



# Instituto Politécnico Nacional Escuela Superior de Cómputo

**Tarea 5. Chat multicast**

PRESENTA  
**Vladimir Azpeitia Hernández**

PROFESOR  
**Carlos Pineda Guerrero**

ASGINATURA  
**Desarrollo de sistemas distribuidos**

20 de abril de 2021

# Tarea 4. Implementación de un token-ring

Vladimir Azpeitia Hernández

20 de abril de 2021

## 1. Introducción

Desarrollar un programa en Java que implemente un chat utilizando comunicación multicast mediante datagramas.

- Se deberá ejecutar el programa en una máquina virtual con Windows 10 en Azure. Solo se admitirá la tarea si se trata de un programa en modo consola de caracteres (no se admitirá el programa en modo gráfico).
- Se deberá pasar como parámetro al programa el nombre del usuario que va escribir en el chat. Para demostrar el programa se deberá utilizar los siguientes usuarios: hugo, paco y luis (no usar otros usuarios).

El funcionamiento general del programa es el siguiente:

- El programa creará un thread que actuará como cliente multicast, el cual recibirá los mensajes del resto de los nodos. Cada mensaje recibido será desplegado en la pantalla. El thread desplegará el mensaje que envía el mismo nodo.
- En el método main(), dentro de un ciclo infinito se desplegará el siguiente prompt: 'Ingrese el mensaje a enviar: ' (sin las comillas), entonces se leerá una string (el mensaje). Se deberá enviar el mensaje a los nodos que pertenecen al grupo identificado por la IP 230.0.0.0 a través del puerto 50000.
- El paquete a enviar deberá tener la siguiente forma: nombre\_usuario:mensaje\_ingresado, dónde nombre\_usuario es el nombre del usuario que pasó como parámetro al programa (hugo, paco o luis) y mensaje\_ingresado el mensaje que el usuario ingresó por el teclado.

## 2. Desarrollo

### 2.1. Código fuente del programa

La implementación de este chat, solo consistió en el uso de las funciones que se describieron en la especificación de la tarea, además de la creación de un servidor y un cliente multicast, estos también ya se habían revisado en una clase previa. Solo tuve que hacer un cambio para que la máquina virtual reconociera los caracteres especiales, y fue el de sustituir 'UTF-8' por 'Windows-1252'.

```
1 import java.io.IOException;
2 import java.net.DatagramPacket;
3 import java.net.DatagramSocket;
4 import java.net.InetAddress;
5 import java.net.MulticastSocket;
6 import java.util.Scanner;
7 import java.util.Locale;
8
9 class Chat {
10
11     public static final int BUFSIZ = 50;
12
13     static void envia_mensaje_multicast(byte[] buffer, String ip, int puerto) throws IOException {
14         DatagramSocket socket = new DatagramSocket();
15         socket.send(new DatagramPacket(buffer, buffer.length, InetAddress.getByName(ip), puerto));
16         socket.close();
17     }
18
19     static byte[] recibe_mensaje_multicast(MulticastSocket socket, int longitud_mensaje) throws
IOException {
```

```

20 byte[] buffer = new byte[longitud_mensaje];
21 DatagramPacket paquete = new DatagramPacket(buffer, buffer.length);
22 socket.receive(paquete);
23 return paquete.getData();
24 }
25
26 static class Worker extends Thread {
27     public void run() {
28         // En un ciclo infinito se recibiran los mensajes enviados al grupo
29         // 230.0.0.0 a traves del puerto 50000 y se desplegarn en la pantalla.
30         while(true) {
31             try{
32                 InetAddress grupo = InetAddress.getByName("230.0.0.0");
33                 MulticastSocket socket = new MulticastSocket(50000);
34                 socket.joinGroup(grupo);
35                 byte[] buffer = recibe_mensaje_multicast(socket, BUFSIZ);
36                 System.out.printf("\n%", new String(buffer, "Windows-1252"));
37                 System.out.printf("\nIngrese el mensaje a enviar:");
38                 socket.leaveGroup(grupo);
39                 socket.close();
40             } catch (IOException e){
41                 e.printStackTrace();
42             }
43         }
44     }
45 }
46
47 public static void main(String[] args) throws Exception {
48     Worker w = new Worker();
49     w.start();
50     String nombre = args[0];
51     Scanner scanner = new Scanner(System.in);
52
53     // En un ciclo infinito se leera cada mensaje del teclado y se enviara el
54     // mensaje al grupo 230.0.0.0 a traves del puerto 50000.
55     System.out.print("Ingrese el mensaje a enviar:");
56     while(true) {
57         String mensaje = scanner.nextLine();
58         byte buffer[] = String.format("%s", nombre, mensaje).getBytes();
59         envia_mensaje_multicast(buffer, "230.0.0.0",50000);
60     }
61 }
62
63 }

```

---

## 2.2. Instalación y configuración de las máquinas virtuales

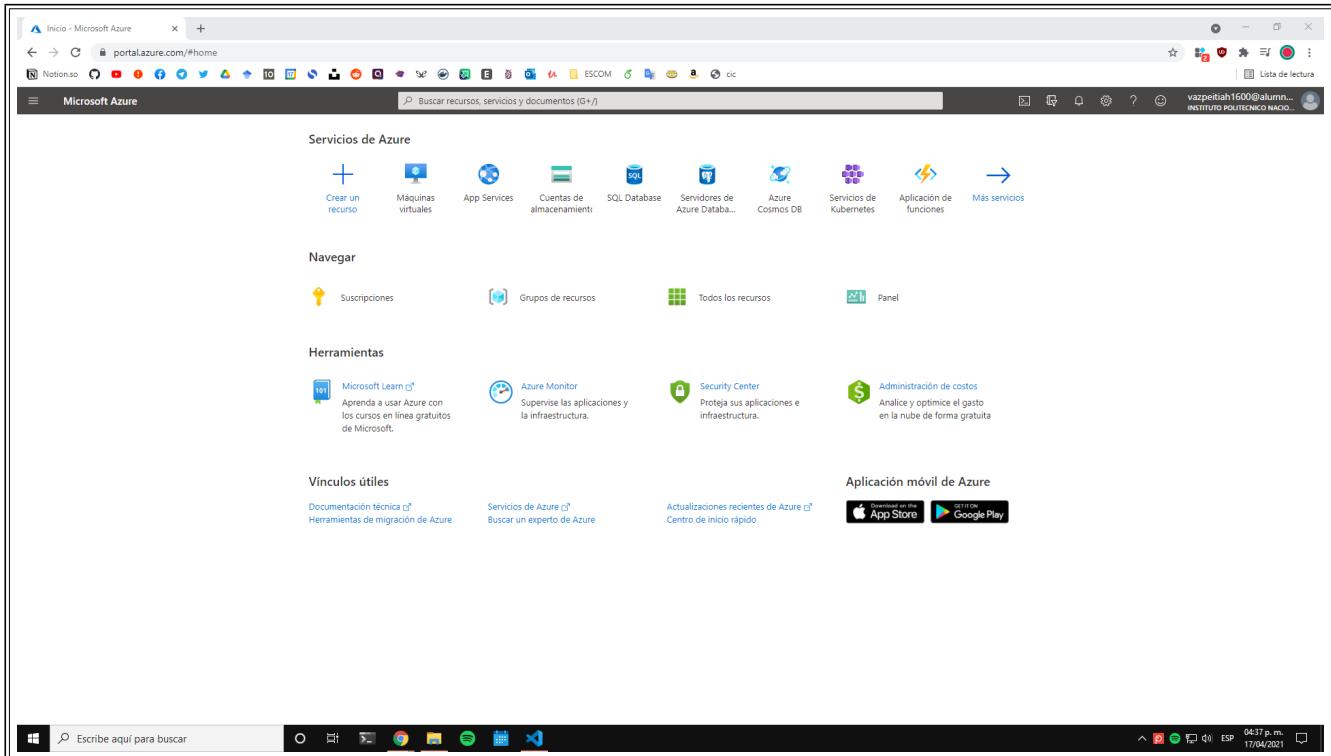


Figura 1: Creación de la máquina virtual de Windows 10 Pro: Portal de Azure

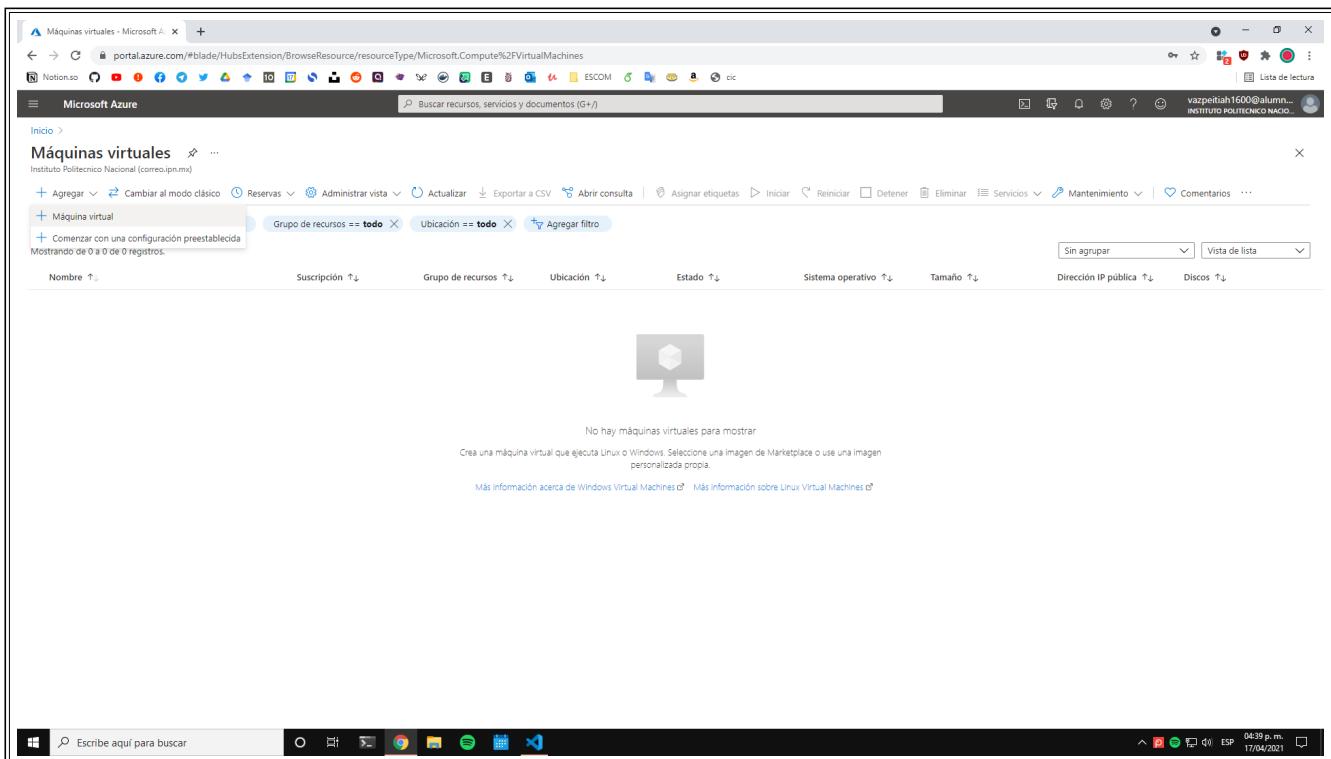


Figura 2: Creación de la máquina virtual de Windows 10 Pro: Máquinas virtuales

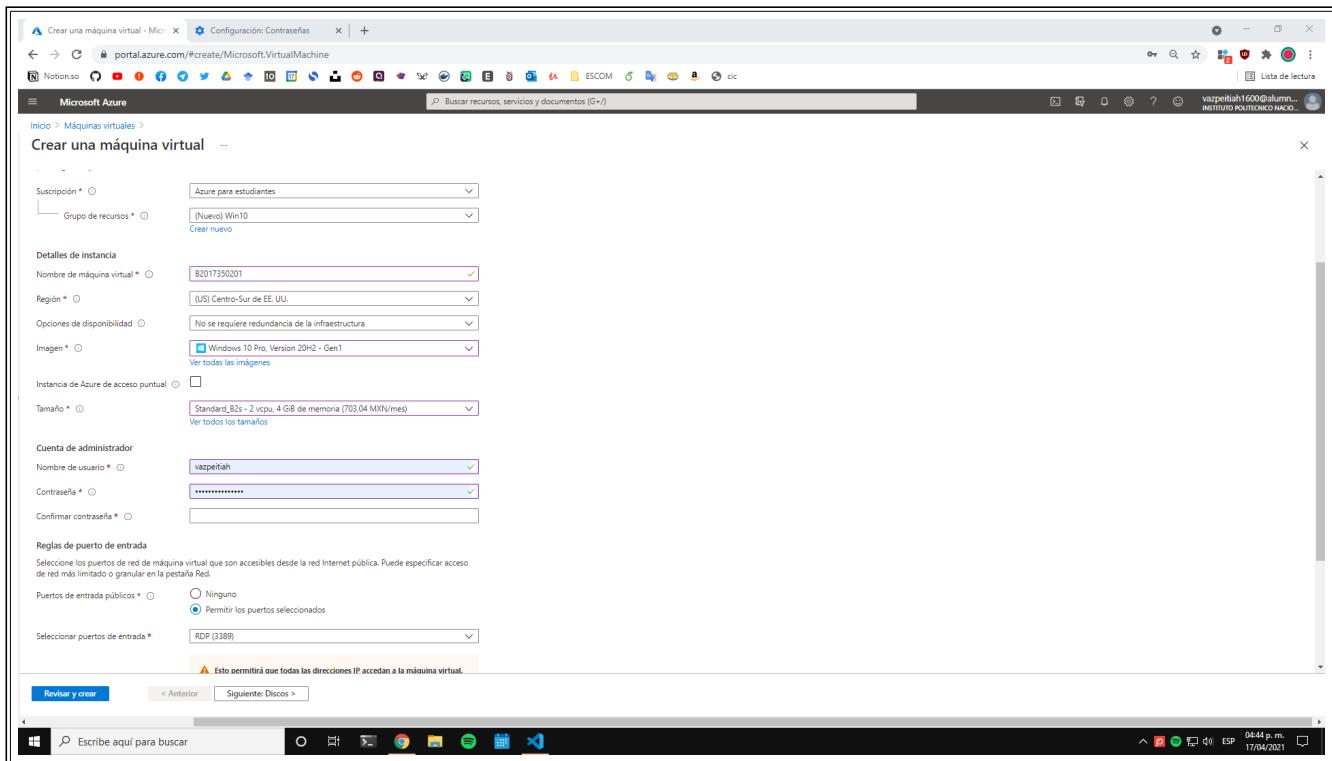


Figura 3: Creación de la máquina virtual de Windows 10 Pro: Información básica

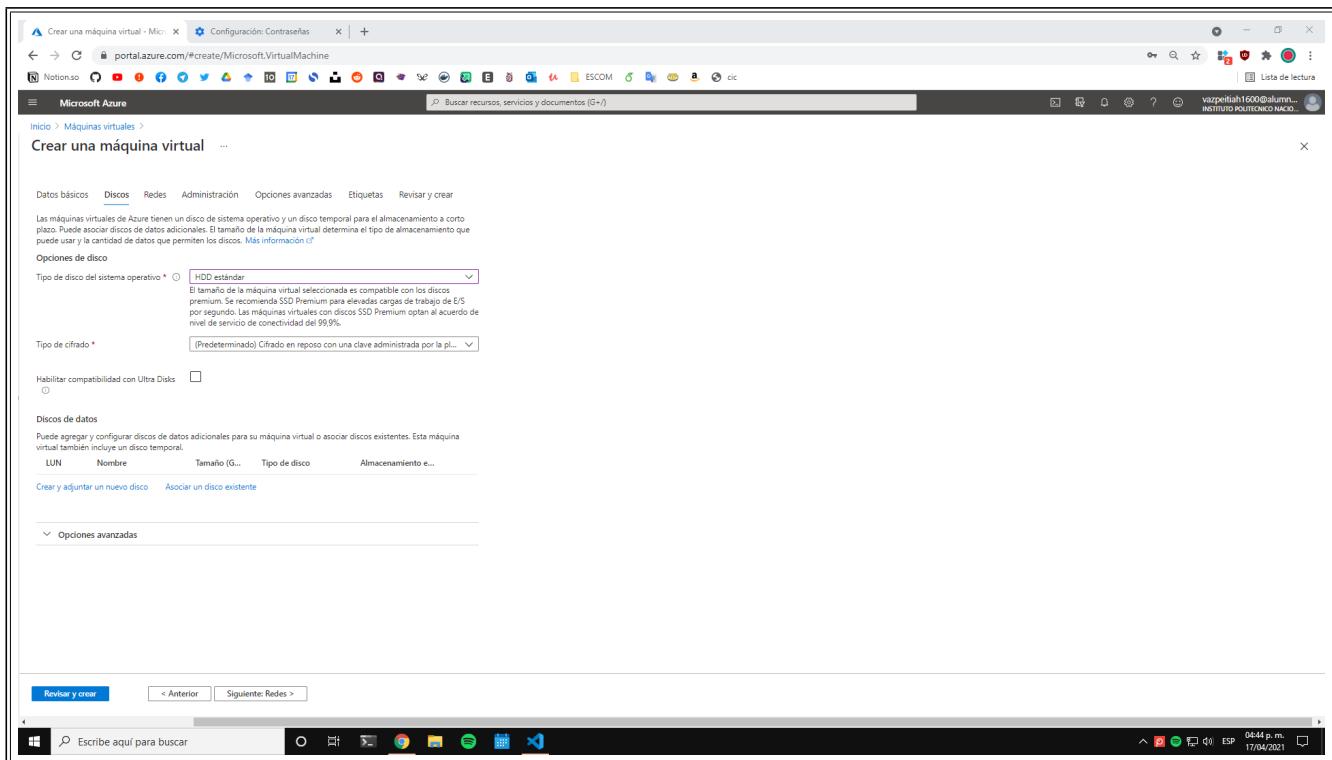


Figura 4: Creación de la máquina virtual de Windows 10 Pro: Discos

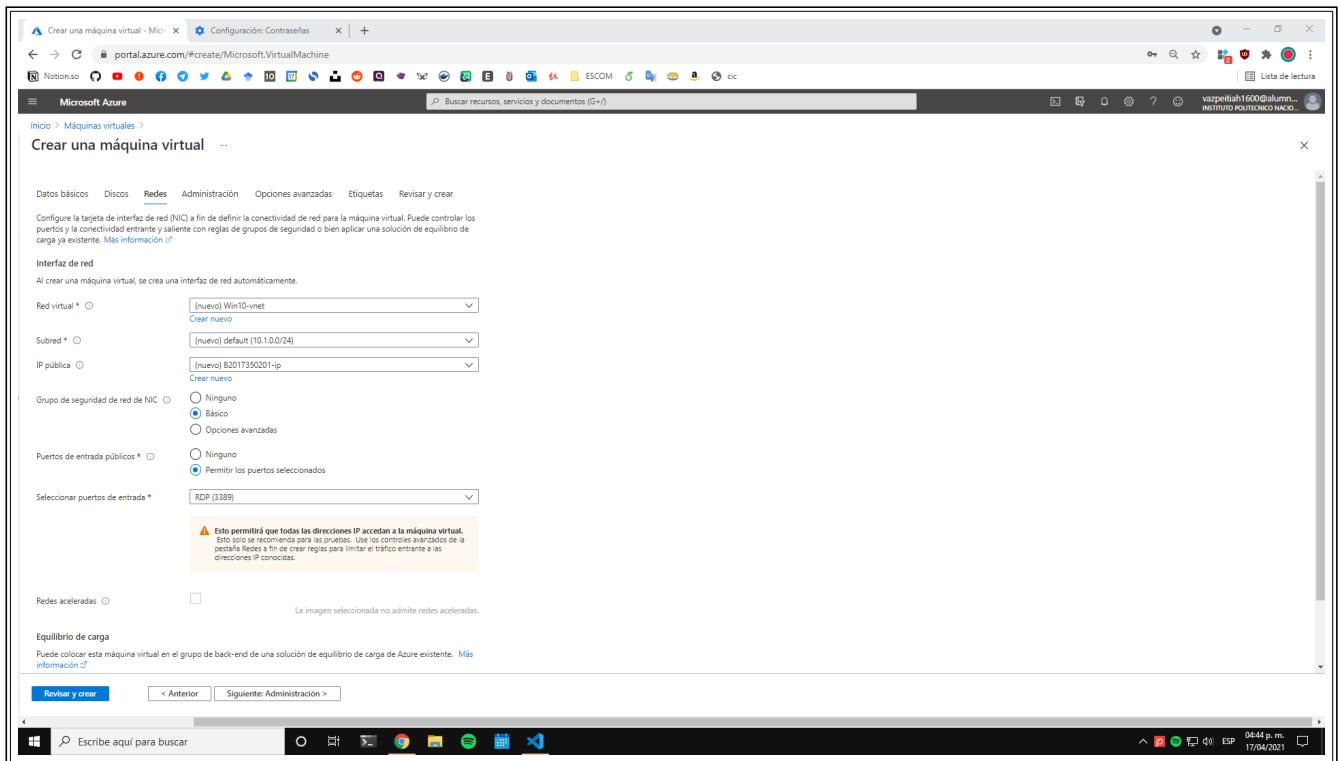


Figura 5: Creación de la máquina virtual de Windows 10 Pro: Redes

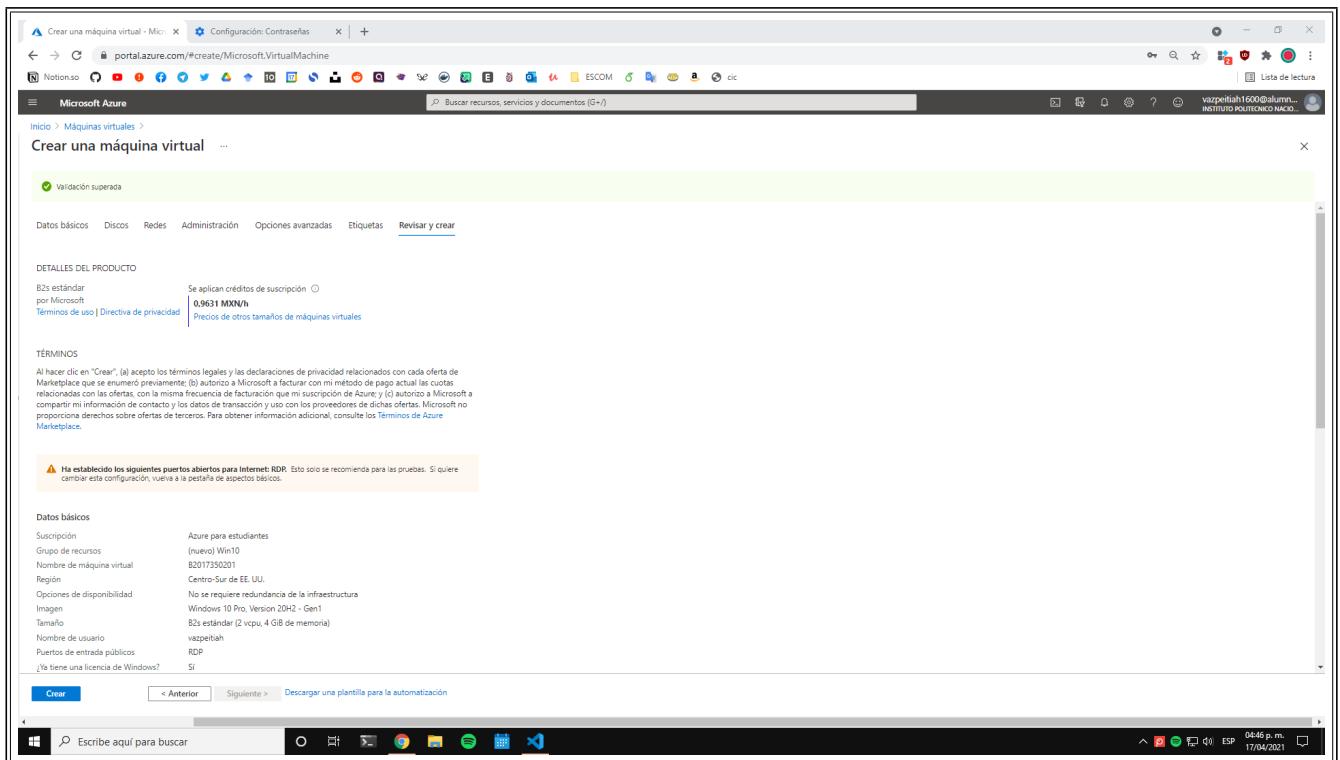


Figura 6: Creación de la máquina virtual de Windows 10 Pro: Revisar y crear

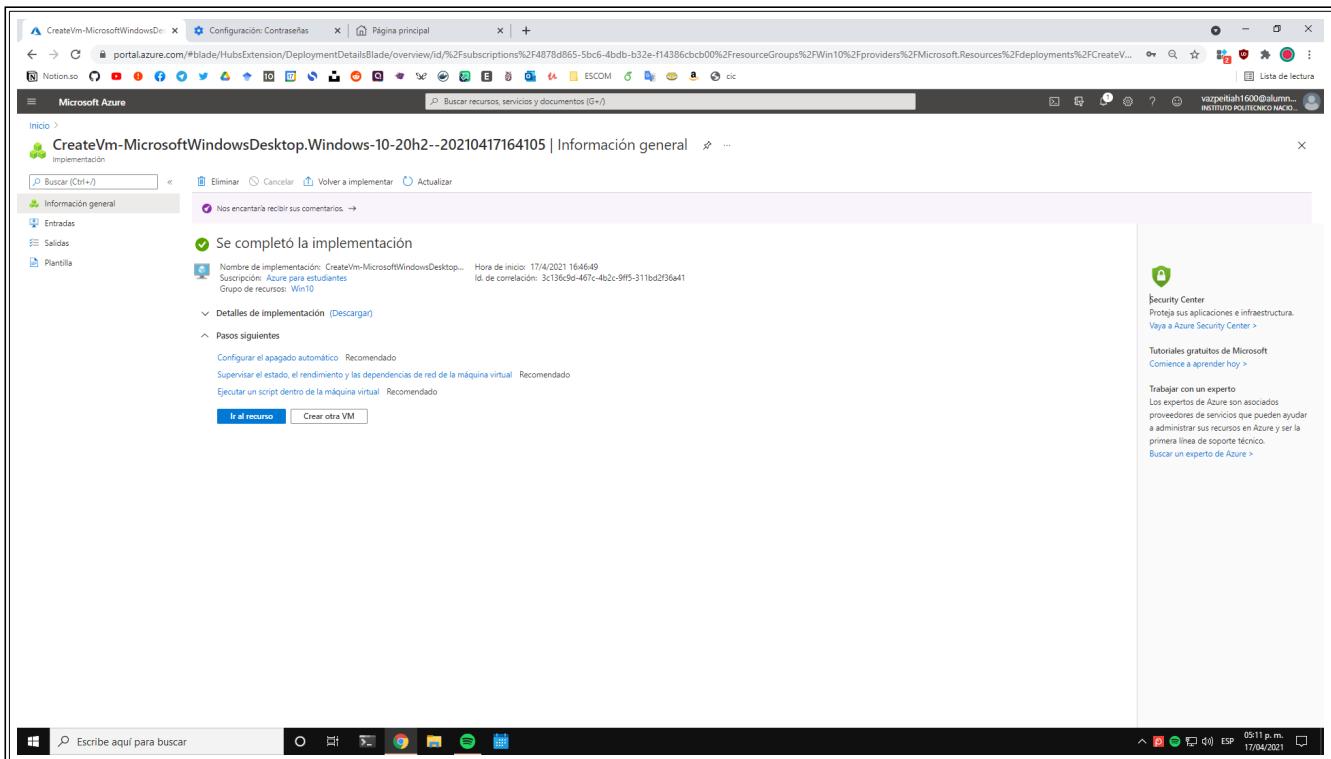


Figura 7: Creación de la máquina virtual de Windows 10 Pro: Implementación completada

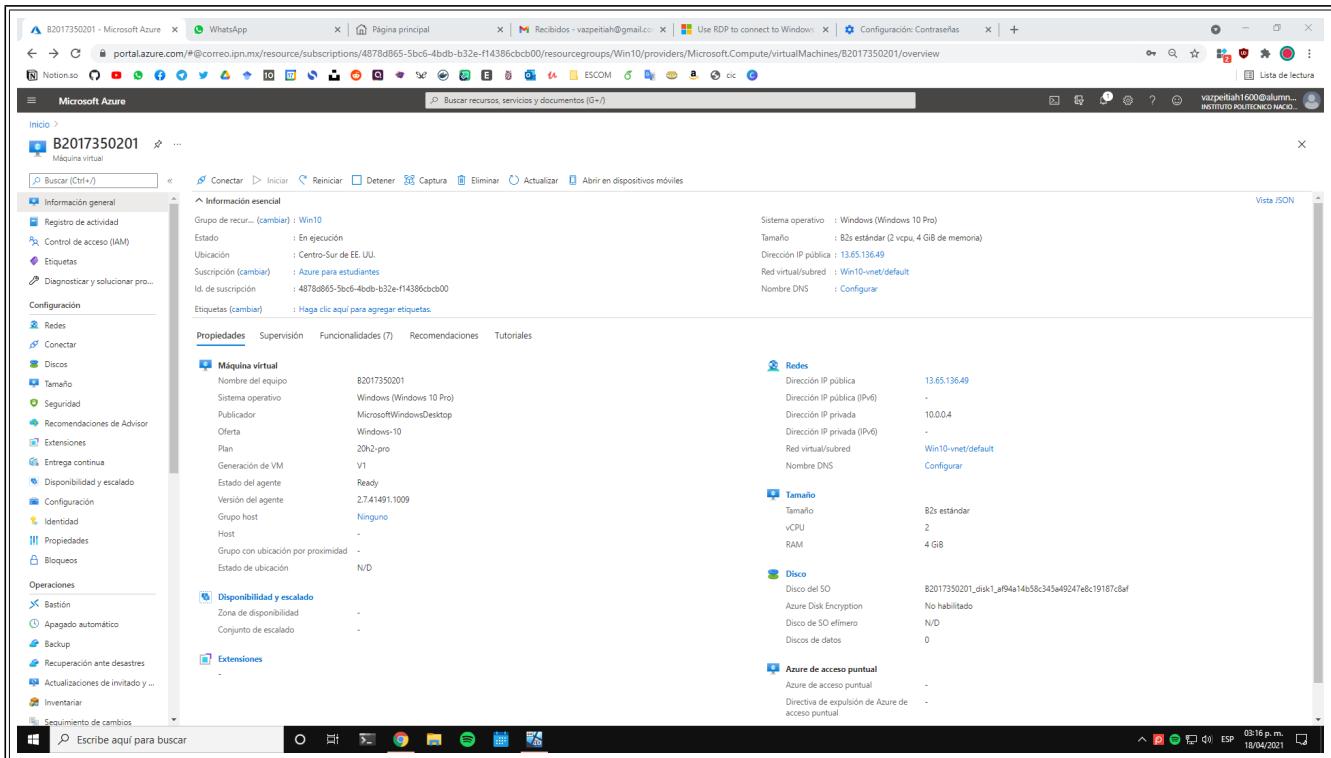


Figura 8: Creación de la máquina virtual de Windows 10 Pro: Panel de control de la máquina

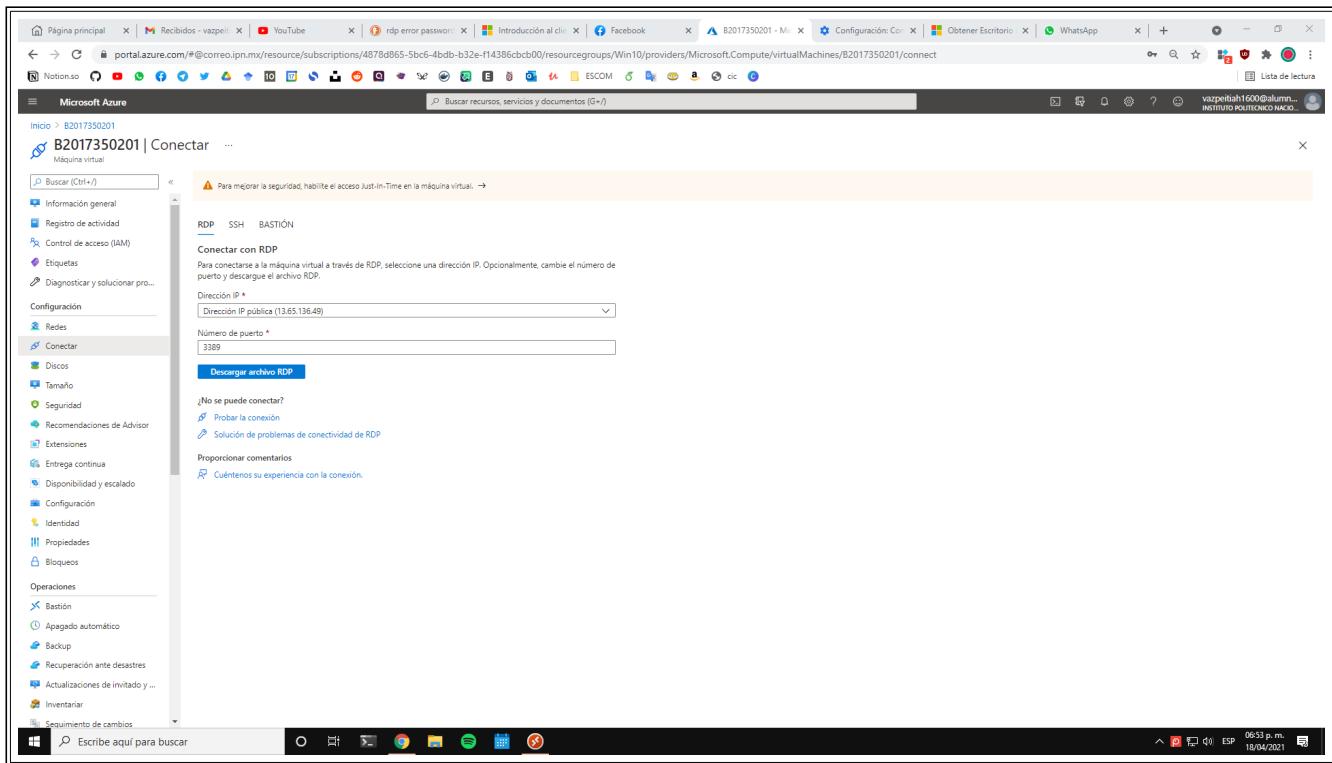


Figura 9: Creación de la máquina virtual de Windows 10 Pro: Conexión RDP

Como no puede establecer una conexión de escritorio remoto con la herramienta que viene por defecto en Windows 10 Pro, tuve que recurrir a esta alternativa:

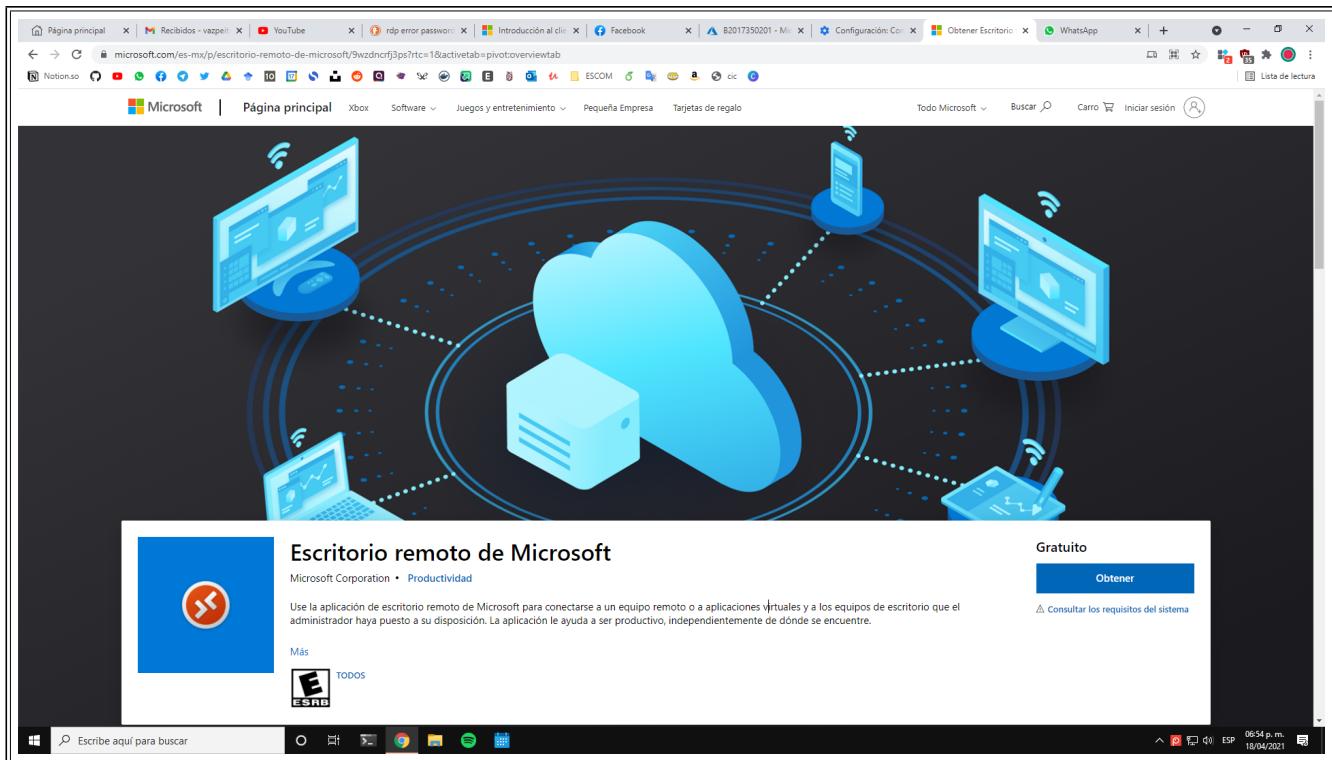


Figura 10: Creación de la máquina virtual de Windows 10 Pro: Instalando cliente RDP

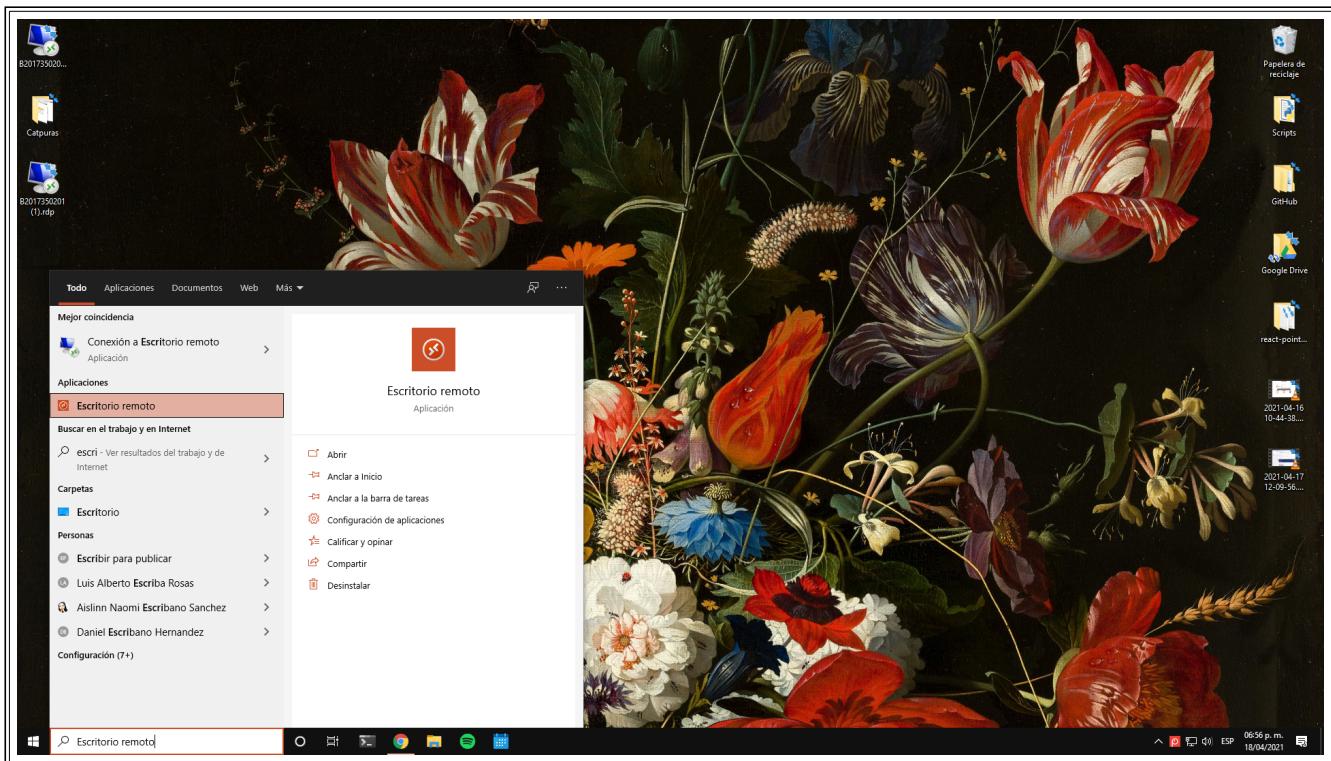


Figura 11: Creación de la máquina virtual de Windows 10 Pro: Abriendo cliente RDP

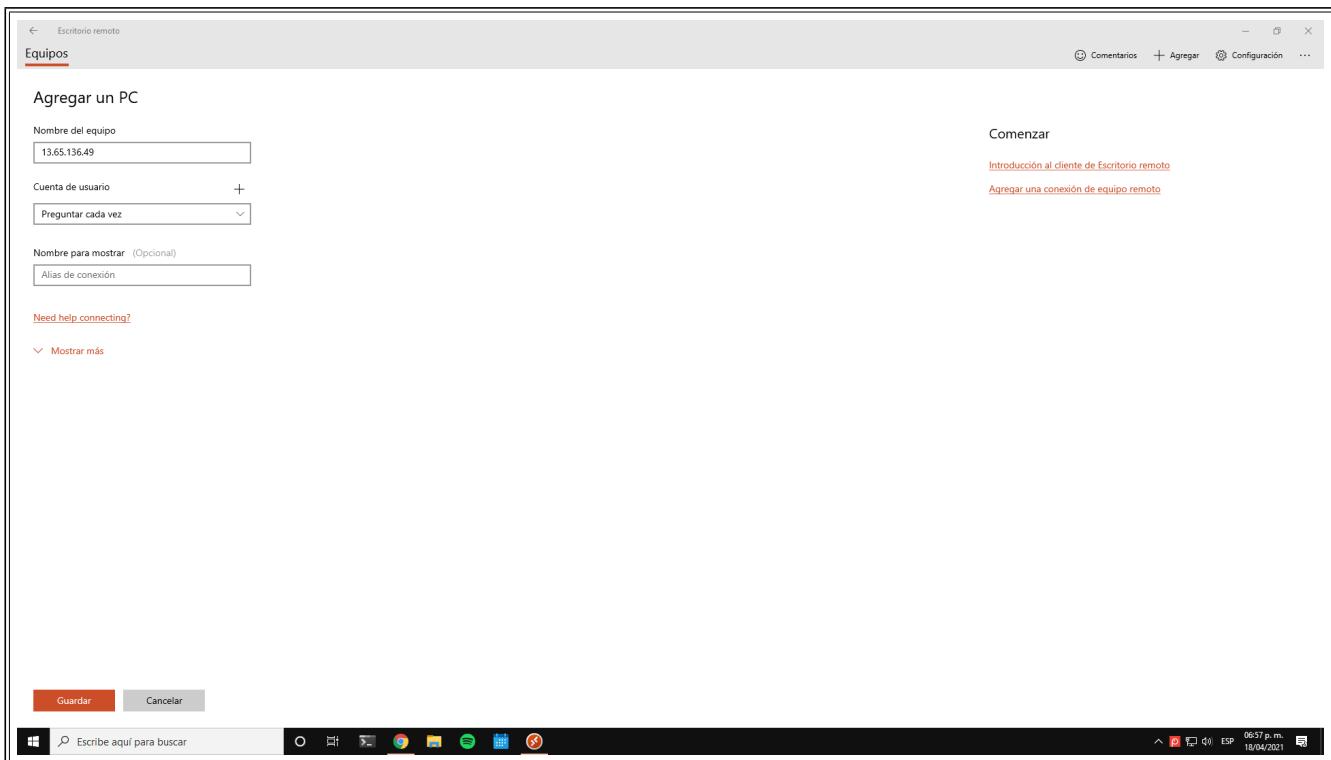


Figura 12: Creación de la máquina virtual de Windows 10 Pro: Ingresando dirección IP de la máquina

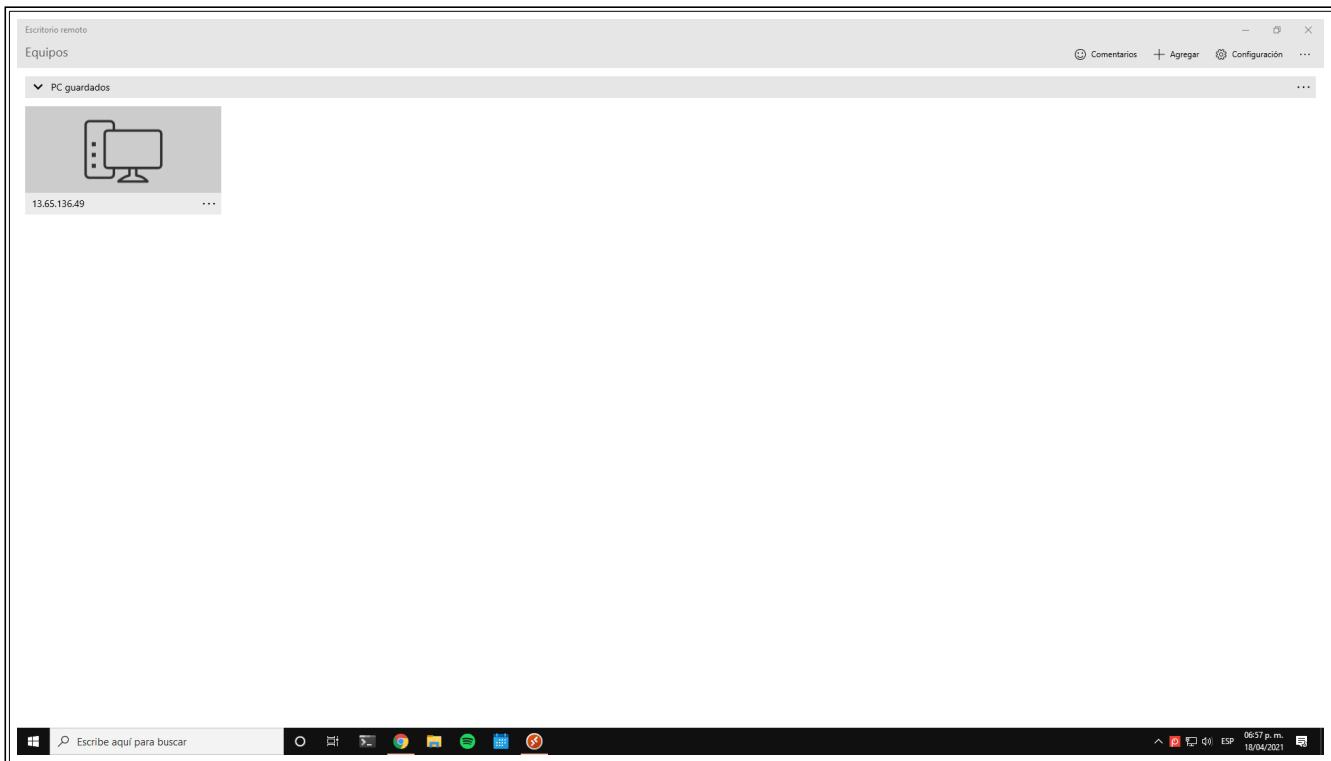


Figura 13: Creación de la máquina virtual de Windows 10 Pro: Abriendo conexión RDP con la máquina

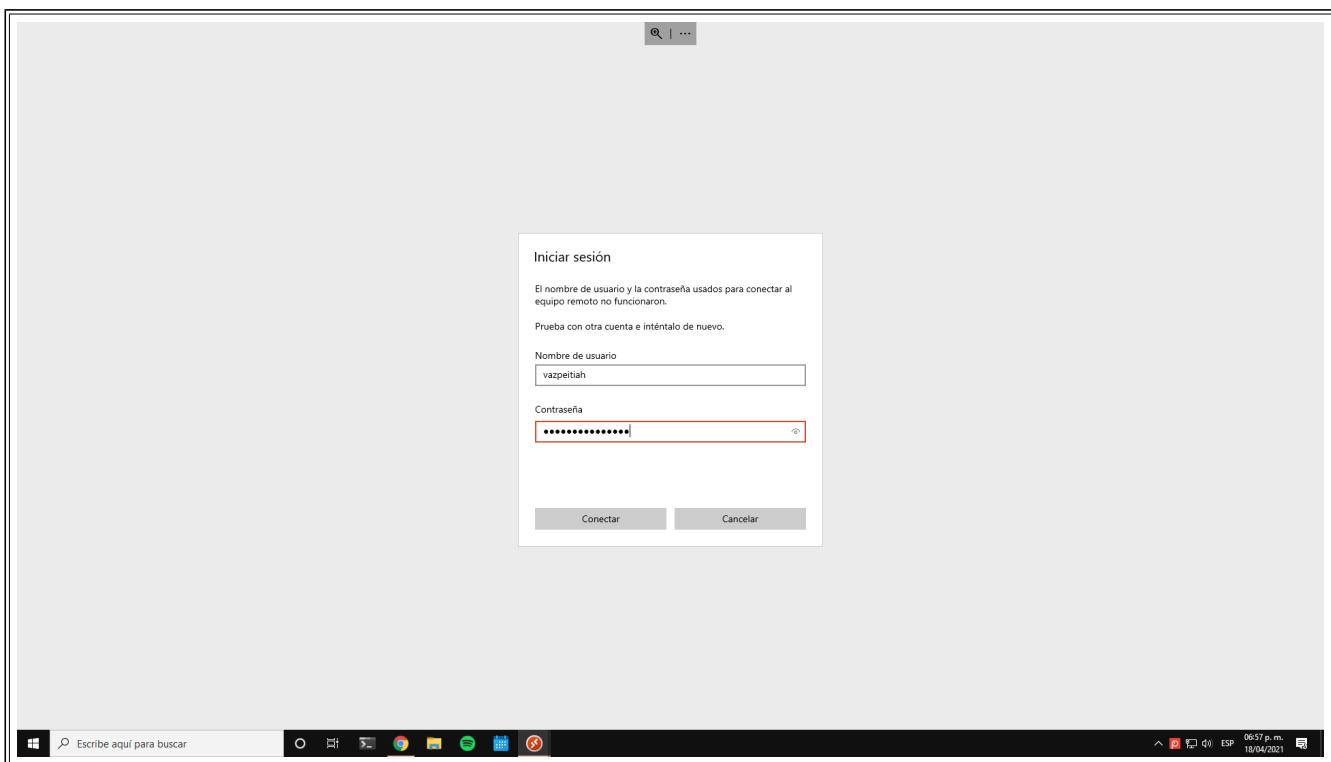


Figura 14: Creación de la máquina virtual de Windows 10 Pro: Iniciar sesión RDP

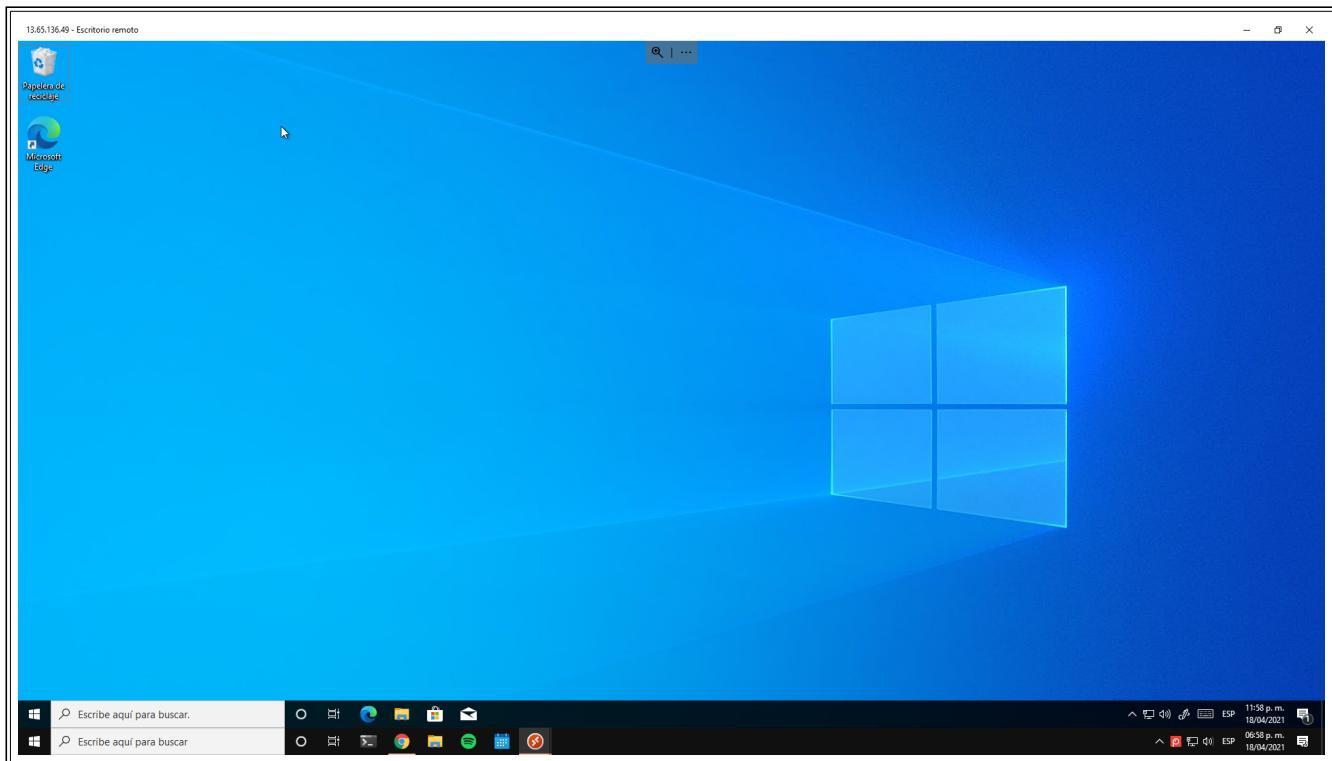


Figura 15: Creación de la máquina virtual de Windows 10 Pro: Escritorio remoto exitoso

Es necesario instalar Java en la máquina virtual, para poder ejecutar nuestros archivos .class.

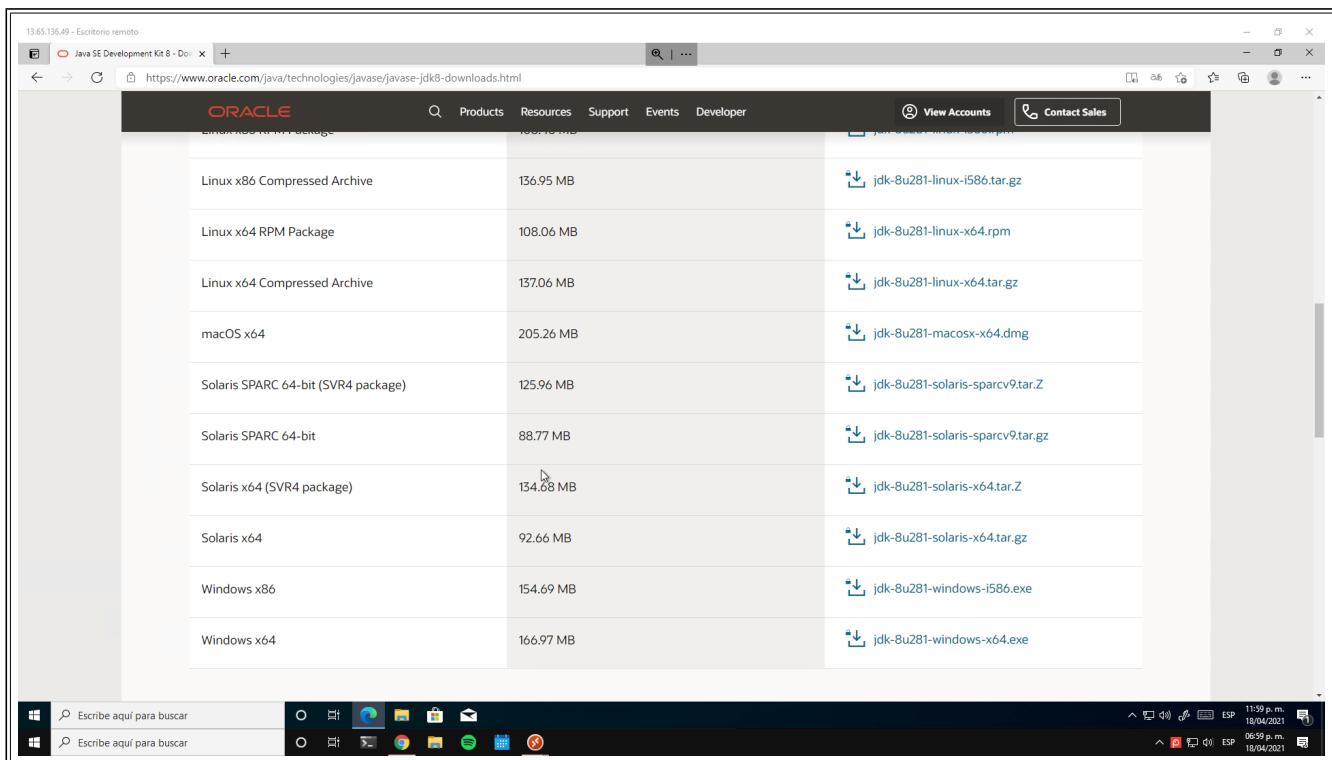


Figura 16: Creación de la máquina virtual de Windows 10 Pro: Instalando Java en la máquina virtual

Para subir el código fuente del chat a la máquina virtual, yo utilice GitHub.

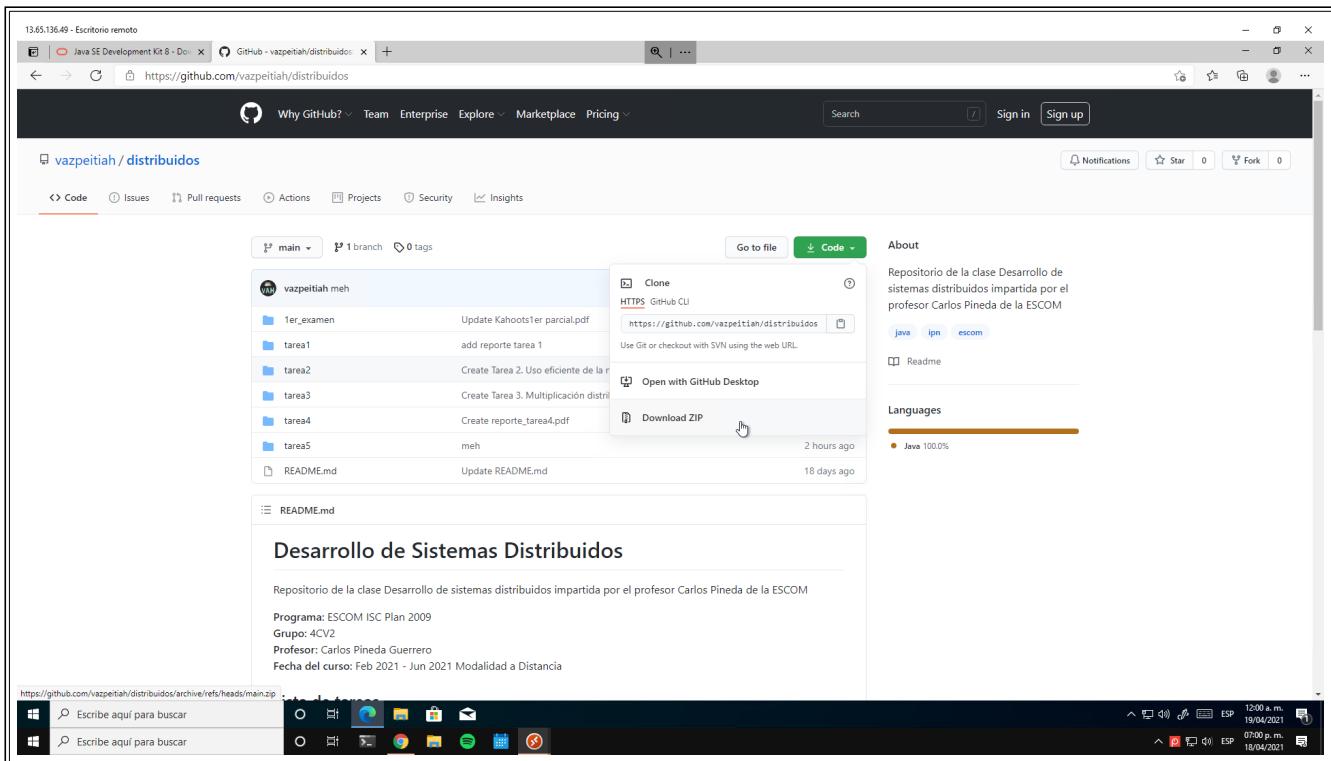


Figura 17: Creación de la máquina virtual de Windows 10 Pro: Subiendo código fuente del chat a la máquina virtual

## 2.3. Compilación y ejecución del programa

Una vez tengamos el código fuente del chat multicast en la máquina virtual, lo colocamos en el directorio que queramos y abrimos una terminal de windows en el directorio. Luego, simplemente compilamos el código de fuente:

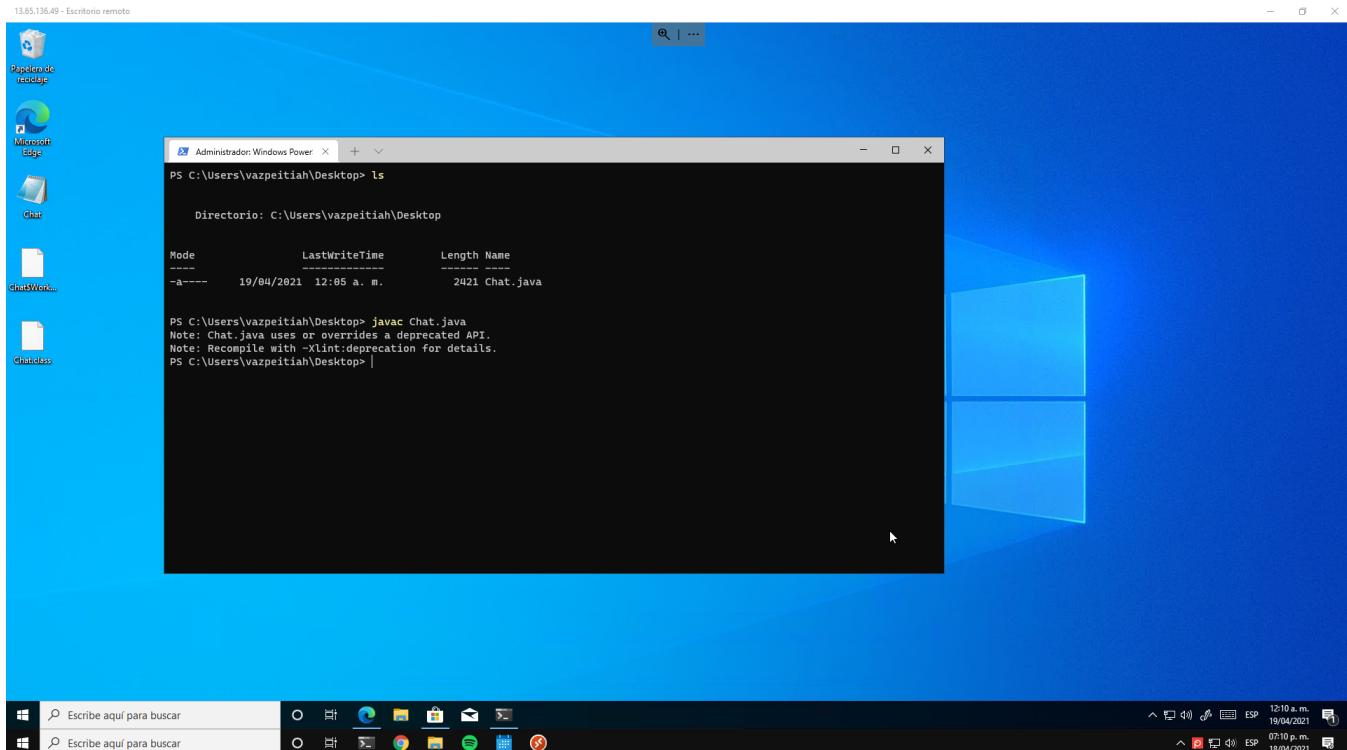


Figura 18: Compilando y ejecutando el programa Chat.java

A continuación se muestran las pruebas que se especificaron en la descripción de la tarea. Previamente a la ejecución, se cambiaron los código de página a 1252, para que se puedan visualizar los caracteres como acentos y signos de interrogación que abren.

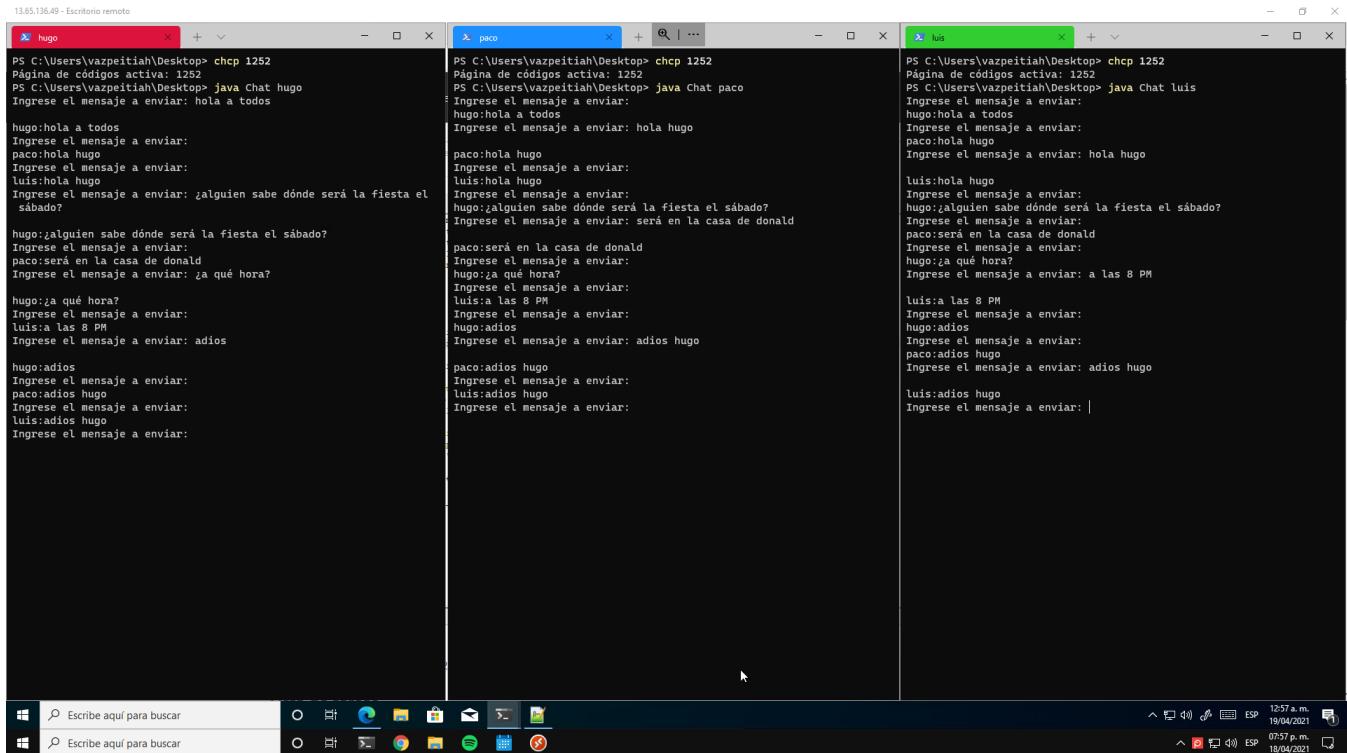


Figura 19: Ejecutando 3 veces el chat multicast

### 3. Conclusiones

En esta práctica desarrollamos un cliente multicast. Es más simple el desarrollo de este tipo de servidores, ya que los mensajes se envían a todos los dispositivos conectados a la misma red y al mismo puerto, realmente no hay mayor problema en el desarrollo del chat. Tardé mucho más tiempo en la implementación de la máquina virtual, principalmente porque mi computadora tiene un problema estableciendo conexiones de escritorio remota mediante RDP. No pude acceder a la máquina virtual con la herramienta que viene instalado por defecto en Windows 10. Afortunadamente, investigando en internet me di cuenta que podemos utilizar más herramientas para realizar este tipo de conexiones, así que descargue una que está disponible dentro de la tienda de Microsoft y finalmente pude establecer una conexión con la máquina virtual de Windows 10 Pro.