

Instituto Politécnico Nacional  
Escuela Superior de Cómputo

# Chat multicast

## Tarea 4

Alumno: Francisco Nicolas Sánchez García  
Asignatura Desarrollo de Sistemas Distribuidos  
Profesor: Carlos Pineda Guerrero  
Grupo 4CV12

---

# ÍNDICE

Introducción .....	1
Desarrollo .....	1
Conclusiones .....	10

## INTRODUCCIÓN

Una de las principales ventajas de un chat multicast es la descentralización de un servidor que sea el encargado de enviar y recibir mensajes para posteriormente distribuirlos con los demás clientes. De este modo, cualquiera en el chat multicast puede recibir y enviar mensajes gracias al uso de la dirección IP multicast local.

A continuación, se muestra el código de como se completó el programa del chat usando multicast, se muestra la creación de la máquina virtual con Windows Server 2012, la compilación del programa dentro de la misma y posteriormente la ejecución de tres ventanas (hugo, paco y luis) usando la conversación solicitada en la tarea, finalmente se dan conclusiones de la misma.

## DESARROLLO

Se presenta el código del chat con el código faltante para poder funcionar.

### Código fuente de Chat.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.MulticastSocket;
import java.net.NetworkInterface;
import java.nio.charset.StandardCharsets;
import java.util.Scanner;

//chcp 1252
public class Chat {
    static class Worker extends Thread {
        Worker() {
        }

        public void run() {
            System.setProperty("java.net.preferIPv4Stack", "true");
            try {
                MulticastSocket socket = new MulticastSocket(20000);
                InetSocketAddress grupo = new
                InetSocketAddress(InetAddress.getByAddress("230.0.0.0"), 20000);
                NetworkInterface netInter = NetworkInterface.getByAddress("em1");// em0, bge0, bge1
                socket.joinGroup(grupo, netInter);
                while (true) {
                    byte[] a = recibe_mensaje_multicast(socket, 48);
                    System.out.println(new String(a, StandardCharsets.UTF_8));
                }
            } catch (Exception e) {
                System.out.println("Error: " + e.getMessage());
            }
            // socket.leaveGroup(grupo, netInter);
            // socket.close();
        }
    }
}
```

```

    }

    public static void main(String[] args) throws Exception {

        new Worker().start();
        String nombre = args[0];

        String mensaje = "";
        Scanner sc = new Scanner(System.in);

        // En un ciclo infinito se leerá cada mensaje del teclado y se enviará el
        // mensaje al grupo 230.0.0.0 a través del puerto 20000.
        System.setProperty("java.net.preferIPv4Stack", "true");

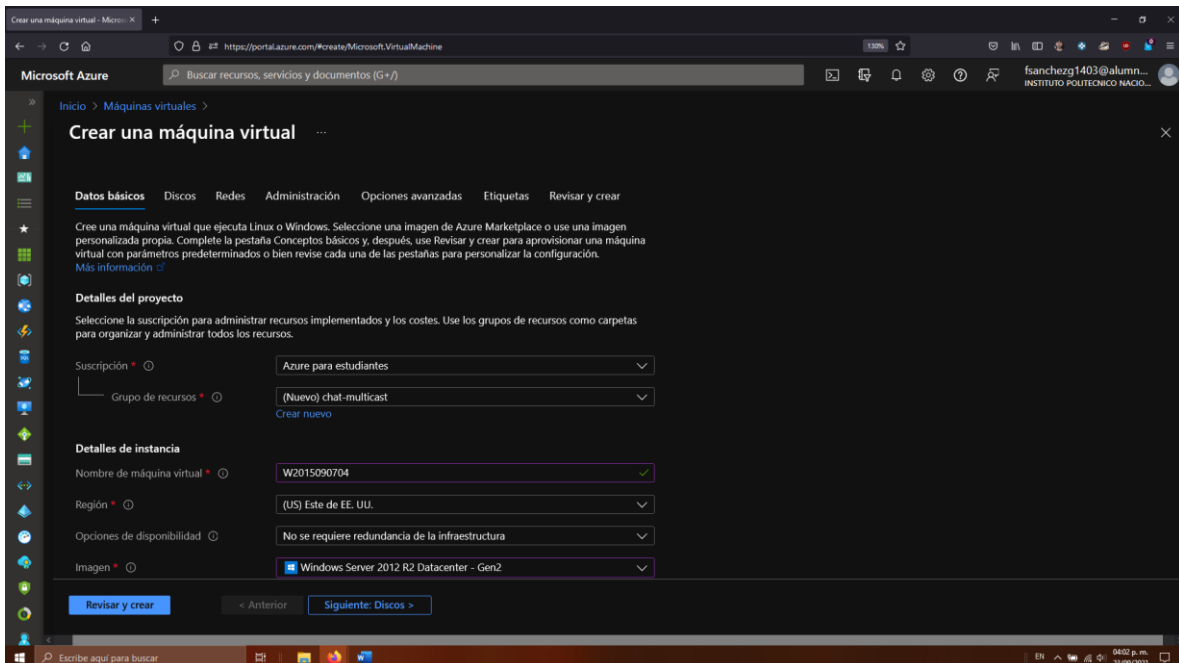
        while (true) {
            System.out.println("Ingrese el mensaje a enviar: ");
            mensaje = sc.nextLine();
            envia_mensaje_multicast((nombre + ":" + mensaje).getBytes(StandardCharsets.UTF_8),
"230.0.0.0", 20000);
        }

        static void envia_mensaje_multicast(byte[] buffer, String ip, int puerto) throws IOException {
            DatagramSocket socket = new DatagramSocket();
            socket.send(new DatagramPacket(buffer, buffer.length, InetAddress.getByName(ip), puerto));
            socket.close();
        }

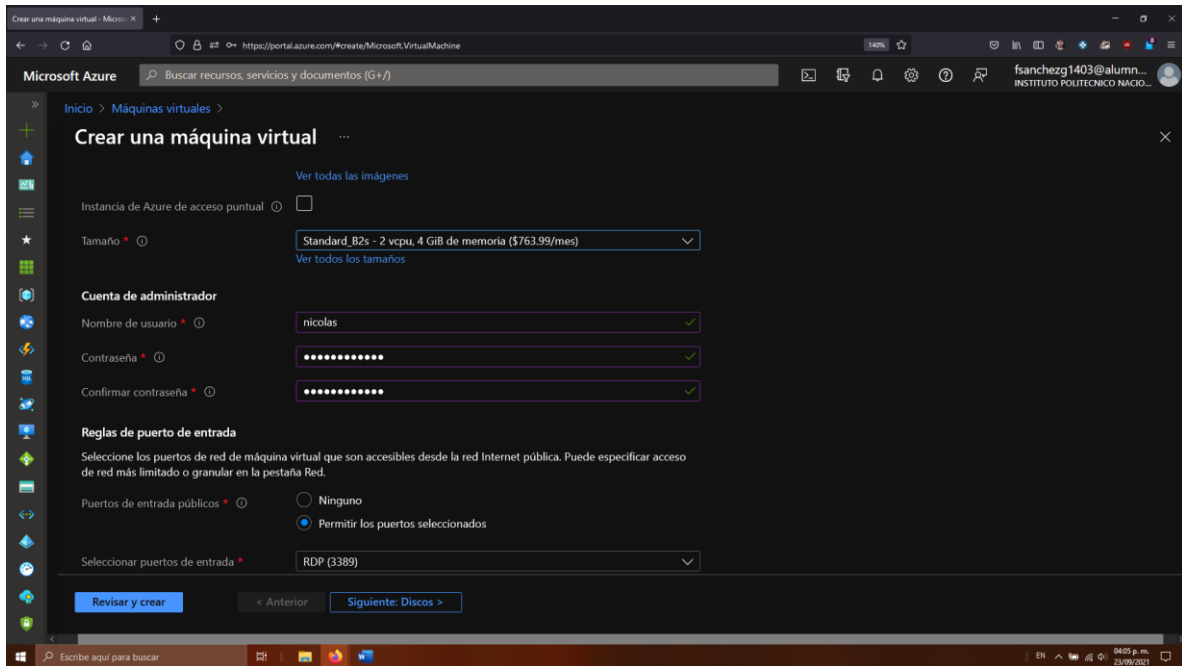
        static byte[] recibe_mensaje_multicast(MulticastSocket socket, int longitud_mensaje) throws
IOException {
            byte[] buffer = new byte[longitud_mensaje];
            DatagramPacket paquete = new DatagramPacket(buffer, buffer.length);
            socket.receive(paquete);
            return paquete.getData();
        }
    }
}

```

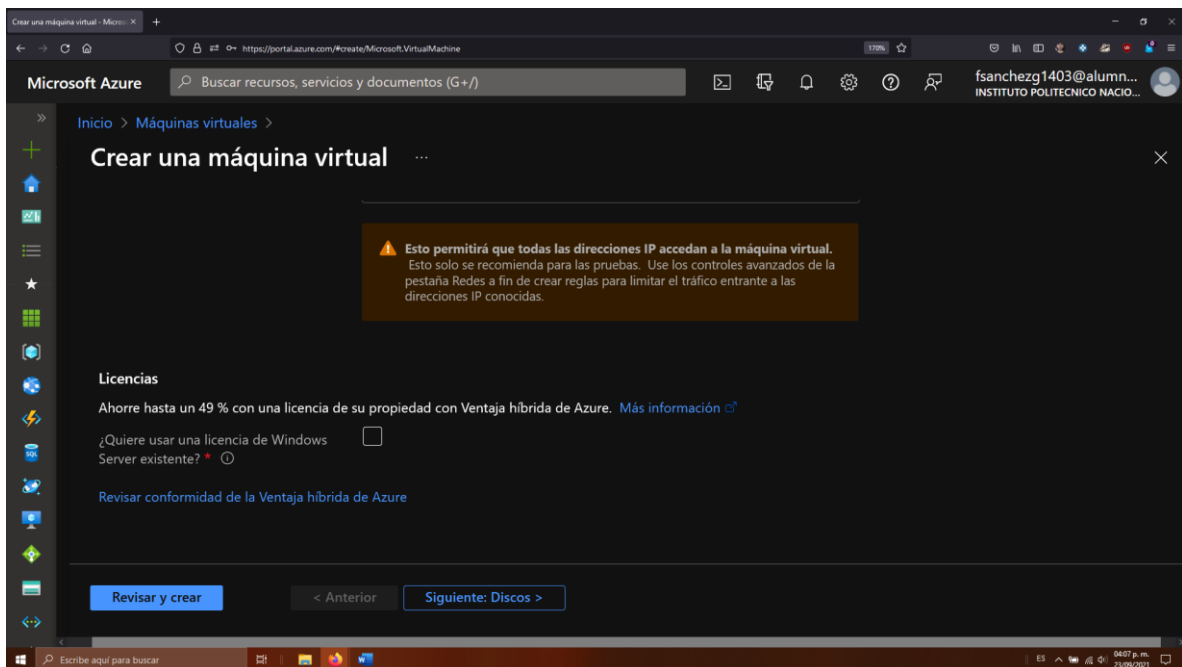
## Creación de máquina virtual



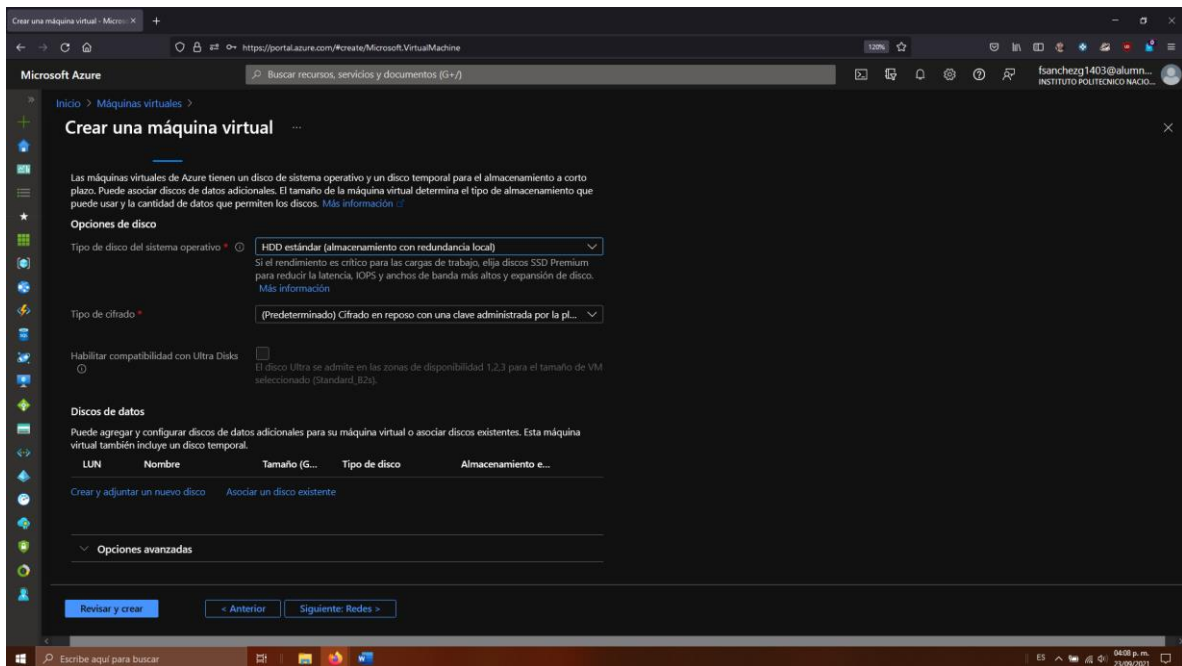
Para crear la máquina virtual se crea un nuevo grupo de recursos para la VM llamado "chat-multicast", se agrega el nombre de la máquina siguiendo la nomenclatura de la tarea (W2015090704). Se selecciona la imagen de Windows Server 2012.



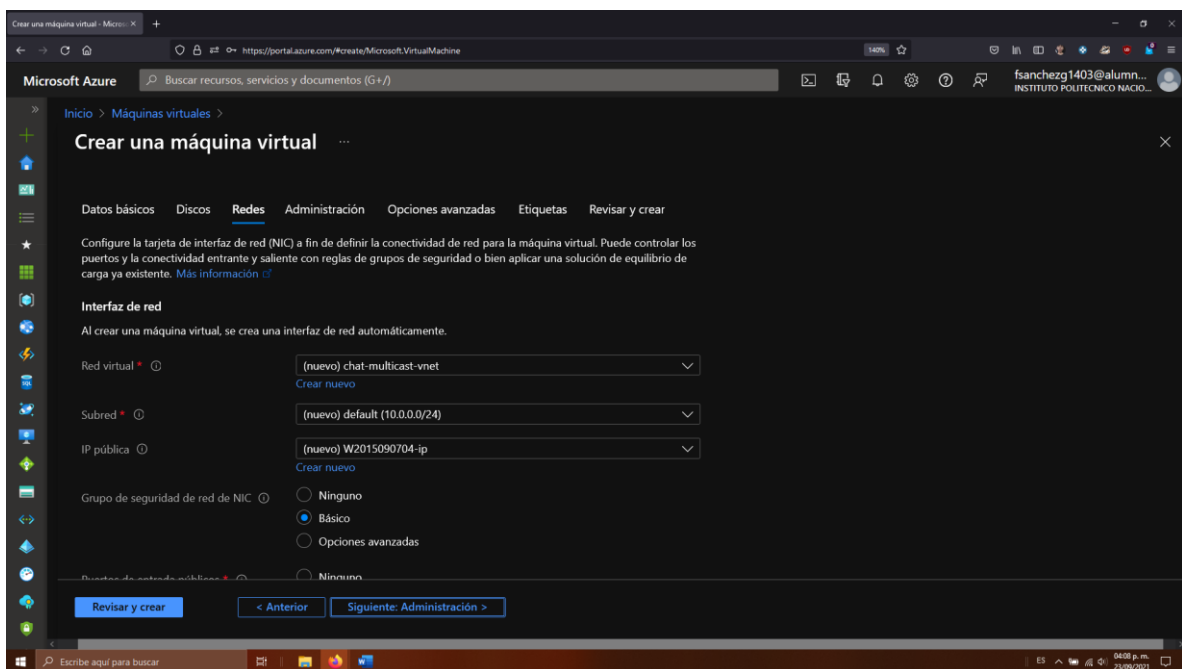
Se selecciona el tamaño B2s ya que tiene 4GB de memoria RAM. Se agrega usuario y contraseña y se deja el puerto de entrada como RDP.



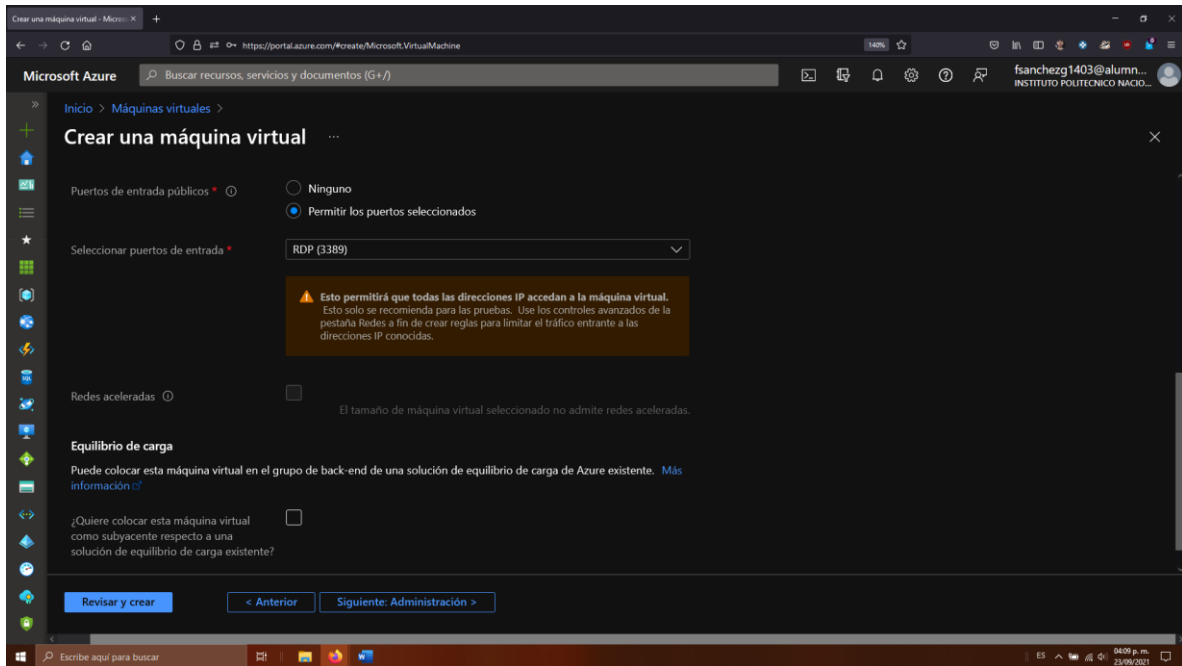
Finalmente, no se selecciona la casilla de licencias ya que no se cuenta con ninguna. Pasamos al apartado de "Discos".



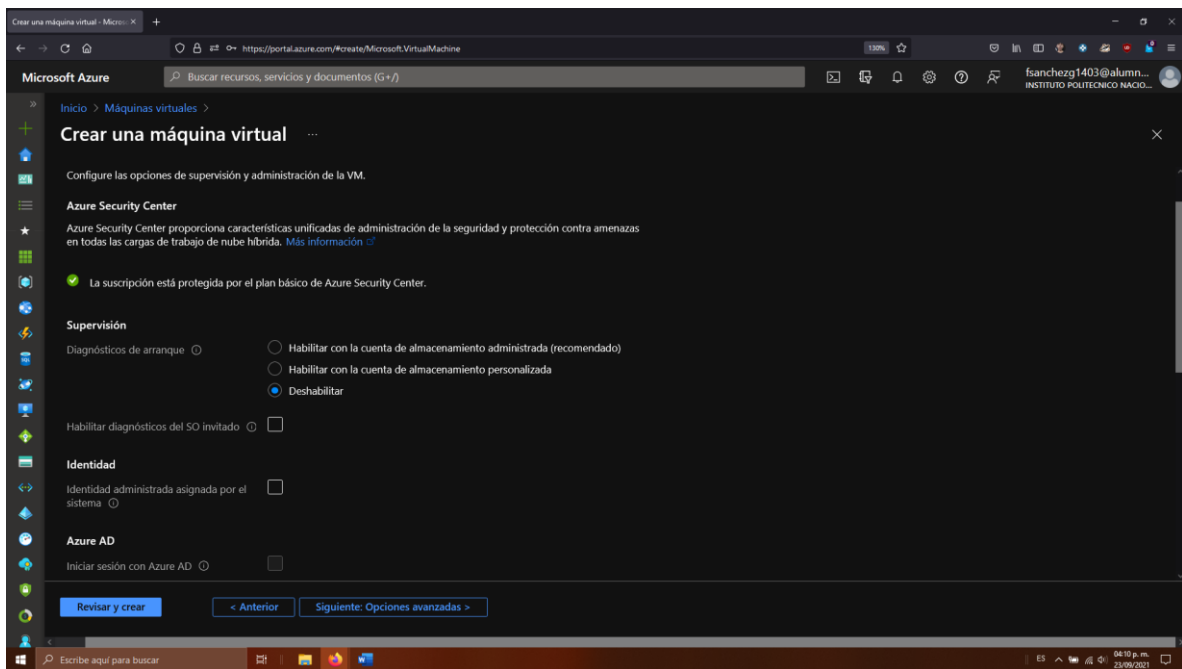
Cambiamos el tipo de disco del sistema operativo a un “HDD estándar”. No se modifica nada de “Opciones avanzadas” y pasamos al apartado de “Redes”.



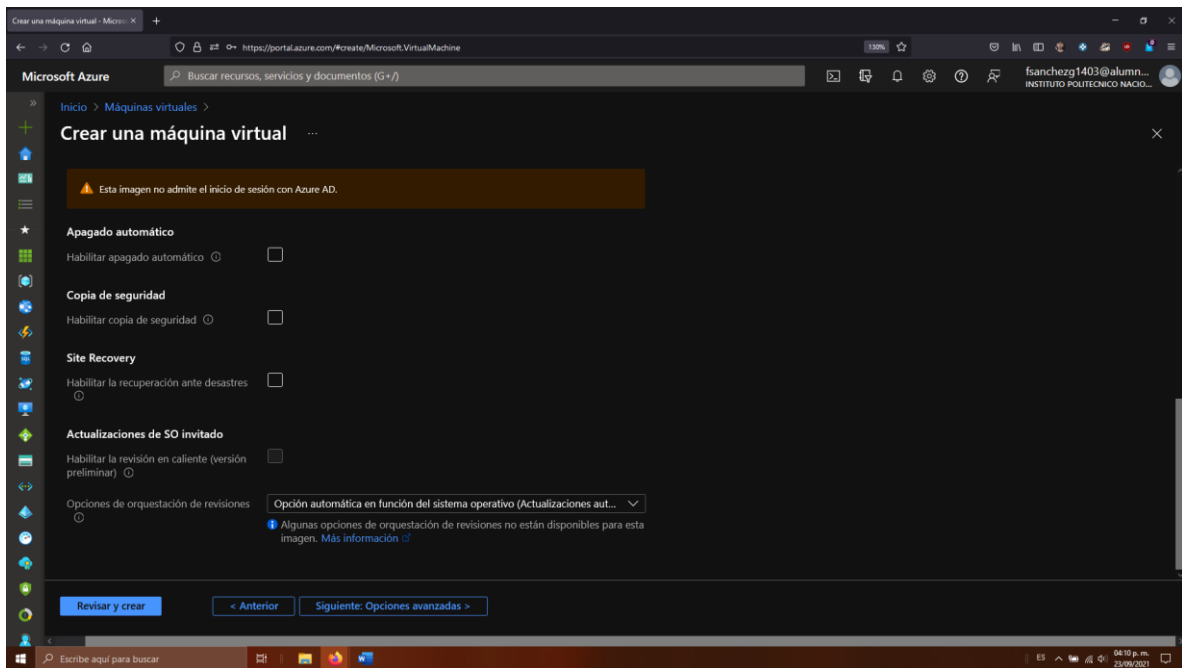
Dejamos los valores generados por defecto en el apartado de “Redes”.



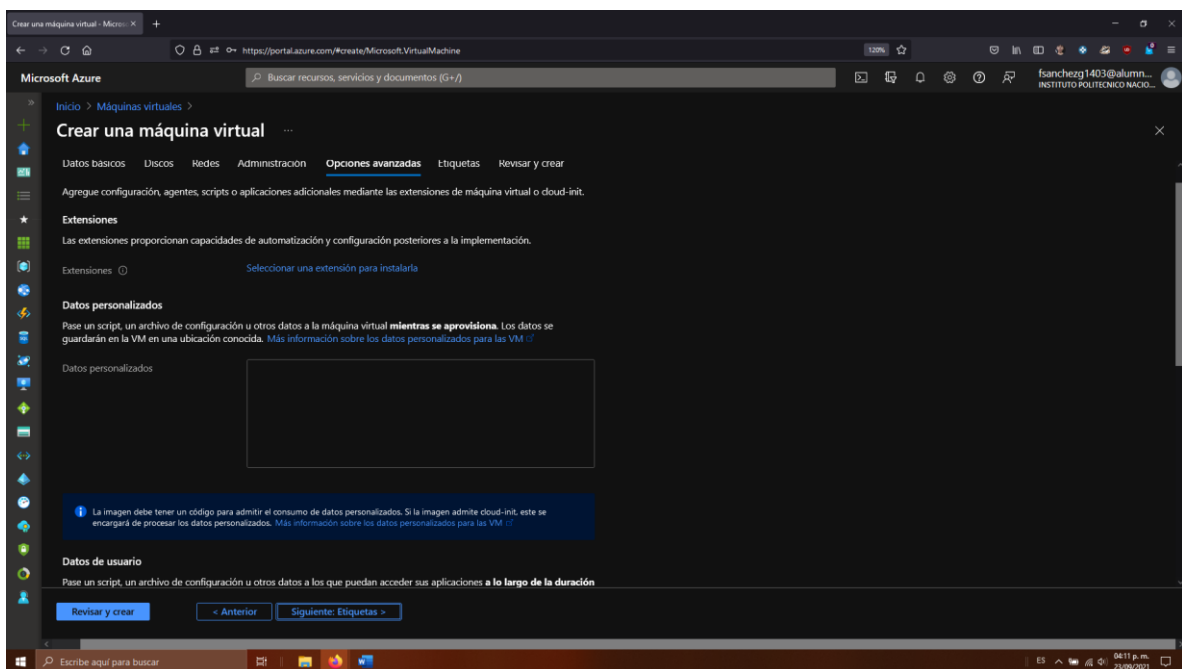
Pasamos al apartado de “Administración”.



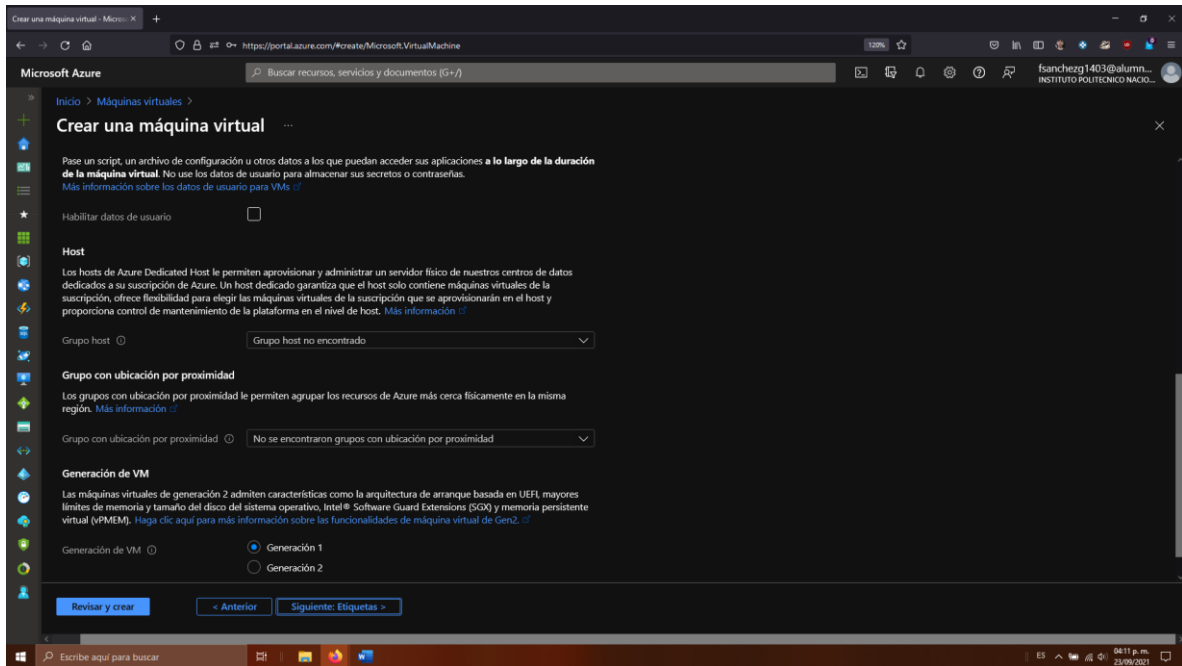
Se deshabilita el diagnóstico de arranque de la máquina virtual.



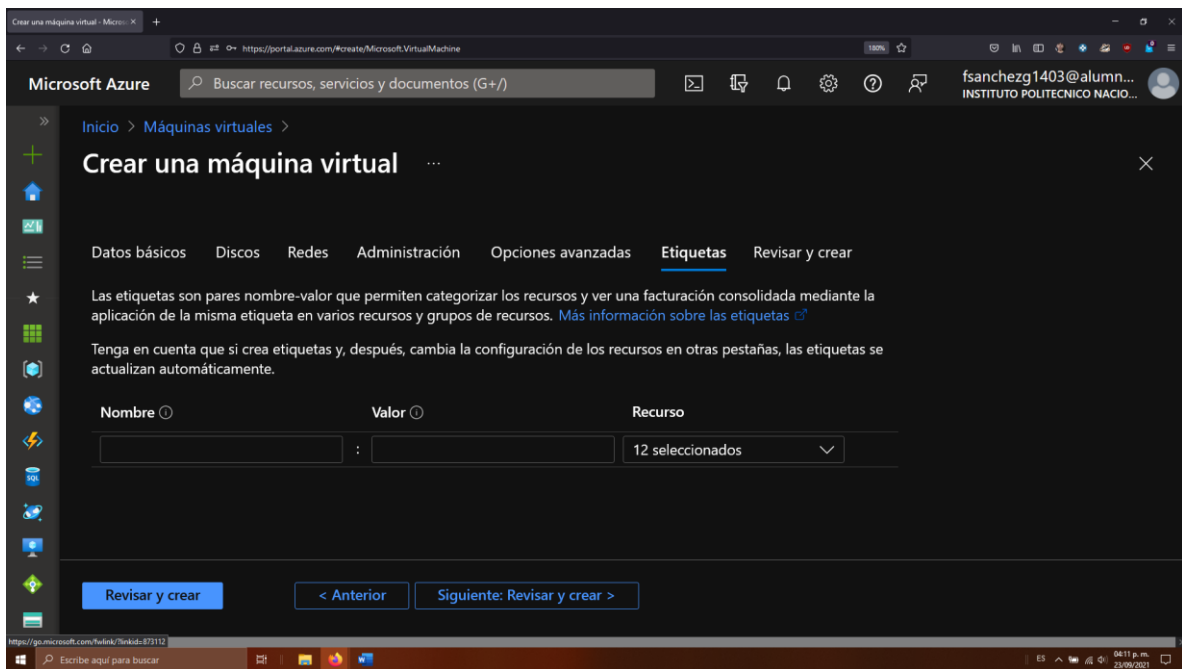
No se modifica ninguna otra opción y pasamos al apartado de “Opciones avanzadas”.



No se modifica nada del apartado de “Opciones avanzadas”.

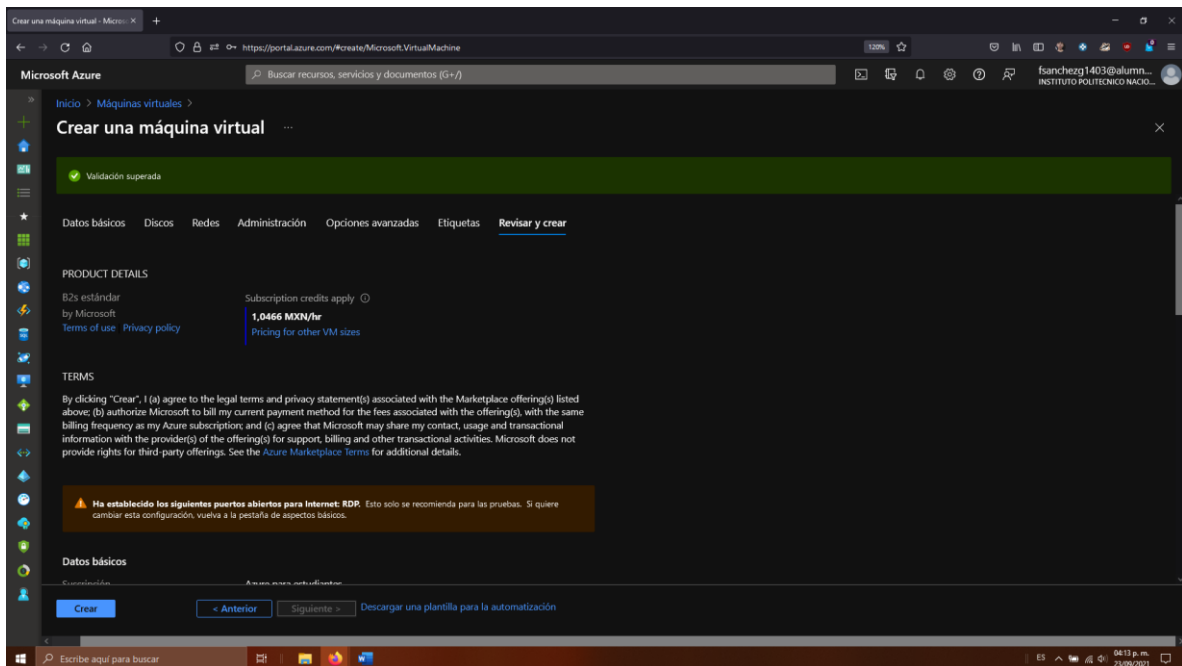


Pasamos al apartado de “Etiquetas”.

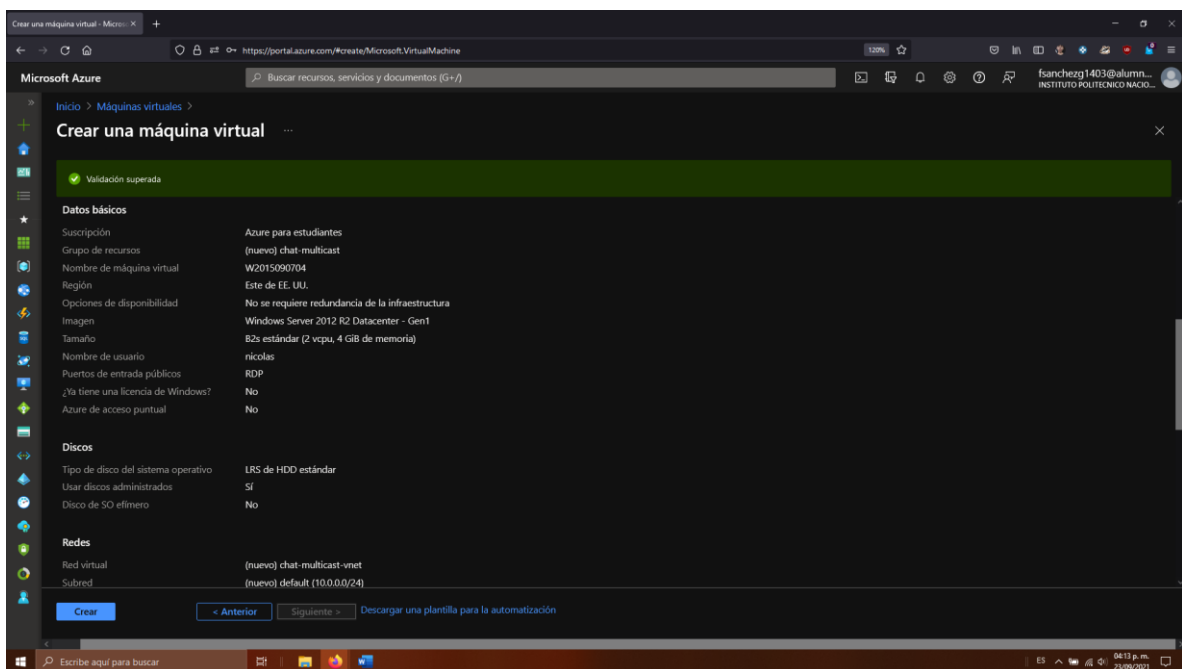


No se agrega ninguna etiqueta. Pasamos al apartado de “Revisar y crear”.

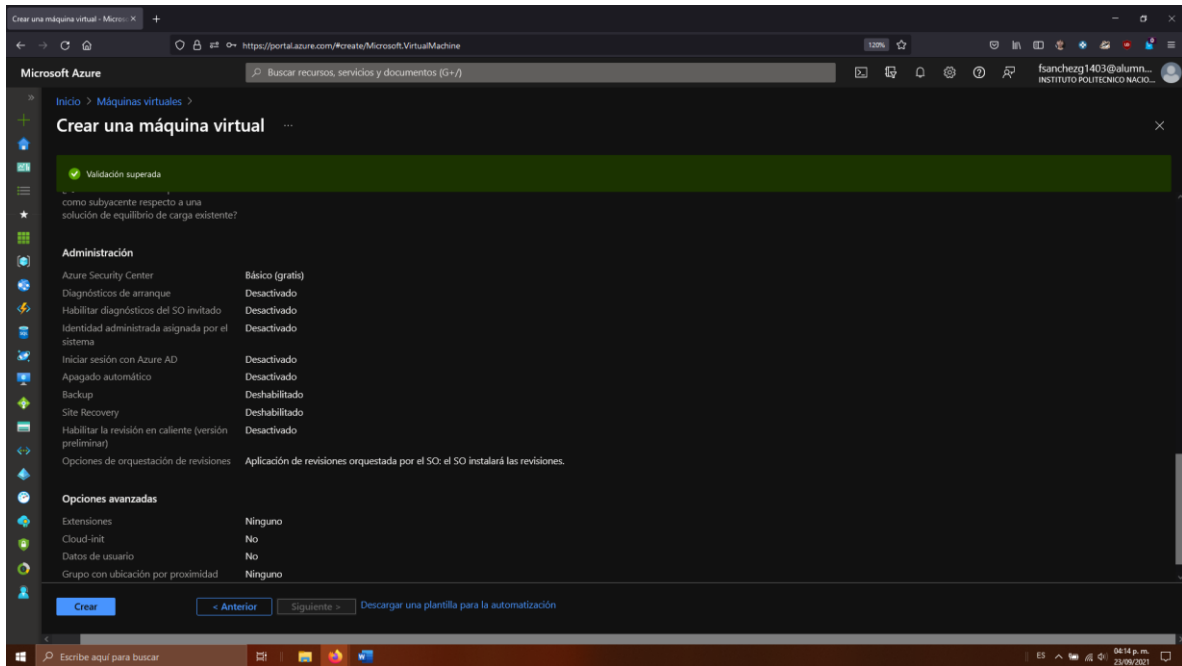




Una vez que la validación ha sido superada, confirmamos los datos anteriormente ingresados.

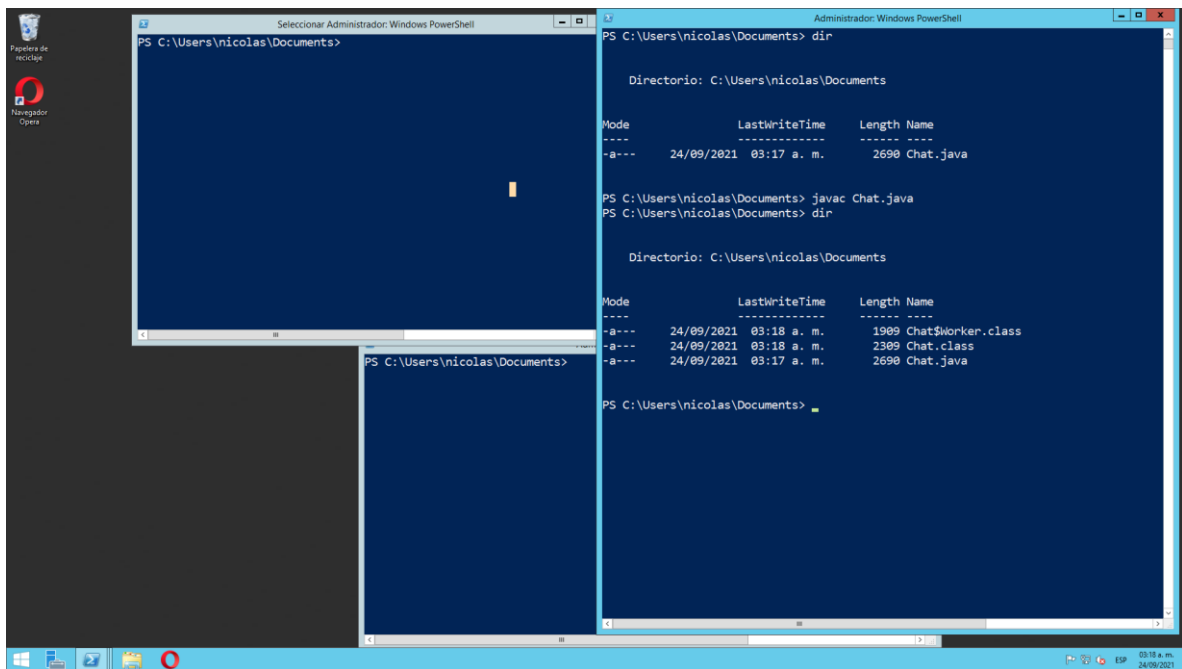


Todos los datos de la máquina virtual son los que ingresamos anteriormente.



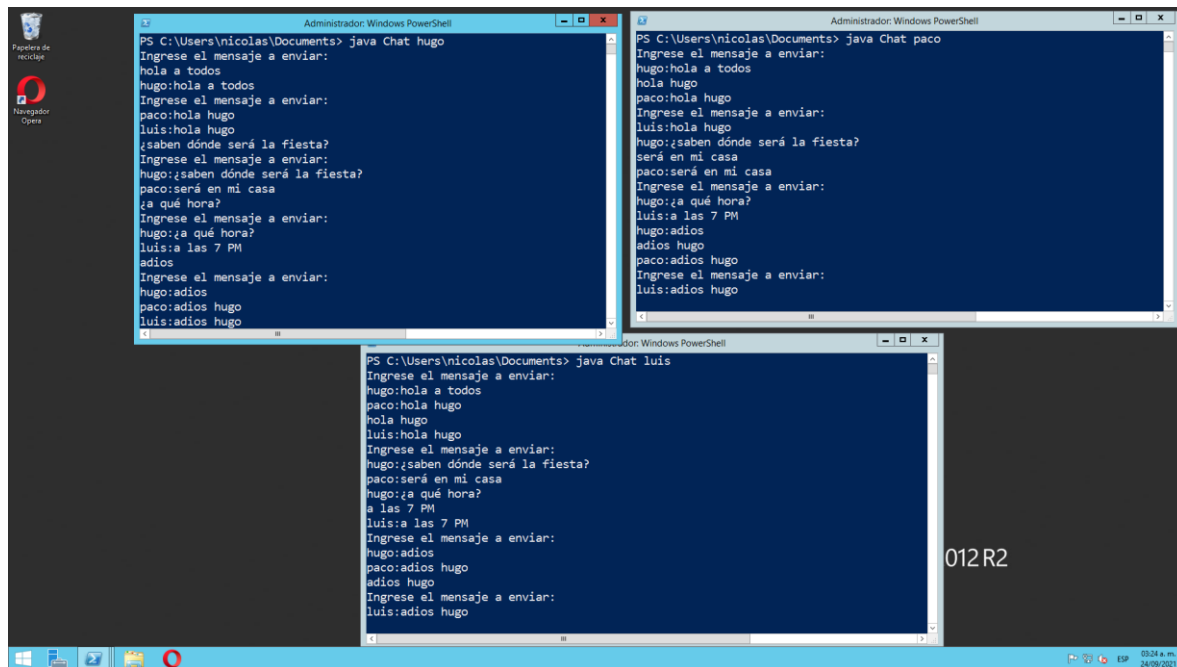
Una vez revisado todos los campos, damos clic en “Crear”.

Compilación del programa



Compilación del programa en la máquina virtual.

## Ejecución del programa



```
PS C:\Users\nicolas\Documents> java Chat hugo
Ingrese el mensaje a enviar:
hola a todos
hugo:hola a todos
Ingrese el mensaje a enviar:
paco:hola hugo
luis:hola hugo
¿saben dónde será la fiesta?
Ingrese el mensaje a enviar:
hugo:¿saben dónde será la fiesta?
paco:será en mi casa
¿a qué hora?
Ingrese el mensaje a enviar:
hugo:¿a qué hora?
luis:a las 7 PM
adios
Ingrese el mensaje a enviar:
hugo:adios
paco:adios hugo
luis:adios hugo

PS C:\Users\nicolas\Documents> java Chat paco
Ingrese el mensaje a enviar:
hugo:hola a todos
hola hugo
paco:hola hugo
Ingrese el mensaje a enviar:
luis:hola hugo
hugo:¿saben dónde será la fiesta?
será en mi casa
paco:será en mi casa
Ingrese el mensaje a enviar:
hugo:¿a qué hora?
luis:a las 7 PM
hugo:adios
adios hugo
paco:adios hugo
Ingrese el mensaje a enviar:
luis:adios hugo

PS C:\Users\nicolas\Documents> java Chat luis
Ingrese el mensaje a enviar:
hugo:hola a todos
paco:hola hugo
hola hugo
luis:hola hugo
Ingrese el mensaje a enviar:
hugo:¿saben dónde será la fiesta?
paco:será en mi casa
hugo:¿a qué hora?
a las 7 PM
luis:a las 7 PM
Ingrese el mensaje a enviar:
hugo:adios
paco:adios hugo
adios hugo
Ingrese el mensaje a enviar:
luis:adios hugo
```

Se realiza el envío de mensajes como se muestra en la tarea, hay que tener en cuenta que el mensaje “Ingrese el mensaje a enviar: ” se desacomoda debido a que nuestro Thread encargado de recibir mensajes tal cual recibe uno lo imprime en consola.

## CONCLUSIONES

Dentro de los usos del multicast se destaca su implementación en chats en redes locales, como podemos ver en la ejecución, es muy fácil realizar una comunicación de este tipo. Encontré que el uso de Windows Server 2012 es algo obsoleto debido al tiempo. El problema principal en esta tarea fue encontrar la página de codificación que permitiera mostrar caracteres ajenos a ASCII, en mi caso usé la página 1252 (Latin 1).