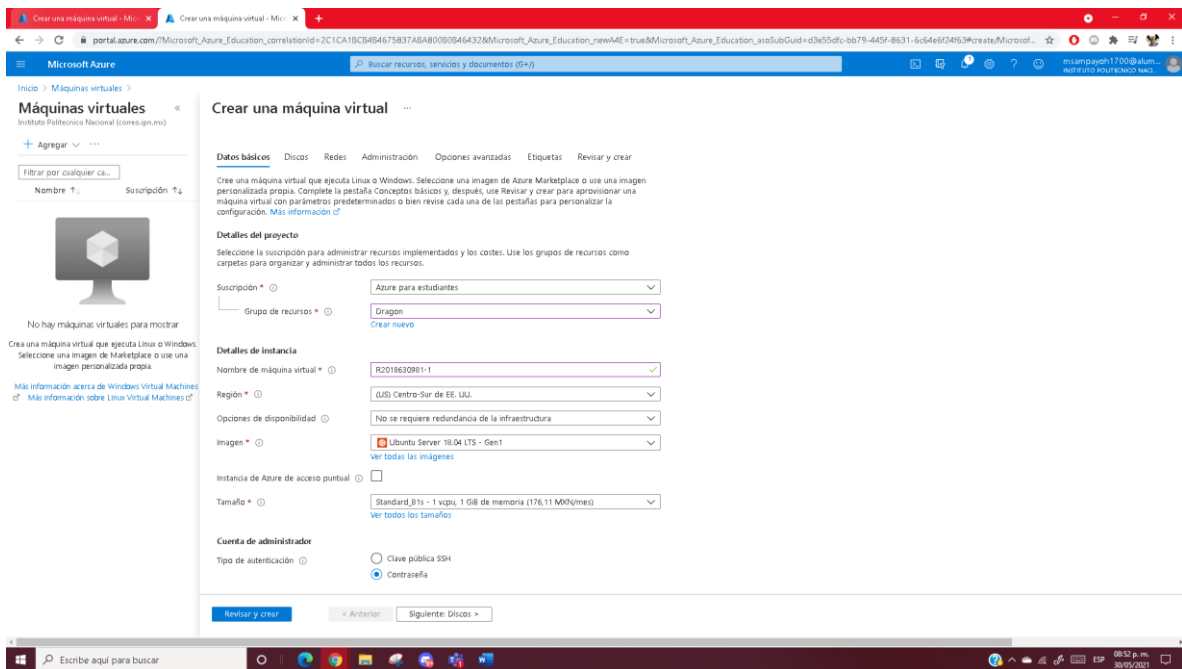
**Cuenta de administrador**

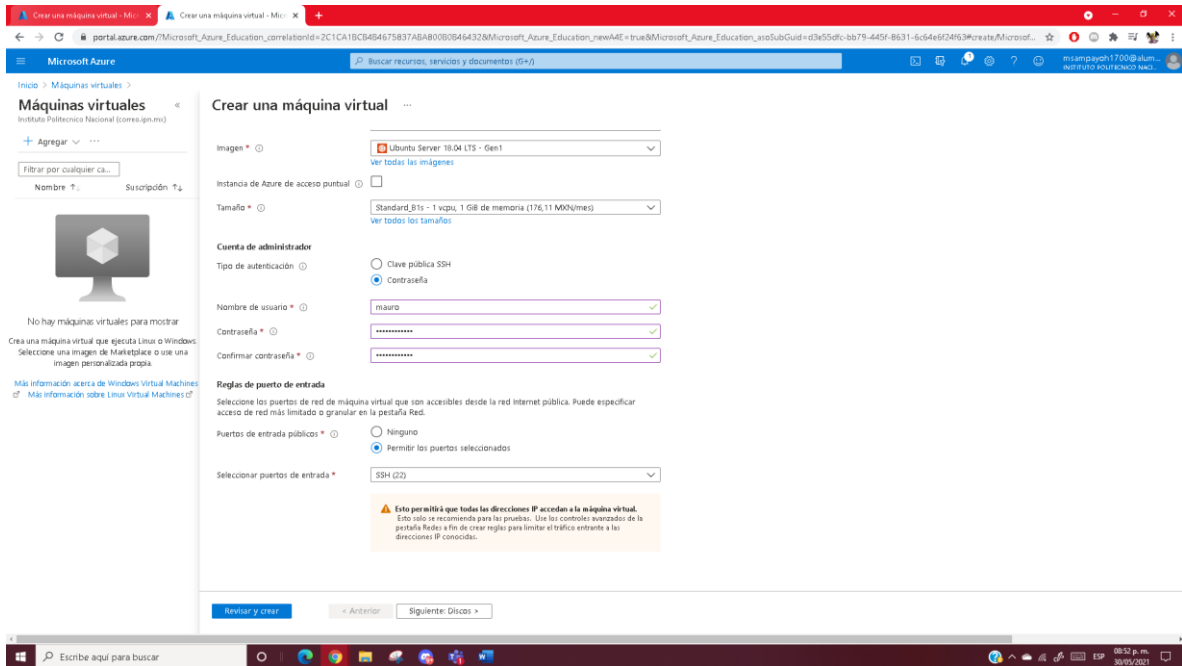
Tipo de autenticación \* ☐ Clave pública SSH ☒ Contraseña

[Revisar y crear](#) [Anterior](#) [Siguiente: Discos](#)



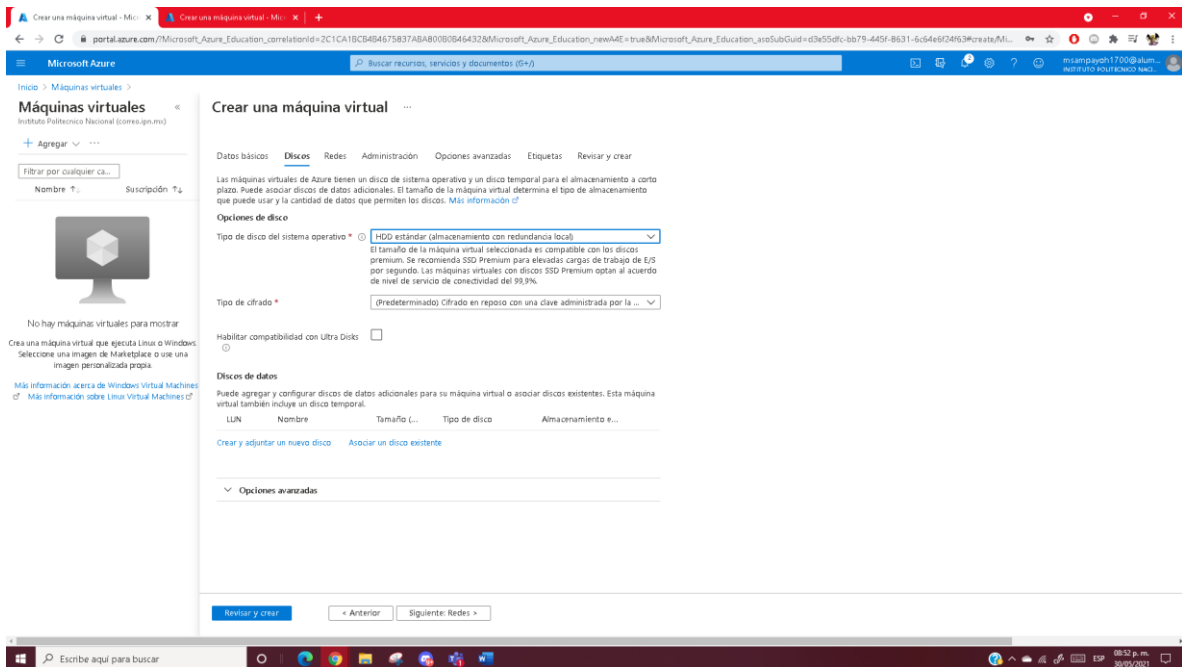
## • Máquina Virtual 2



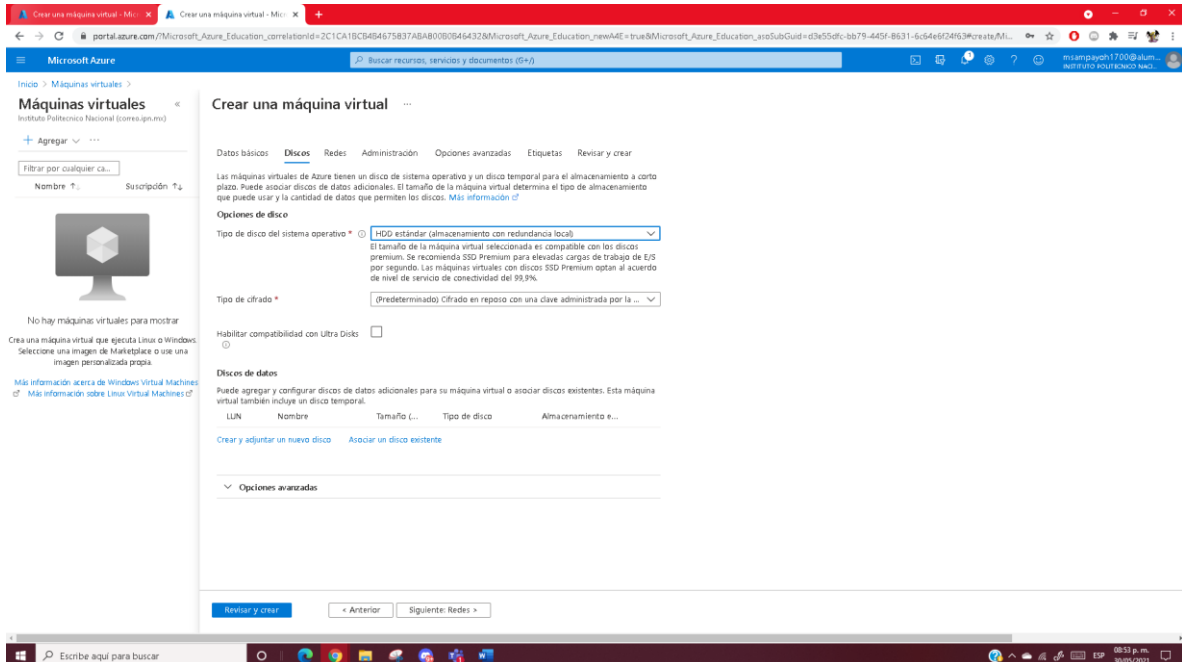


En la sección de “Discos” seleccionamos la opción “HDD estándar” como tipo de disco del sistema operativo de las máquinas virtuales.

- **Máquina Virtual 1**

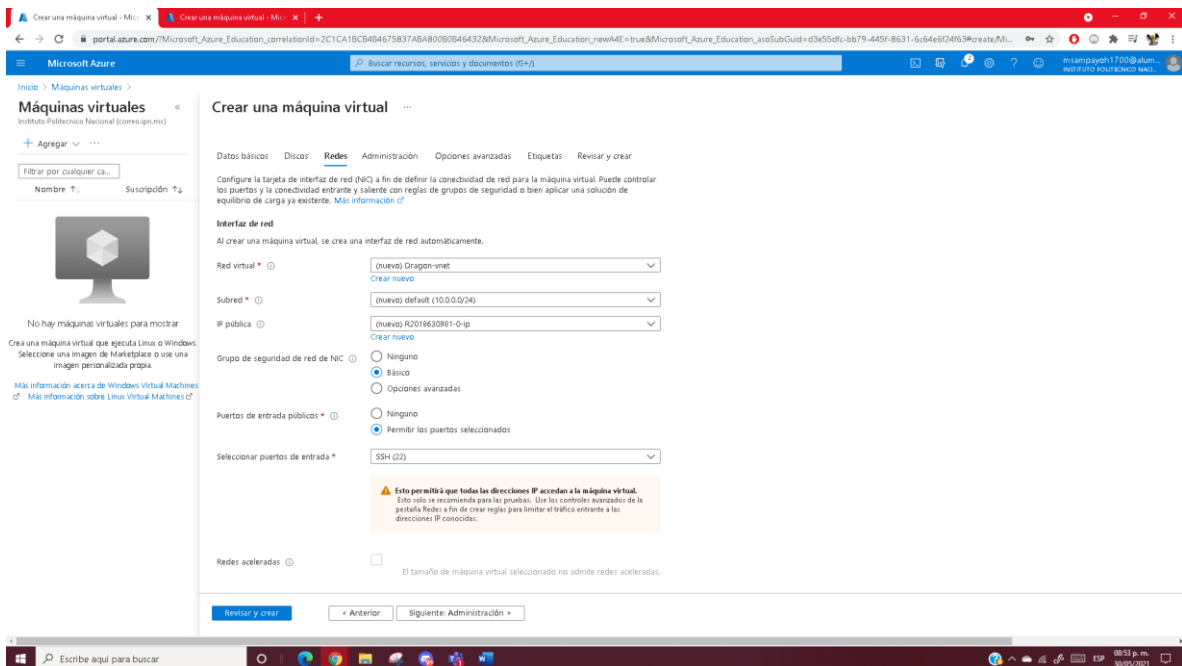


- Máquina Virtual 2

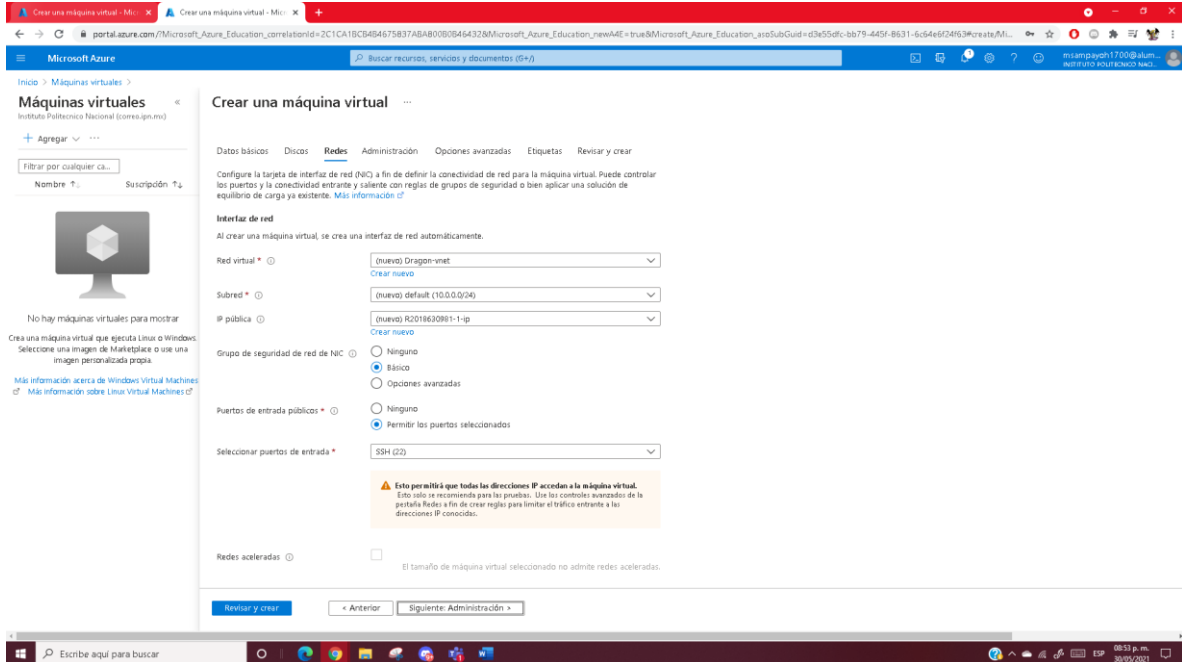


La sección de “Redes” se deja sin alterar, y con su configuración por defecto.

- Máquina Virtual 1

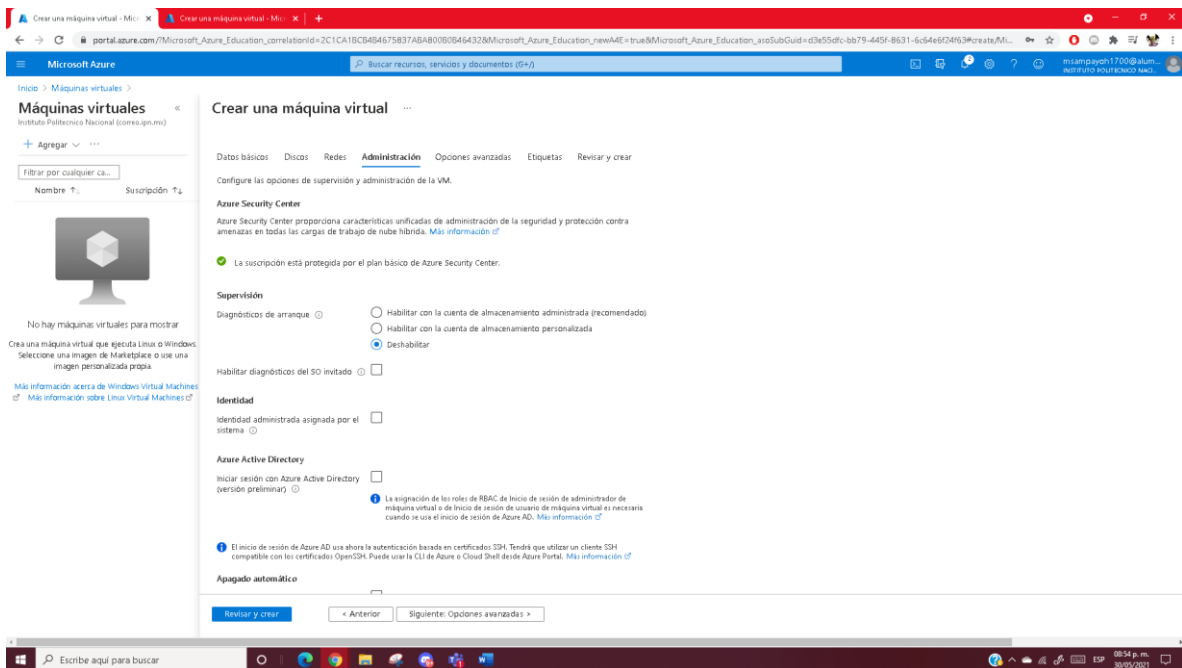


## • Máquina Virtual 2

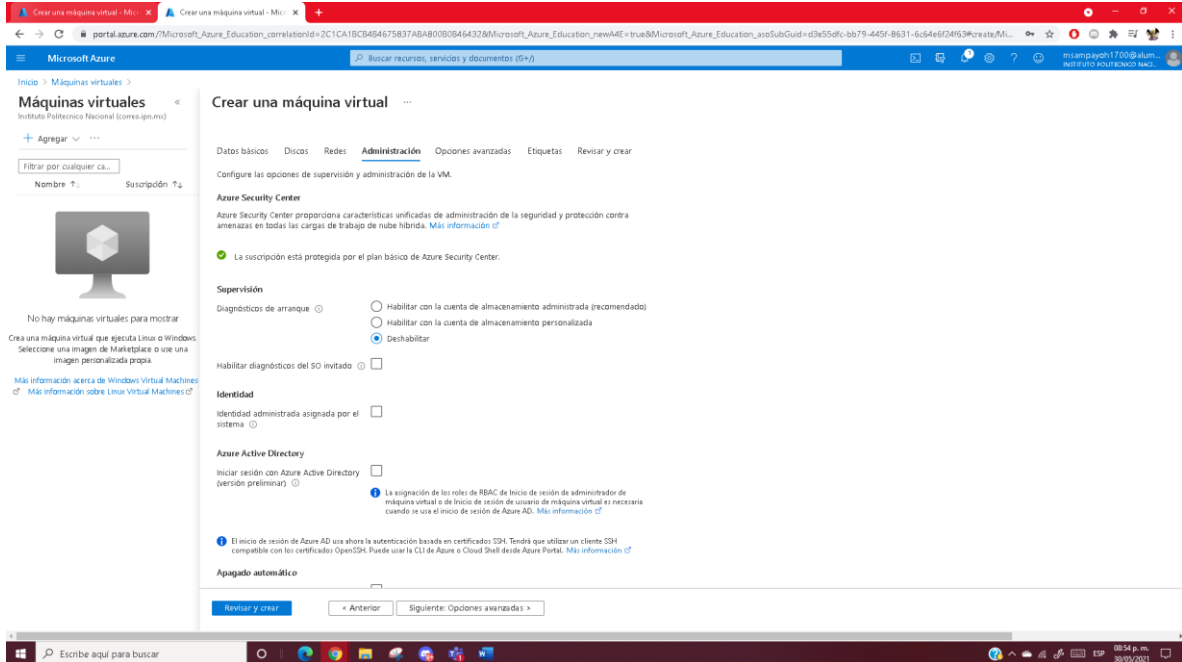


En la sección de “Administración” dejaremos el campo de “Diagnóstico de Arranque” como “Deshabilitado”.

## • Máquina Virtual 1

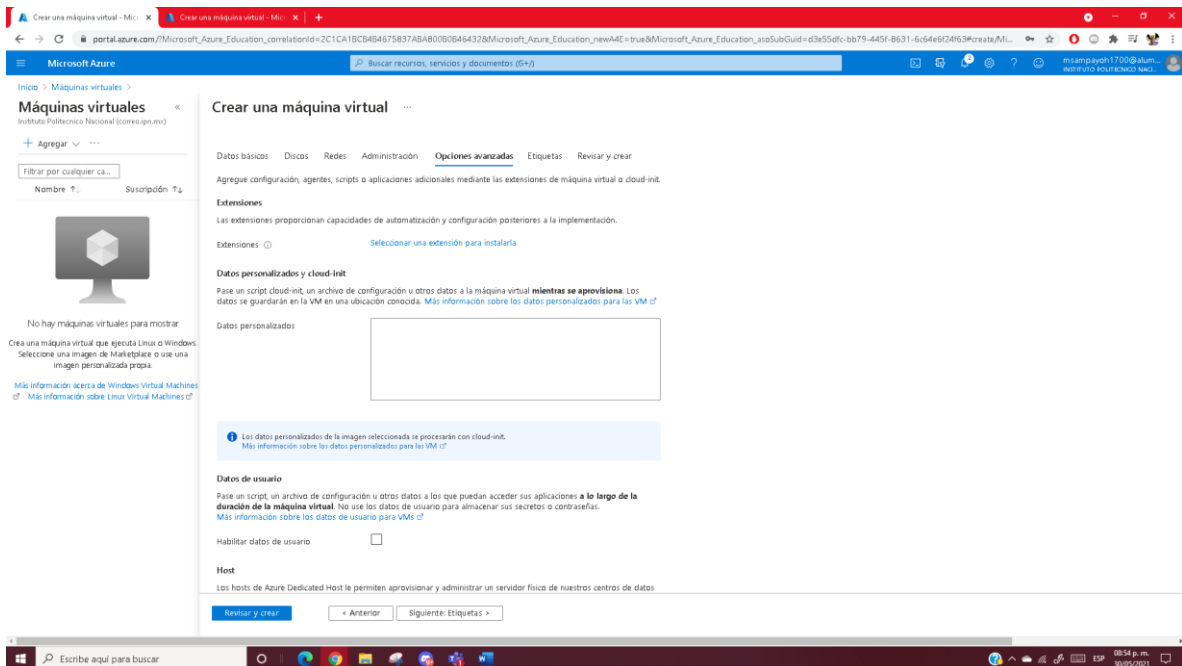


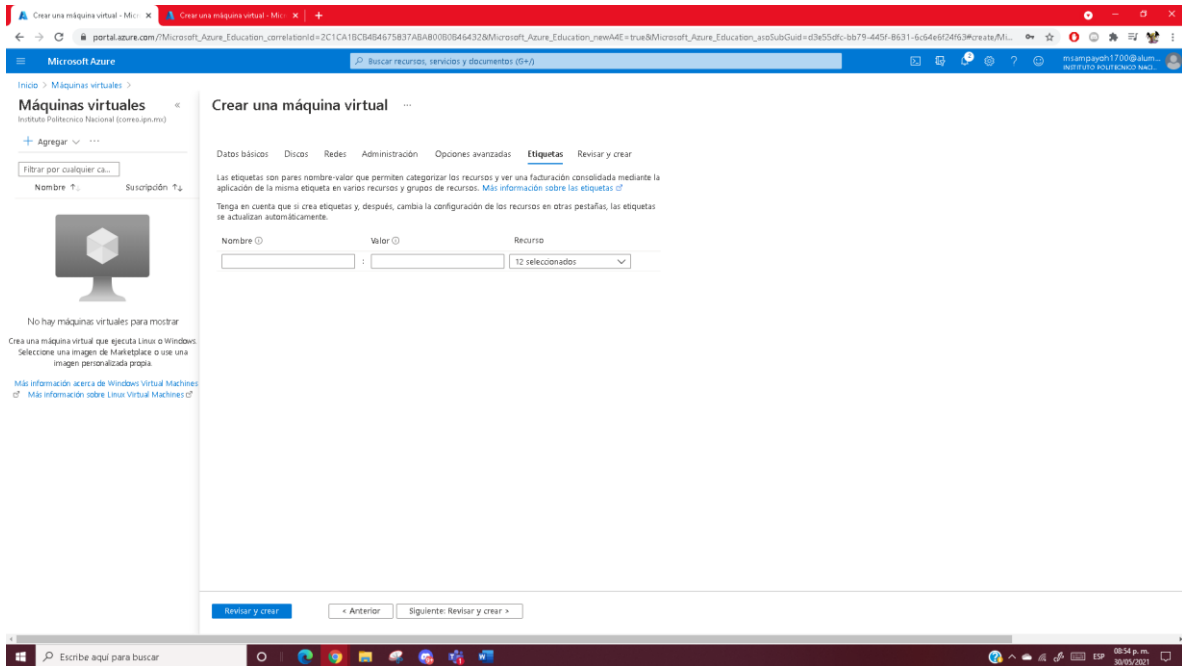
## • Máquina Virtual 2



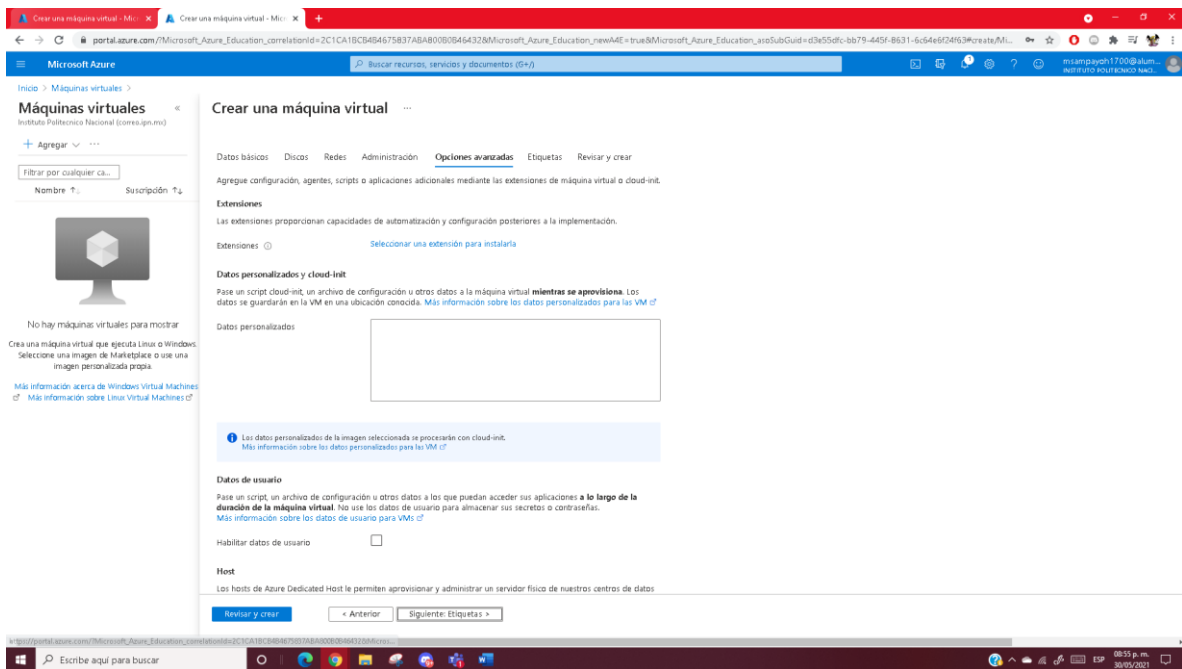
Finalmente, las secciones “Opciones Avanzadas” y “Etiquetas” se dejarán sin modificación alguna, con su configuración por defecto.

## • Máquina Virtual 1

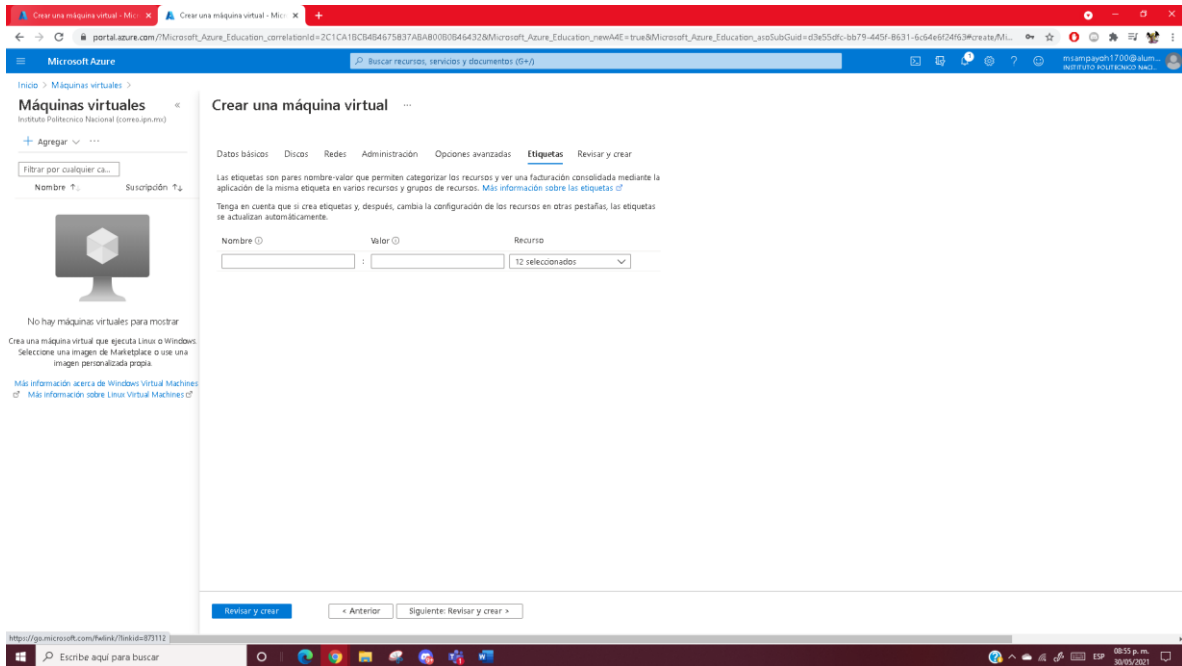




## • Máquina Virtual 2

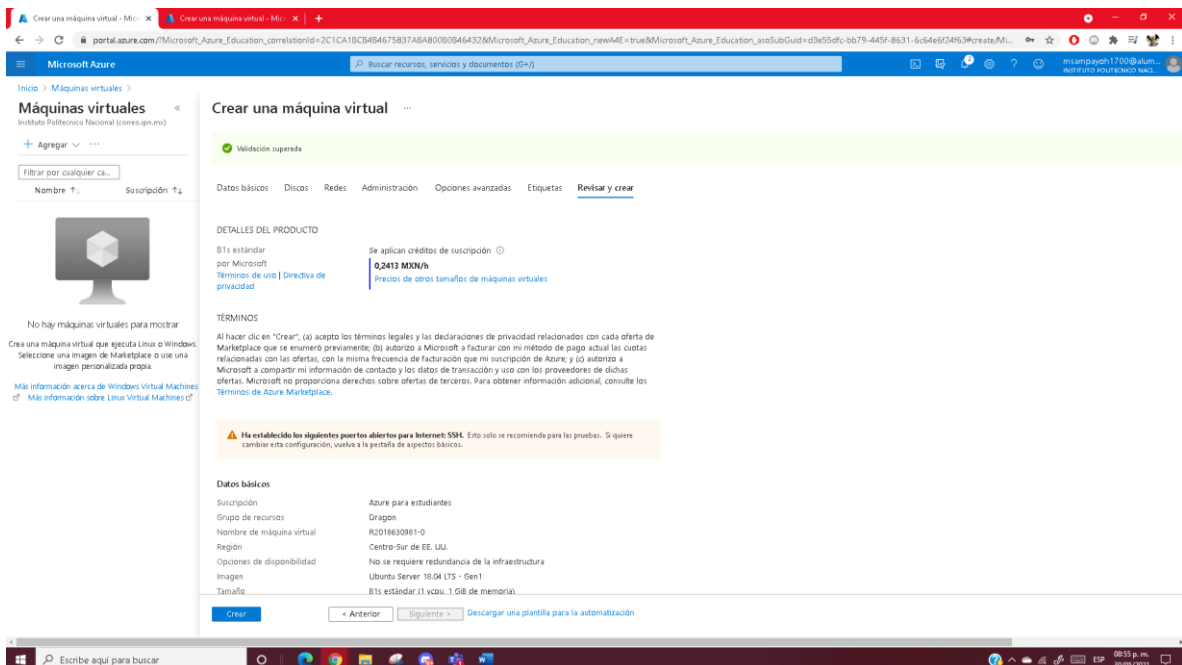




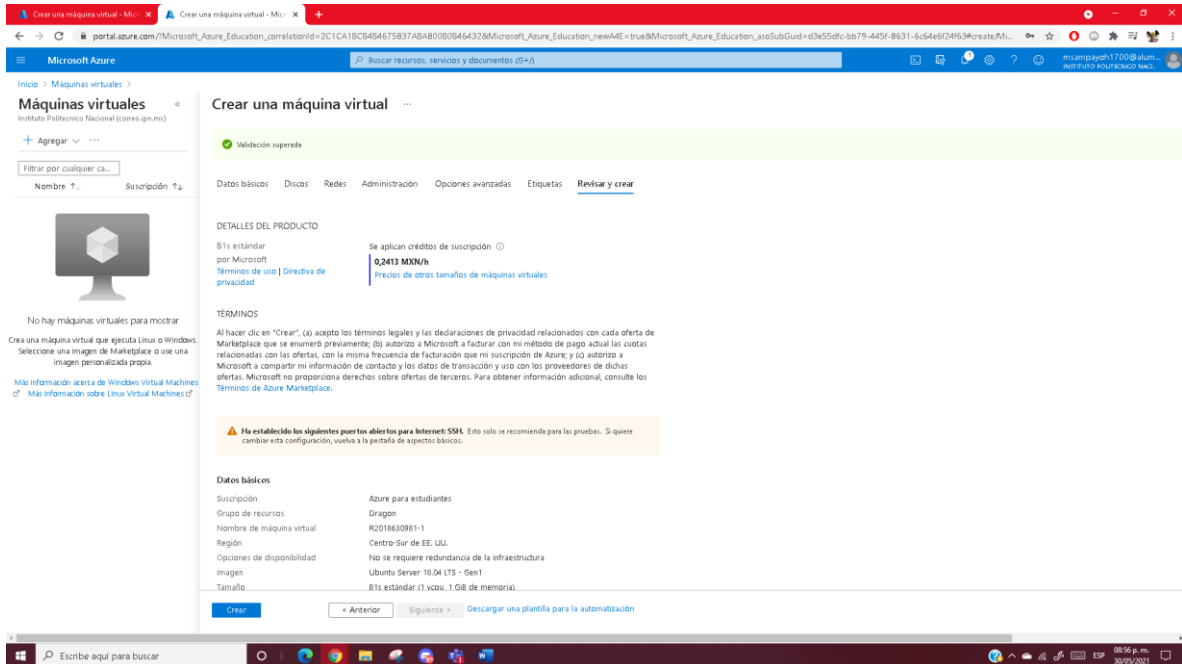


Damos click en el botón “Revisar y Crear”.

- **Máquina Virtual 1**

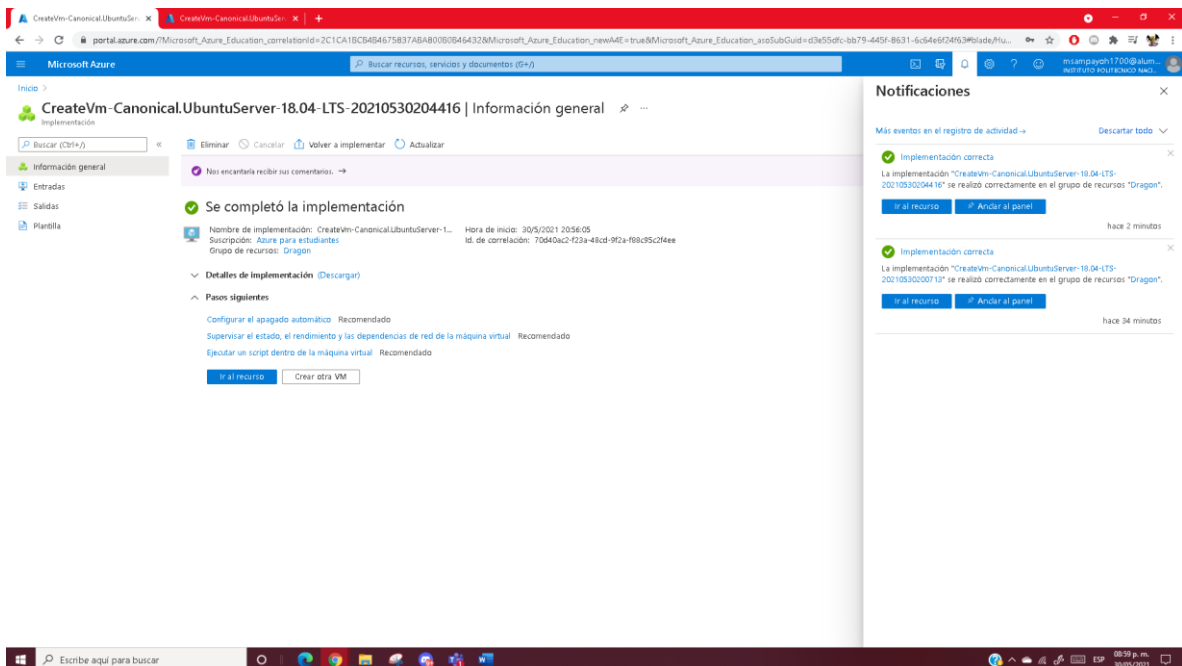


## • Máquina Virtual 2

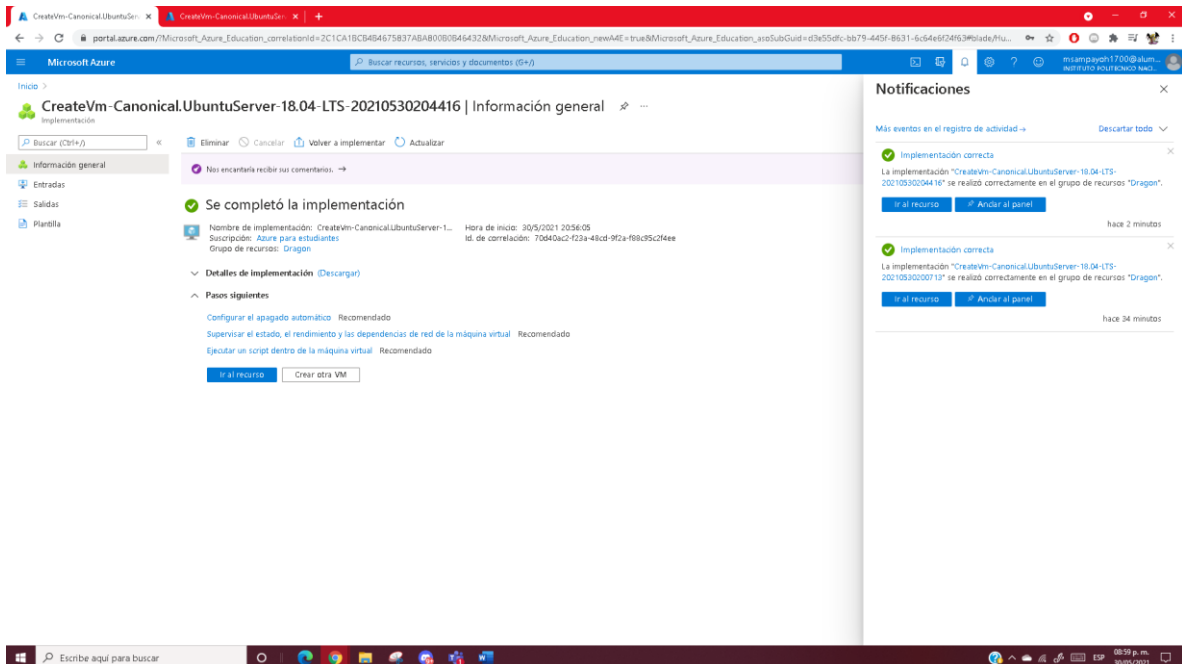


Una vez las máquinas virtuales son validadas, se da click en el botón de “Crear” para así completar su implementación.

## • Máquina Virtual 1



- **Máquina Virtual 2**



Las IPs públicas para las máquinas virtuales son:

- **Maquina virtual R2018630981-0:** 20.97.7.162
- **Maquina virtual R2018630981-1:** 20.97.4.109

A partir de este punto se continuará con la implementación de la practica tomando como base el procedimiento propuesto por el profesor.

## Desarrollo de la Tarea

### 1. Crear dos máquinas virtuales en la nube de Azure con Ubuntu 18, 1 GB de RAM y disco HDD estándar.

#### • Máquina Virtual 1

The screenshot displays the Azure portal interface for a virtual machine named "R2018630981-0". The left sidebar contains navigation options such as "Información general", "Registro de actividad", "Control de acceso (IAM)", "Etiquetas", "Diagnóstico y solución pro...", "Configuración", "Redes", "Conectar", "Discos", "Tamaño", "Seguridad", "Recomendaciones de Advisor", "Extensiones", "Entrega continua", "Disponibilidad y escalado", "Configuración", "Identidad", "Propiedades", "Bloqueos", "Operaciones", "Estado", "Apagado automático", "Backup", and "Recuperación ante desastres". The main content area is divided into several sections:

- Información esencial:** Includes details like "Grupo de recursos", "Estado", "Ubicación", "Suscripción", "Id. de suscripción", "Etiquetas", "Sistema operativo", "Tamaño", "Dirección IP pública", "Red virtual/subred", and "Nombre DNS".
- Propiedades:** A table listing various properties of the virtual machine, including "Nombre del equipo", "Sistema operativo", "Publicador", "Oferta", "Plan", "Generación de VM", "Estado del agente", "Versión del agente", "Grupo host", "Host", "Grupo con ubicación por proximidad", "Estado de ubicación", "Zona de disponibilidad", and "Conjunto de escalado".
- Redes:** A table showing network configuration, including "Dirección IP pública", "Dirección IP pública (IPv6)", "Dirección IP privada", "Dirección IP privada (IPv6)", "Red virtual/subred", and "Nombre DNS".
- Tamaño:** A table showing the size of the virtual machine, including "Tamaño", "vCPU", and "RAM".
- Discos:** A table showing the disks attached to the virtual machine, including "Disco del SO", "Azure Disk Encryption", "Disco de SO efímero", and "Discos de datos".
- Extensiones:** A section for managing extensions, currently showing "Azure de acceso puntual".

#### • Máquina Virtual 2

The screenshot displays the Azure portal interface for a virtual machine named "R2018630981-1". The layout is similar to the first screenshot, showing the same navigation sidebar and main content sections:

- Información esencial:** Details for the virtual machine, including "Grupo de recursos", "Estado", "Ubicación", "Suscripción", "Id. de suscripción", "Etiquetas", "Sistema operativo", "Tamaño", "Dirección IP pública", "Red virtual/subred", and "Nombre DNS".
- Propiedades:** A table listing various properties of the virtual machine, including "Nombre del equipo", "Sistema operativo", "Publicador", "Oferta", "Plan", "Generación de VM", "Estado del agente", "Versión del agente", "Grupo host", "Host", "Grupo con ubicación por proximidad", "Estado de ubicación", "Zona de disponibilidad", and "Conjunto de escalado".
- Redes:** A table showing network configuration, including "Dirección IP pública", "Dirección IP pública (IPv6)", "Dirección IP privada", "Dirección IP privada (IPv6)", "Red virtual/subred", and "Nombre DNS".
- Tamaño:** A table showing the size of the virtual machine, including "Tamaño", "vCPU", and "RAM".
- Discos:** A table showing the disks attached to the virtual machine, including "Disco del SO", "Azure Disk Encryption", "Disco de SO efímero", and "Discos de datos".
- Extensiones:** A section for managing extensions, currently showing "Azure de acceso puntual".

## 2. Abrir el puerto 50000 protocolo TCP en la máquina virtual 1.

Accedemos a la sección de “Redes” de la máquina virtual y damos click el botón “Agregar regla de puerto de entrada” para así abrir el puerto 50000.

The screenshot shows the 'Agregar regla de seguridad de entrada' (Add inbound security rule) dialog in the Microsoft Azure portal. The rule is being configured for the VM R2018630981-0. The configuration includes:

- Origen (Origin):** Any
- Intervalos de puertos de origen (Origin port ranges):** \*
- Destino (Destination):** Any
- Servicio (Service):** Custom
- Intervalos de puertos de destino (Destination port ranges):** 50000
- Protocolo (Protocol):** TCP
- Acción (Action):** Permitir (Allow)
- Prioridad (Priority):** 310
- Nombre (Name):** Port\_50000

The background shows the 'Reglas de punto de entrada' (Inbound rules) table for the VM, which lists existing rules for SSH, VirtualNetwork, and AzureLoadBalancer.

Seleccionamos “Agregar”.

The screenshot shows the 'Reglas de punto de entrada' (Inbound rules) table for the VM R2018630981-0. A new rule has been added, highlighted with a red box:

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
310	Port_50000	50000	TCP	Cualquiera	Cualquiera	Permitir
65000	AllowVirtualNetwork			VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInbound			AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInbound			Cualquiera	Cualquiera	Denegar

### 3. Abrir el puerto 50000 protocolo TCP en la máquina virtual 2.

Accedemos a la sección de “Redes” de la máquina virtual y damos click el botón “Agregar regla de puerto de entrada” para así abrir el puerto 50000.

The screenshot shows the Azure portal interface for a virtual machine named R2018630981-1. The 'Redes' (Network) section is active. A dialog box titled 'Agregar regla de seguridad de entrada' (Add inbound security rule) is open. The rule is being configured with the following settings:

- Origen (Source):** Any
- Intervalos de puertos de origen (Source port ranges):** \*
- Destino (Destination):** Any
- Servicio (Service):** Custom
- Intervalos de puertos de destino (Destination port ranges):** 50000
- Protocolo (Protocol):** TCP
- Acción (Action):** Permitir (Allow)
- Prioridad (Priority):** 310
- Nombre (Name):** Port\_50000

The background shows the network configuration for the VM, including the public IP address 20.97.4.109 and the current inbound security rules.

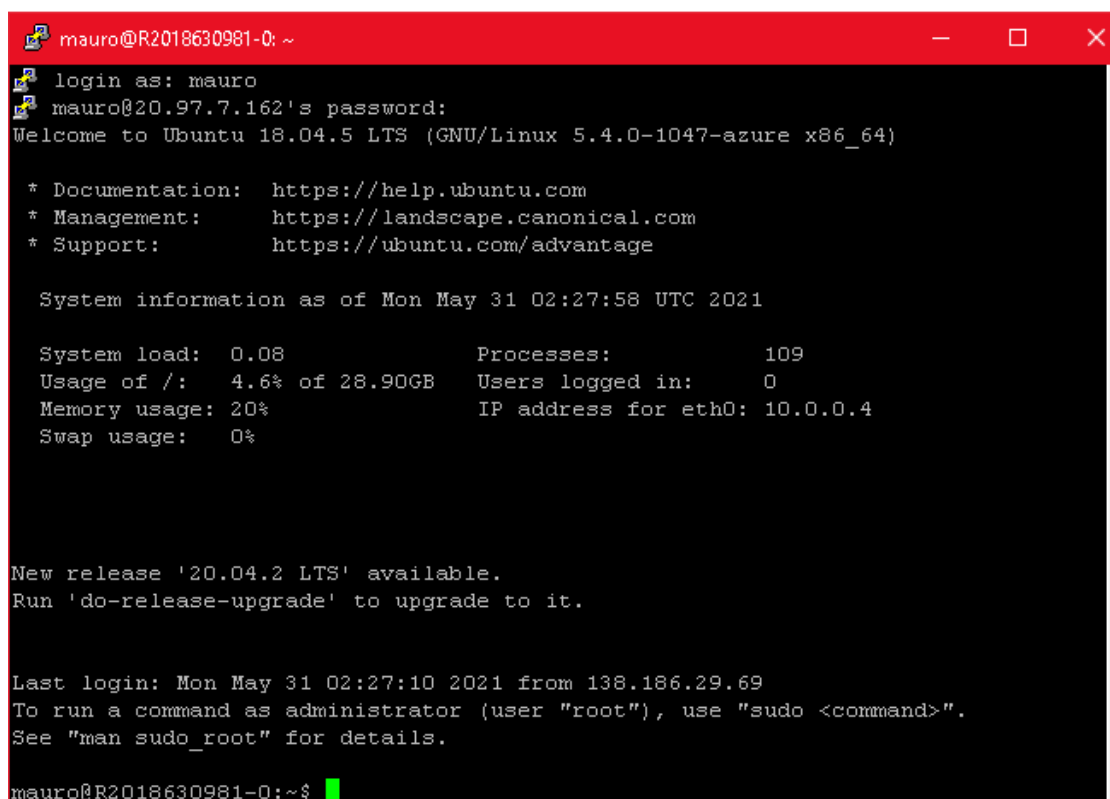
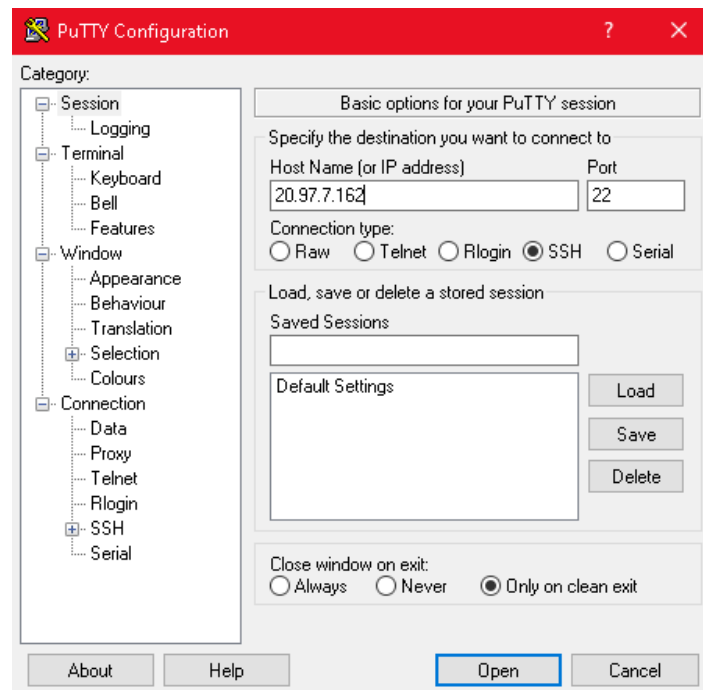
Seleccionamos “Agregar”.

The screenshot shows the Azure portal interface for the same virtual machine. The 'Reglas de punto de entrada' (Inbound security rules) table is displayed. The rule 'Port\_50000' is highlighted with a red box, showing it is configured to allow TCP traffic on port 50000 from any source to the VM's public IP (20.97.4.109).

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
300	SSH	22	TCP	Cualquiera	Cualquiera	Permitir
310	Port_50000	50000	TCP	Cualquiera	Cualquiera	Permitir
65000	AllowVnetInbound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir
65001	AllowAzureLoadBalancerInbound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir
65500	DenyAllInbound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar

#### 4. Conectar a la máquina virtual 1 (sistema principal) utilizando el programa putty.exe

Ejecutamos putty.exe y accedemos a la máquina virtual ingresando la IP pública en el programa y posteriormente el usuario y la contraseña de esta.

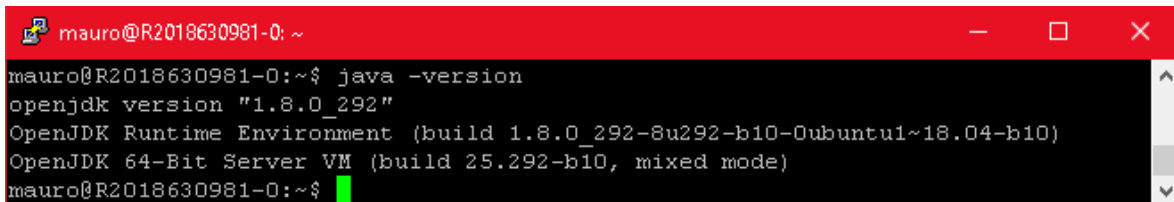


5. Instalar JDK-8 en la máquina virtual 1, ejecutando los comandos:

```
sudo apt update
```

```
sudo apt install openjdk-8-jdk-headless
```

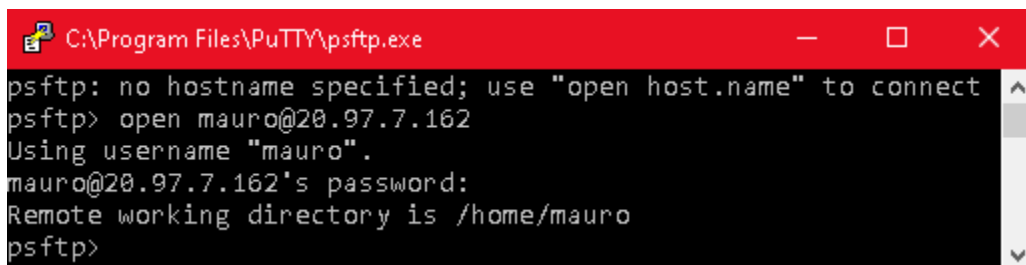
Comprobamos la instalación con el comando “*java -version*”.



```
mauro@R2018630981-0: ~  
mauro@R2018630981-0:~$ java -version  
openjdk version "1.8.0_292"  
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10)  
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)  
mauro@R2018630981-0:~$
```

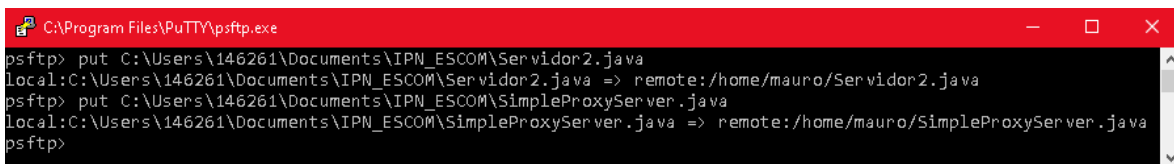
6. Utilizando el programa psftp.exe enviar a la máquina virtual 1 los archivos: [Servidor2.java](#) y [SimpleProxyServer.java](#)

Ejecutamos el programa psftp.exe y ejecutamos el comando “*open <open @direccionIPpublica>*” con los datos del usuario e IP pública de la maquina virtual 1, seguido de la contraseña del usuario. De esta manera podremos trabajar en el directorio remoto de la máquina virtual.



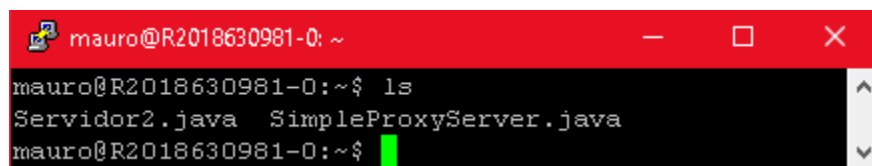
```
C:\Program Files\PuTTY\psftp.exe  
psftp: no hostname specified; use "open host.name" to connect  
psftp> open mauro@20.97.7.162  
Using username "mauro".  
mauro@20.97.7.162's password:  
Remote working directory is /home/mauro  
psftp>
```

Enviamos los archivos Servidor2.java y SimpleProxyServer.java con el comando “*put <rutaArchivo>*”.



```
C:\Program Files\PuTTY\psftp.exe  
psftp> put C:\Users\146261\Documents\IPN_ESCOM\Servidor2.java  
local:C:\Users\146261\Documents\IPN_ESCOM\Servidor2.java => remote:/home/mauro/Servidor2.java  
psftp> put C:\Users\146261\Documents\IPN_ESCOM\SimpleProxyServer.java  
local:C:\Users\146261\Documents\IPN_ESCOM\SimpleProxyServer.java => remote:/home/mauro/SimpleProxyServer.java  
psftp>
```

Comprobamos que los archivos hayan sido enviados correctamente:

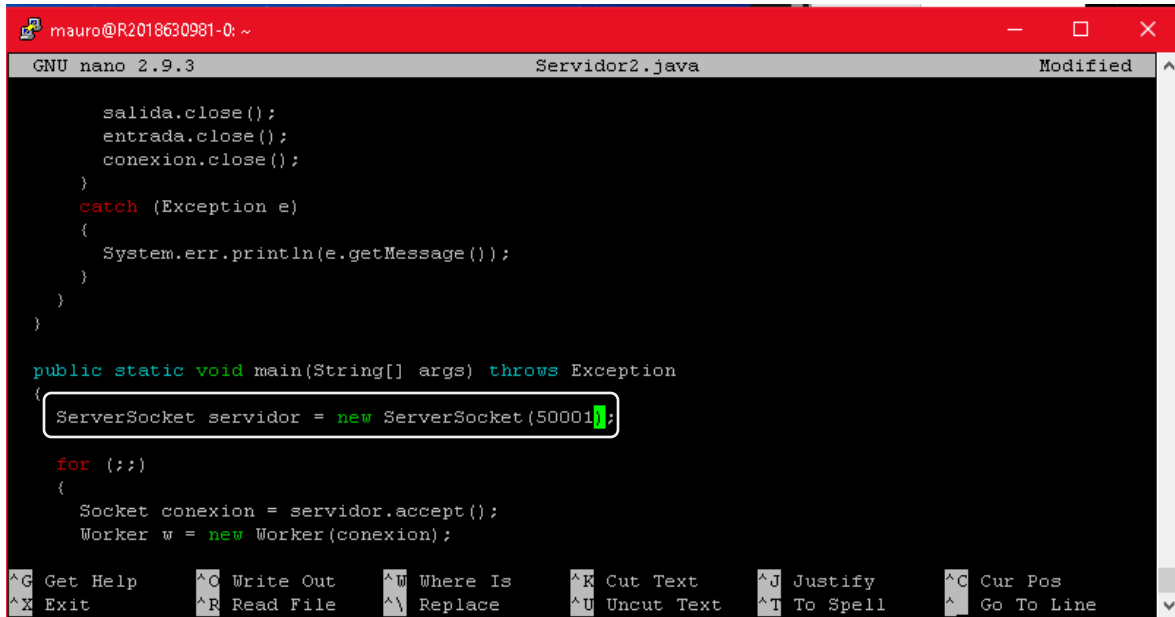


```
mauro@R2018630981-0: ~  
mauro@R2018630981-0:~$ ls  
Servidor2.java SimpleProxyServer.java  
mauro@R2018630981-0:~$
```



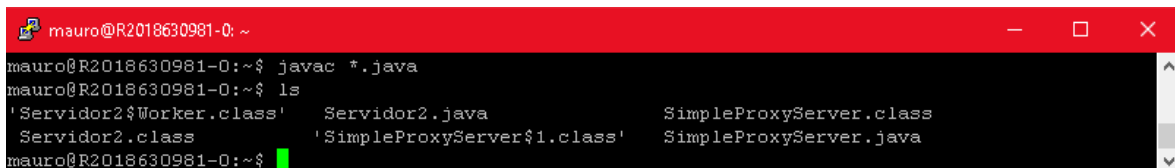
7. En la máquina virtual 1 editar el método "main" en el archivo [Servidor2.java](#):

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



```
mauro@R2018630981-0: ~  
GNU nano 2.9.3 Servidor2.java Modified  
  
        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);  
    }  
}
```

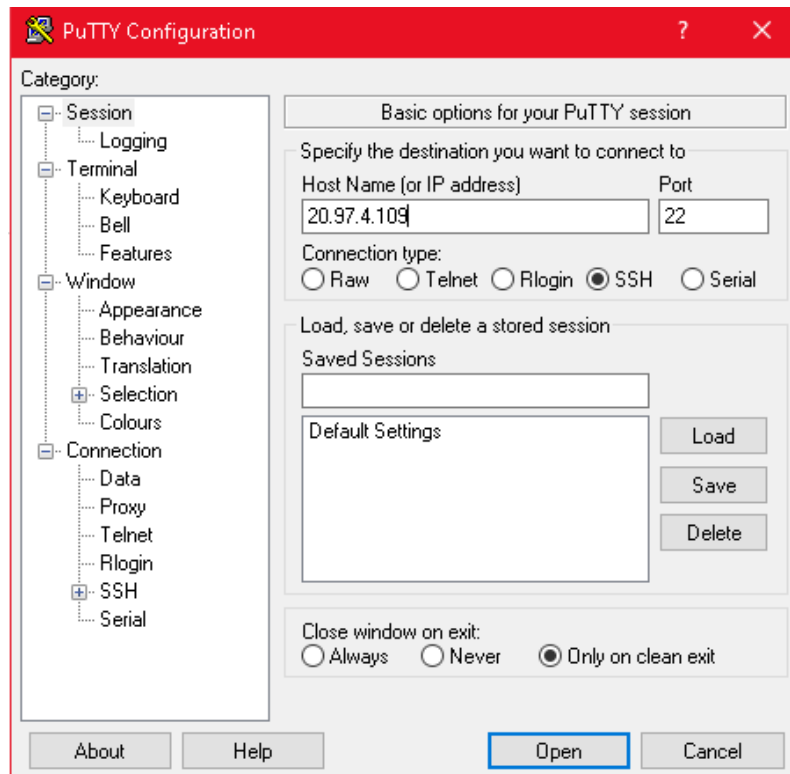
8. Compilar en la máquina virtual 1 los programas [Servidor2.java](#) y [SimpleProxyServer.java](#)



```
mauro@R2018630981-0: ~$ javac *.java  
mauro@R2018630981-0: ~$ ls  
'Servidor2$Worker.class'  Servidor2.java      SimpleProxyServer.class  
Servidor2.class          'SimpleProxyServer$1.class' SimpleProxyServer.java  
mauro@R2018630981-0: ~$
```

9. Conectar a la máquina virtual 2 (réplica) utilizando el programa **putty.exe**

Ejecutamos putty.exe y accedemos a la máquina virtual ingresando la IP pública en el programa y posteriormente el usuario y la contraseña de esta.



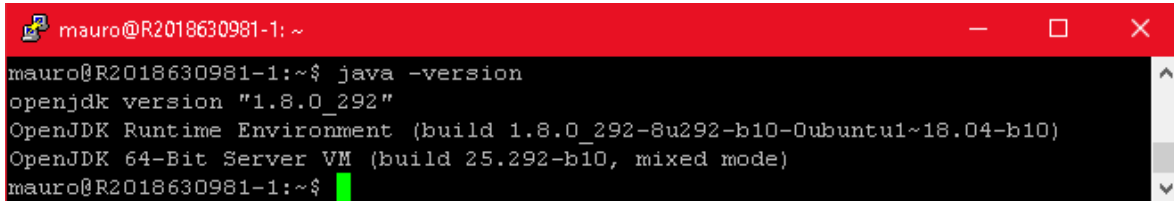
```
mauro@R2018630981-1: ~  
login as: mauro  
mauro@20.97.4.109's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1047-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Mon May 31 03:13:52 UTC 2021  
  
System load:  0.0          Processes:            108  
Usage of /:   4.5% of 28.90GB Users logged in:      0  
Memory usage: 20%         IP address for eth0: 10.0.0.5  
Swap usage:   0%  
  
* Super-optimized for small spaces - read how we shrank the memory  
  footprint of MicroK8s to make it the smallest full K8s around.  
  
https://ubuntu.com/blog/microk8s-memory-optimisation  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
mauro@R2018630981-1:~$
```

10. Instalar JDK-8 en la máquina virtual 2, ejecutando los comandos:

```
sudo apt update
```

```
sudo apt install openjdk-8-jdk-headless
```

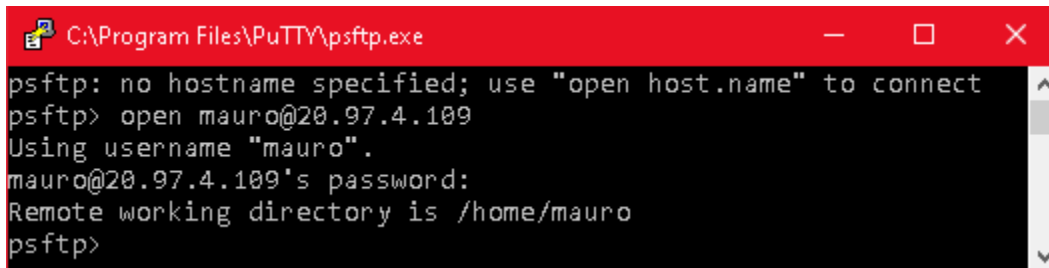
Comprobamos la instalación con el comando ***“java -version”***.



```
mauro@R2018630981-1: ~  
mauro@R2018630981-1:~$ java -version  
openjdk version "1.8.0_292"  
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10)  
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)  
mauro@R2018630981-1:~$
```

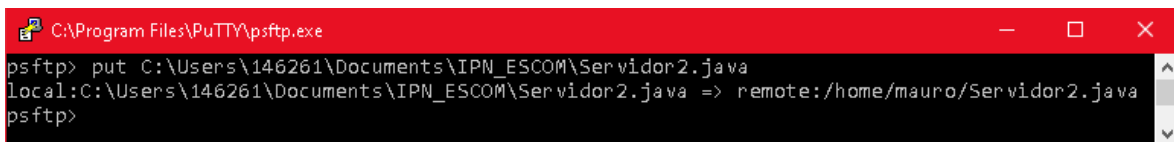
11. Utilizando el programa psftp.exe enviar a la máquina virtual 2 el archivo Servidor2.java

Ejecutamos el programa psftp.exe y ejecutamos el comando ***“open <open @direccionIPpublica>”*** con los datos del usuario e IP pública de la máquina virtual 2, seguido de la contraseña del usuario. De esta manera podremos trabajar en el directorio remoto de la máquina virtual.



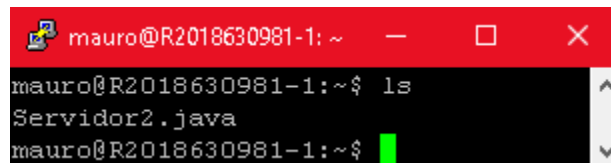
```
C:\Program Files\PuTTY\psftp.exe  
psftp: no hostname specified; use "open host.name" to connect  
psftp> open mauro@20.97.4.109  
Using username "mauro".  
mauro@20.97.4.109's password:  
Remote working directory is /home/mauro  
psftp>
```

Enviamos el archivo Servidor2.java y SimpleProxyServer.java con el comando ***“put <rutaArchivo>”***.



```
C:\Program Files\PuTTY\psftp.exe  
psftp> put C:\Users\146261\Documents\IPN_ESCOM\Servidor2.java  
local:C:\Users\146261\Documents\IPN_ESCOM\Servidor2.java => remote:/home/mauro/Servidor2.java  
psftp>
```

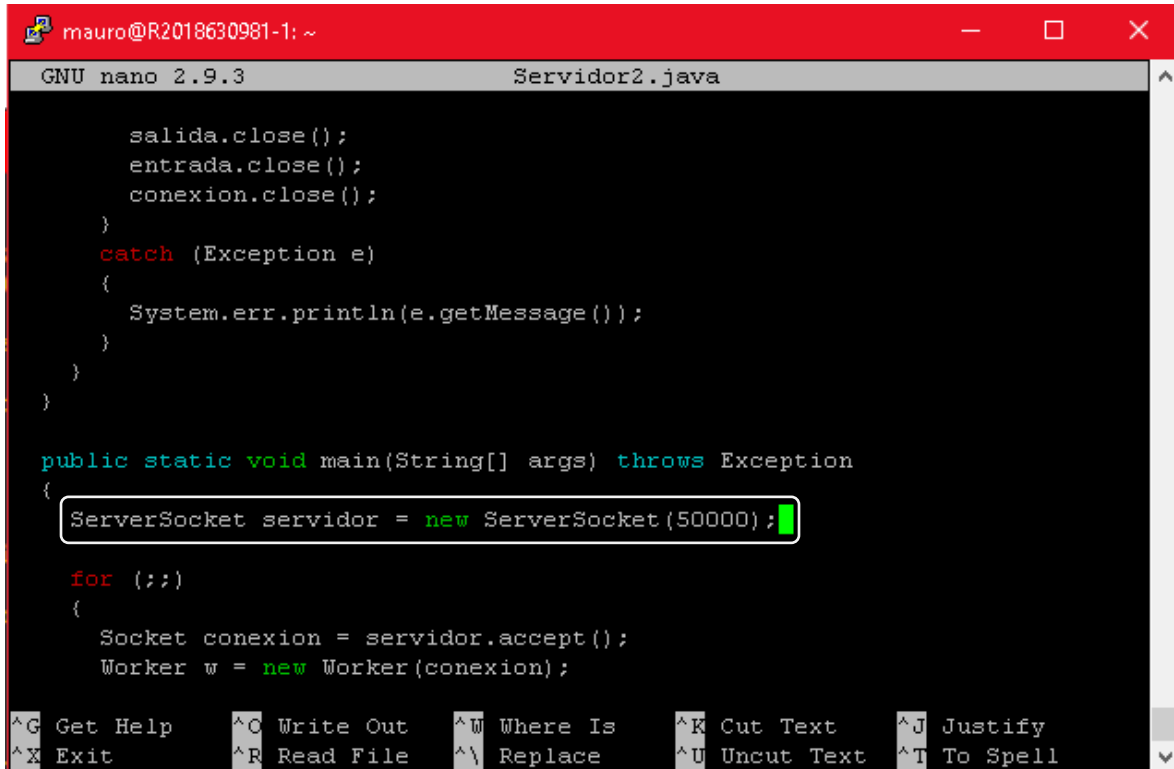
Comprobamos que los archivos hayan sido enviados correctamente:



```
mauro@R2018630981-1: ~  
mauro@R2018630981-1:~$ ls  
Servidor2.java  
mauro@R2018630981-1:~$
```

12. En la máquina virtual 2 editar el método "main" en el archivo [Servidor2.java](#):

`ServerSocket servidor = new ServerSocket(50000);`



The screenshot shows a terminal window titled "mauro@R2018630981-1: ~". The editor is GNU nano 2.9.3, editing the file Servidor2.java. The code visible is:

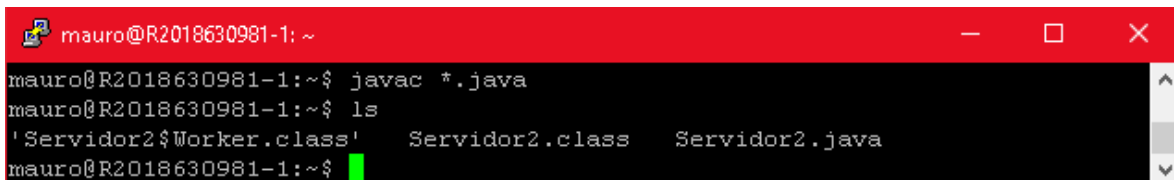
```
        salida.close();
        entrada.close();
        conexion.close();
    }
    catch (Exception e)
    {
        System.err.println(e.getMessage());
    }
}

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

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

The bottom of the window shows nano editor shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell.

13. Compilar el programa [Servidor2.java](#)

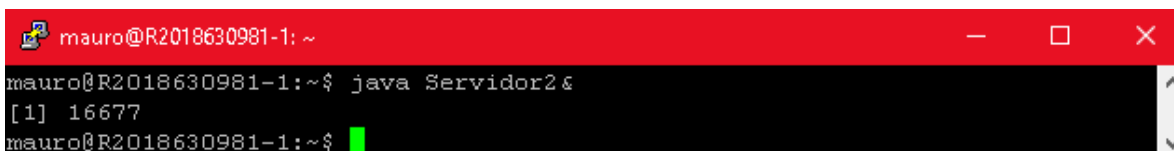


The screenshot shows a terminal window titled "mauro@R2018630981-1: ~". The user enters the following commands:

```
mauro@R2018630981-1:~$ javac *.java
mauro@R2018630981-1:~$ ls
'Servidor2$Worker.class'  Servidor2.class  Servidor2.java
mauro@R2018630981-1:~$
```

14. Ejecutar el programa [Servidor2.java](#) en la máquina virtual 2:

`java Servidor2&`



The screenshot shows a terminal window titled "mauro@R2018630981-1: ~". The user enters the command:

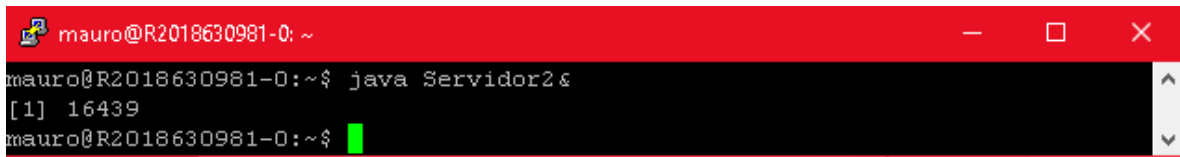
```
mauro@R2018630981-1:~$ java Servidor2&
```

The output shows:

```
[1] 16677
mauro@R2018630981-1:~$
```

## 15. Ejecutar el programa Servidor2.java en la máquina virtual 1:

java Servidor2&




```
mauro@R2018630981-0: ~  
mauro@R2018630981-0:~$ java Servidor2&  
[1] 16439  
mauro@R2018630981-0:~$
```

## 16. Ejecutar el máquina virtual 1 el proxy:

java SimpleProxyServer IP-máquina-virtual-2 50000 50000 50001&

(IP-máquina-virtual-2 es la IP de la réplica, 50000 es el puerto abierto en la réplica, 50000 es el puerto abierto en el sistema principal y 50001 es el puerto en la máquina virtual 1 donde el programa [Servidor2.java](#) recibe las peticiones. Notar que el puerto 50001 no se debe abrir en la máquina virtual 1, ya que el proxy y [Servidor2.java](#) se comunican mediante *loopback*).

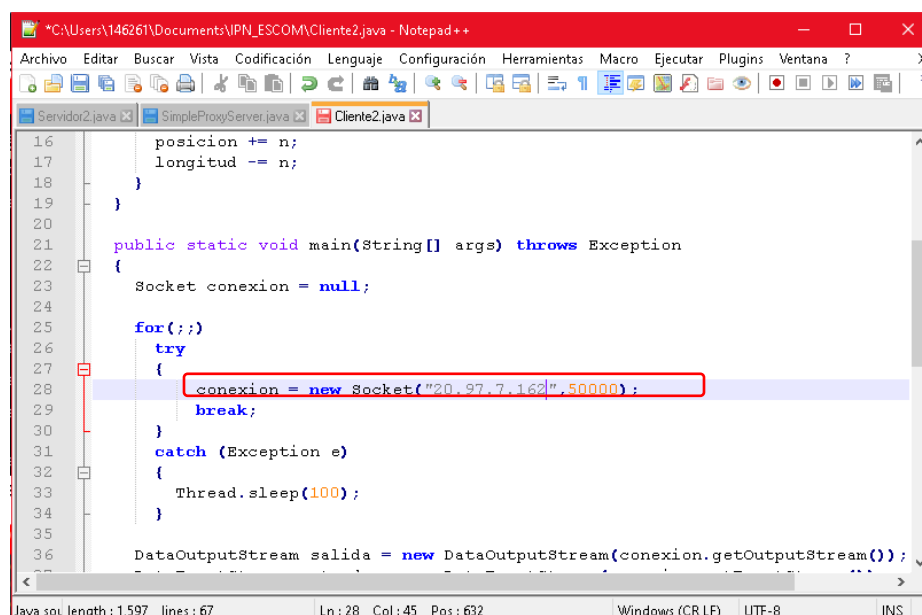


```
mauro@R2018630981-0: ~  
mauro@R2018630981-0:~$ java SimpleProxyServer 20.97.4.109 50000 50000 50001&  
[2] 16482  
mauro@R2018630981-0:~$ Iniciando SimpleProxyServer: 20.97.4.109:50000 on port  
50000
```

## 17. En Windows:

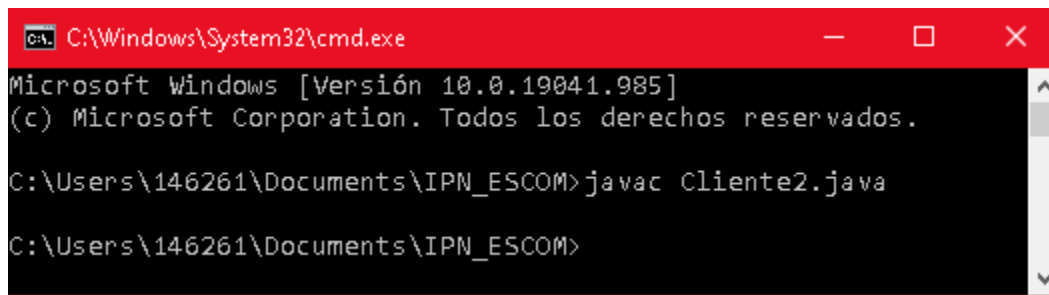
### 17.1 Editar el programa [Cliente2.java](#) para que se conecte a la máquina virtual 1.

Esto se hace editando el método "main" en el archivo Cliente2.java:



```
*C:\Users\146261\Documents\VPN_ESCOM\Cliente2.java - Notepad++  
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X  
Servidor2.java  SimpleProxyServer.java  Cliente2.java  
16      posicion += n;  
17      longitud -= n;  
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("20.97.7.162", 50000);  
30              break;  
31          }  
32          catch (Exception e)  
33          {  
34              Thread.sleep(100);  
35          }  
36      }  
37  
38      DataOutputStream salida = new DataOutputStream(conexion.getOutputStream());  
39  }  
40  }  
Java src. length: 1,597  lines: 67  Ln: 28  Col: 45  Pos: 632  Windows (CR LF)  UTF-8  INS
```

## 17.2 Compilar el programa [Cliente2.java](#)



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19041.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\146261\Documents\IPN_ESCOM>javac Cliente2.java

C:\Users\146261\Documents\IPN_ESCOM>
```

## 17.3 Ejecutar el programa [Cliente2.java](#)

- **Cliente**



```
C:\Windows\System32\cmd.exe
C:\Users\146261\Documents\IPN_ESCOM>java Cliente2
HOLA

C:\Users\146261\Documents\IPN_ESCOM>
```

- **Servidor en la Máquina Virtual 1**



```
mauro@R2018630981-0: ~
mauro@R2018630981-0:~$ 123
1.2345678901234567E9
hola
1.1
1.2
1.3
1.4
1.5
mauro@R2018630981-0:~$
```

- **Servidor en la Máquina Virtual 2**



```
mauro@R2018630981-1: ~
mauro@R2018630981-1:~$ 123
1.2345678901234567E9
hola
1.1
1.2
1.3
1.4
1.5
mauro@R2018630981-1:~$
```

**Conclusión:**

La replicación de un sistema es una herramienta que resulta bastante útil al brindarnos una mejora en el rendimiento de los sistemas distribuidos y una mejora en la consistencia de los datos; así como también la posibilidad de poder realizar la recuperación de los datos y archivos de nuestro sistema en caso de algún evento catastrófico.