



**INSTITUTO POLITÉCNICO NACIONAL**

ESCUELA SUPERIOR DE CÓMPUTO

# Desarrollo de Sistemas Distribuidos

**“Tarea 3: Chat Multicast”**

**Grupo: 4CV13**

**Alumno:**

**Godinez Morales Mario Sebastian**

**Profesor:**

**Pineda Guerrero Carlos**

# Desarrollo

Para la primera parte de la práctica le pedí a ChatGPT que desarrollara un chat acorde a las instrucciones de la práctica 3 en un solo prompt, como respuesta me devolvió el código funcional, solo le di formato a las impresiones en la consola.

## Creación de la máquina virtual

Página principal de azure

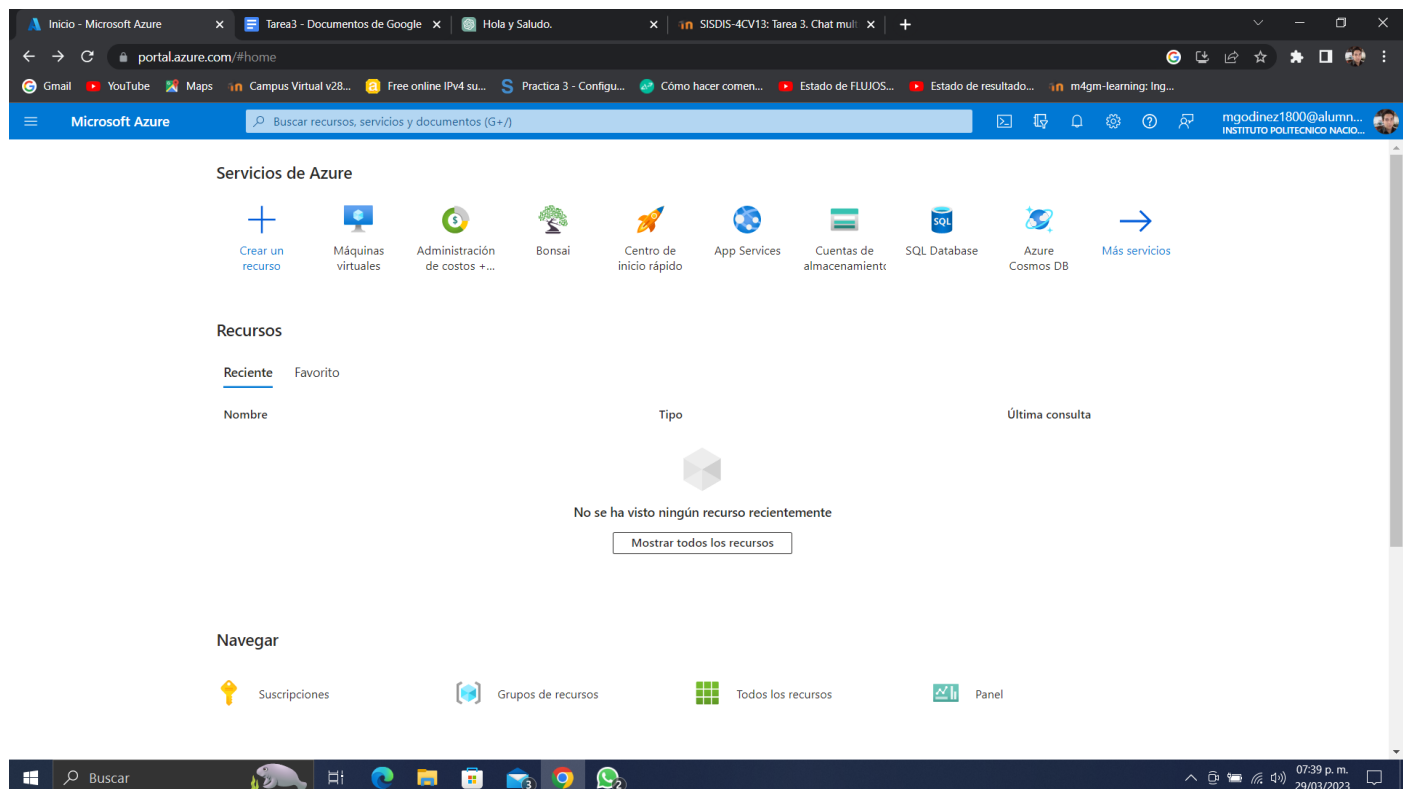


imagen 1

Seleccionamos máquinas virtuales y en el botón de crear seleccionamos máquina virtual de Azure como se muestra en la imagen 2.

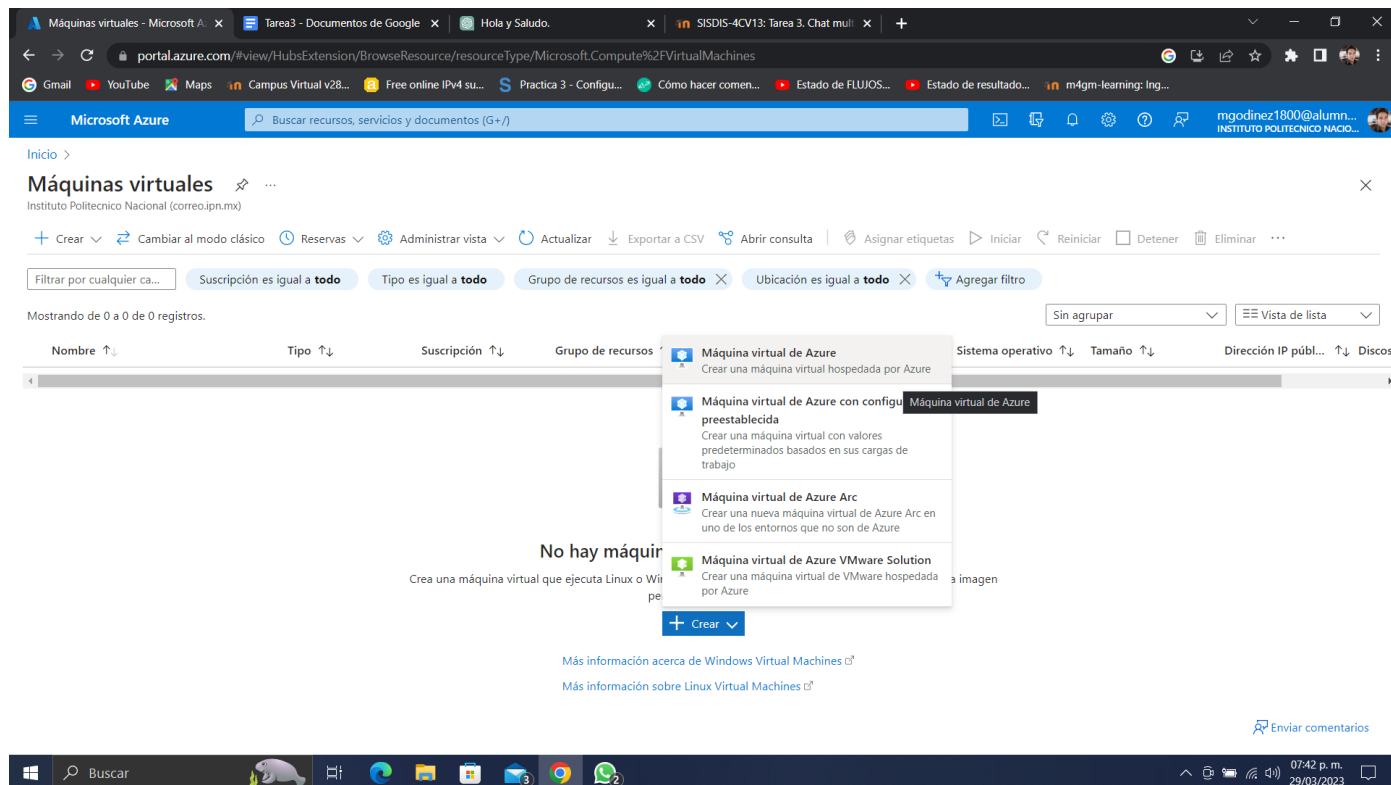


imagen 2

En la información básica seleccionamos la suscripción y el grupo de recursos en donde se creará el recurso, posteriormente nombramos la máquina virtual, en mi caso es T3-2019630034 y la región en donde se creará como se muestra en la imagen 3.

Crear una máquina virtual - Microsoft Azure

portal.azure.com/#create/Microsoft.VirtualMachine

Inicio >

## Crear una máquina virtual

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

Cree 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](#)

Es posible que esta suscripción no sea apta para implementar máquinas virtuales de ciertos tamaños en determinadas regiones.

### 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 \* Azure for Students

Grupo de recursos \* (Nuevo) Tarea3

[Crear nuevo](#)

### Detalles de instancia

Nombre de máquina virtual \* T3-2019630034

Región \* (US) East US

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

Enviar comentarios

imagen 3

Posteriormente se elegimos el SO que va a tener y configuramos el nombre de usuario y contraseña como se muestra en la imagen 4.

Crear una máquina virtual

Imagen \* ⓘ Windows Server 2012 Datacenter - x64 Gen1  
[Ver todas las imágenes](#) | [Configurar la generación de máquinas virtuales](#)

There is a generation 2 version of this image available which has higher feature compatibility. [Click here to swap to the generation 2 version](#)

Arquitectura de VM ⓘ  
☐ Arm64  
☒ x64  
i Arm64 no es compatible con la imagen seleccionada.

Ejecución de Azure Spot con descuento ⓘ ☐

Tamaño \* ⓘ Standard\_B1s - 1 vcpu, 1 GiB de memoria (10,22 US\$/mes)  
[Ver todos los tamaños](#)

Cuenta de administrador

Nombre de usuario \* ⓘ mario ✓

Contraseña \* ⓘ ..... ✓

Confirmar contraseña \* ⓘ ..... ✓

Revisar y crear < Anterior Siguiente: Discos > Enviar comentarios

imagen 4

Finalmente revisamos los puertos de entrada que nos van a servir para conectarnos a nuestra VM como se muestra en la imagen 5.

Crear una máquina virtual - Micr...

Tarea3 - Documentos de Google

Hola y Saludo.

SISDIS-4CV13: Tarea 3. Chat mult...

portal.azure.com/#create/Microsoft.VirtualMachine

Gmail

YouTube

Maps

Campus Virtual v28...

Free online IPv4 su...

Practica 3 - Configu...

Cómo hacer comen...

Estado de FLUJOS...

Estado de resultado...

m4gm-learning: Ing...

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Inicio >

### Crear una máquina virtual

Cuenta de administrador

Nombre de usuario \* ⓘ

Contraseña \* ⓘ

Confirmar contraseña \* ⓘ

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puestos de entrada públicos \* ⓘ

Seleccionar puertos de entrada \*

Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Revisar y crear

< Anterior

Siguiendo: Discos >

Enviar comentarios

Buscar

07:52 p. m.  
29/03/2023

imagen 5

Posteriormente configuramos la parte del almacenamiento en la VM, en nuestro caso elegimos un HDD estándar como se muestra en la imagen 6

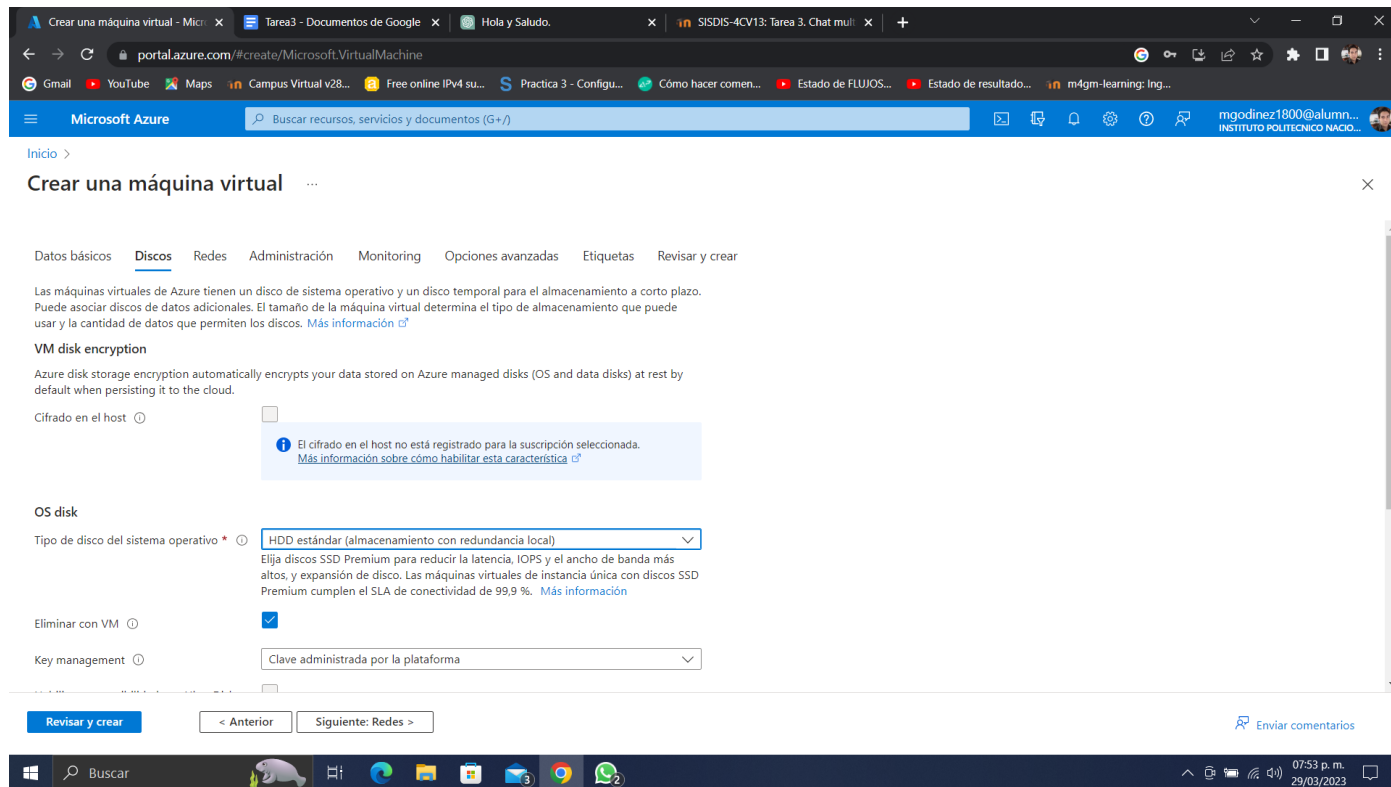


imagen 6

Para la parte de redes no configuramos nada.

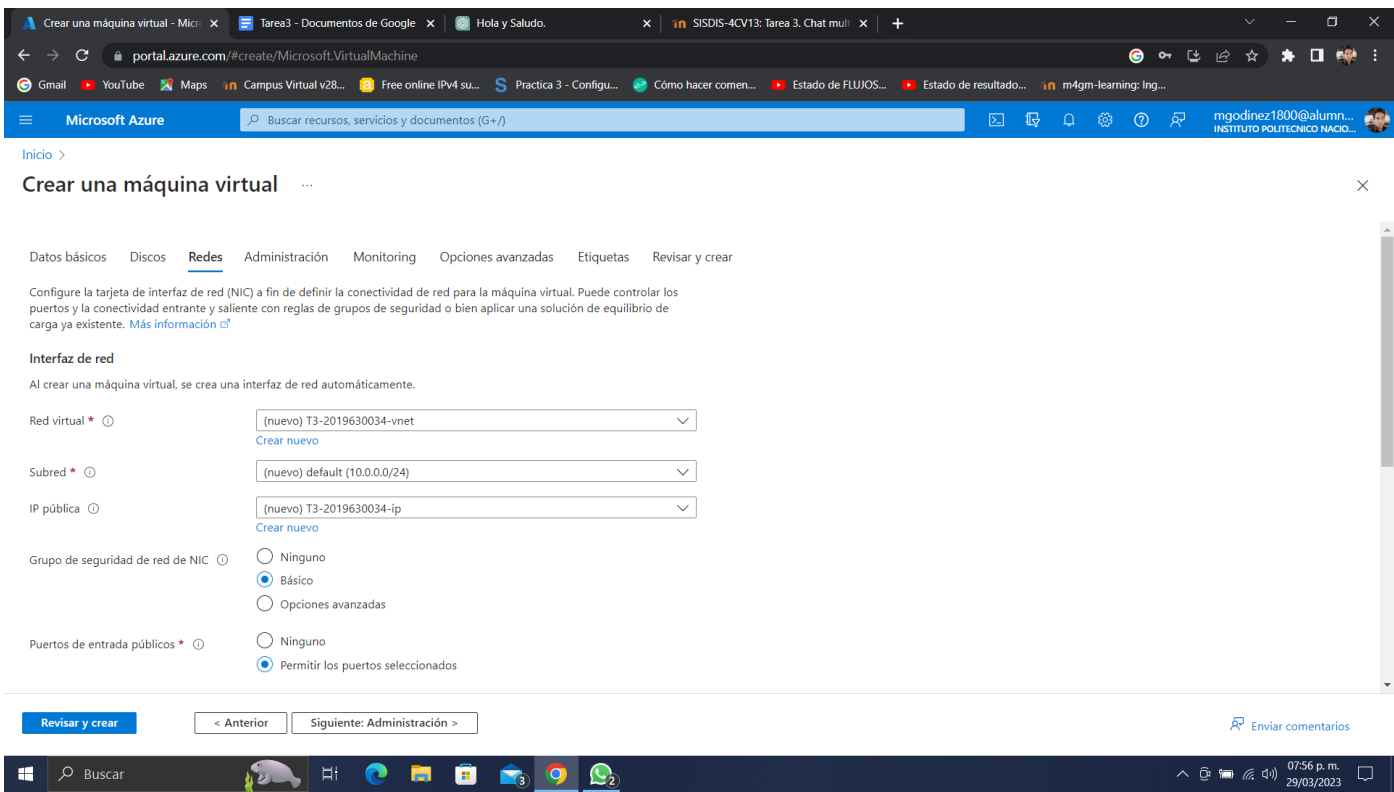


imagen 7



Para la parte de administración no modificamos nada.

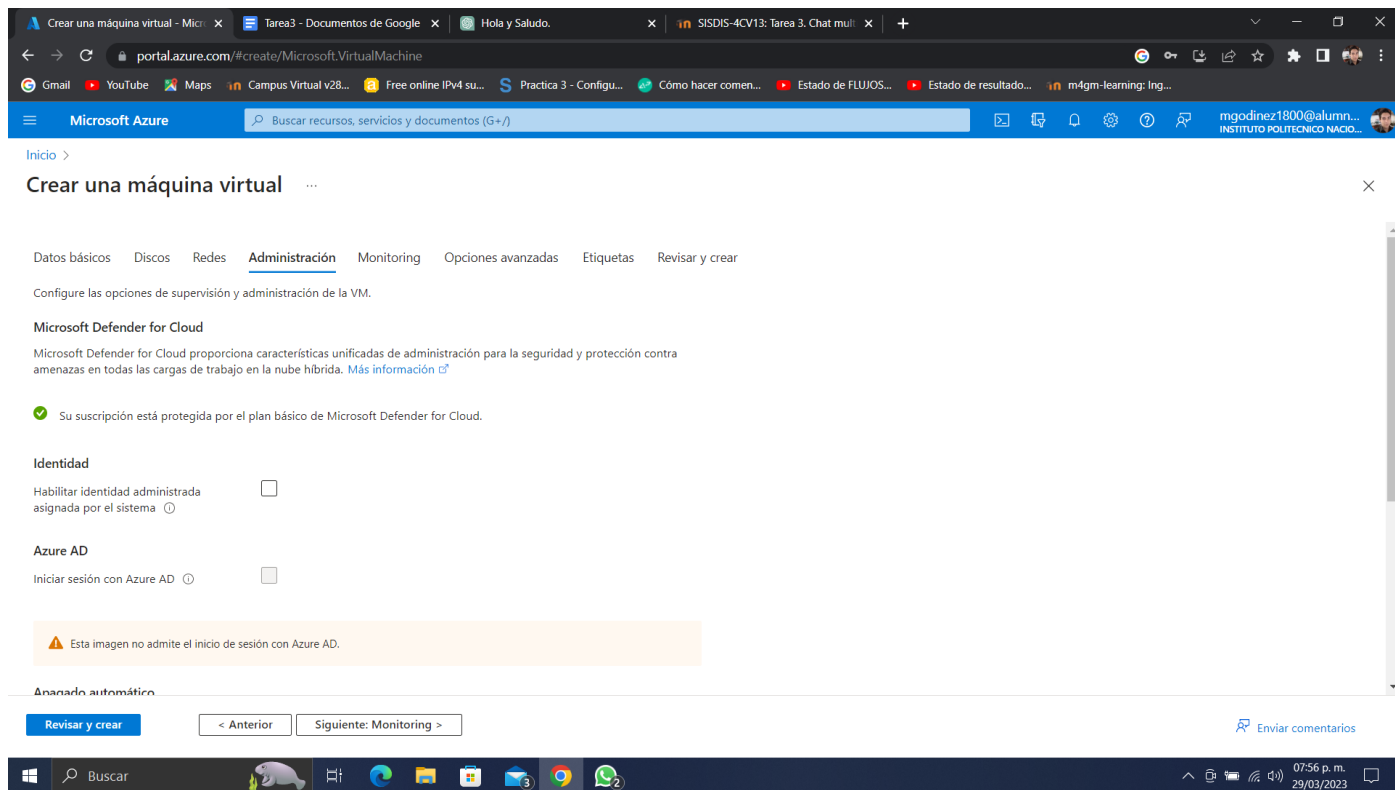


imagen 8

Finalmente damos en revisar y crear.

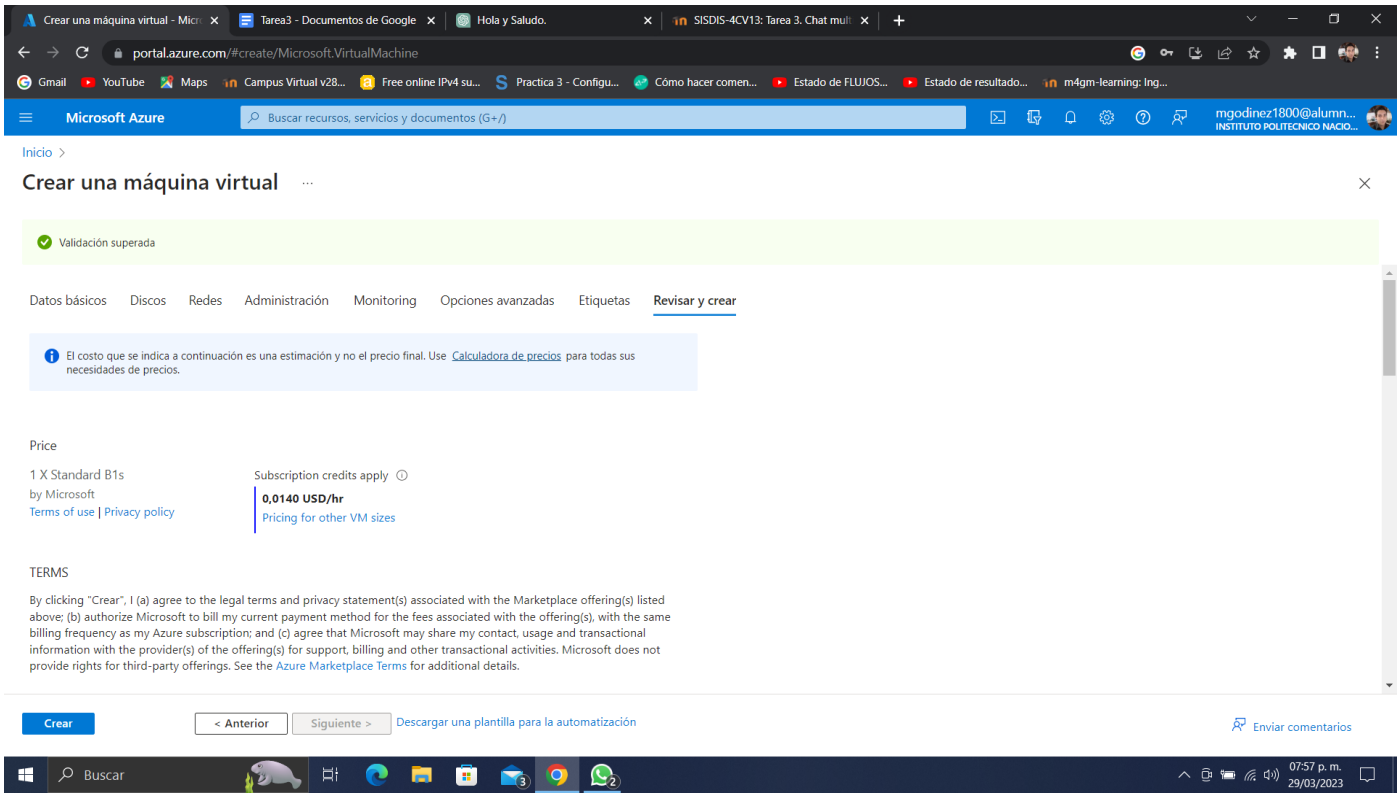


imagen 9

Seleccionamos el botón crear y observamos que la implementación está en curso como se muestra en la imagen 10.

The screenshot displays the Microsoft Azure portal interface. The browser's address bar shows the URL: `portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F057dbe04-e860-4a02-b4e2-eee84f97b246%2FresourceGroups%2FTarea3.../deployments/2`. The page title is "CreateVm-MicrosoftWindowsServer.WindowsServer-201-20230329194600 | Información general". The left sidebar contains a search bar and a list of navigation items: "Información general", "Entradas", "Salidas", and "Plantilla". The main content area features a status message: "La implementación está en curso" (The deployment is in progress). Below this, a table provides details about the deployment:

Recurso	Tipo	Estado	Detalles de la operación
No hay ningún resultado.			

Additional information includes the deployment name "CreateVm-MicrosoftWindowsServer...", the subscription "Azure for Students", and the resource group "Tarea3". The start time is "29/3/2023, 19:57:46" and the correlation ID is "c2bf2727-b8bc-4bcb-ae5f-56acfb36b7". The right sidebar contains links to "Microsoft Defender for Cloud", "Tutoriales gratuitos de Microsoft", and "Trabajar con un experto". The Windows taskbar at the bottom shows the time as 07:57 p.m. on 29/03/2023.

imagen 10

Una vez implementada la VM para conectarnos lo haremos mediante el protocolo RDP, para esto iremos al recurso de la VM y seleccionamos en conectar como se muestra en la imagen 11.

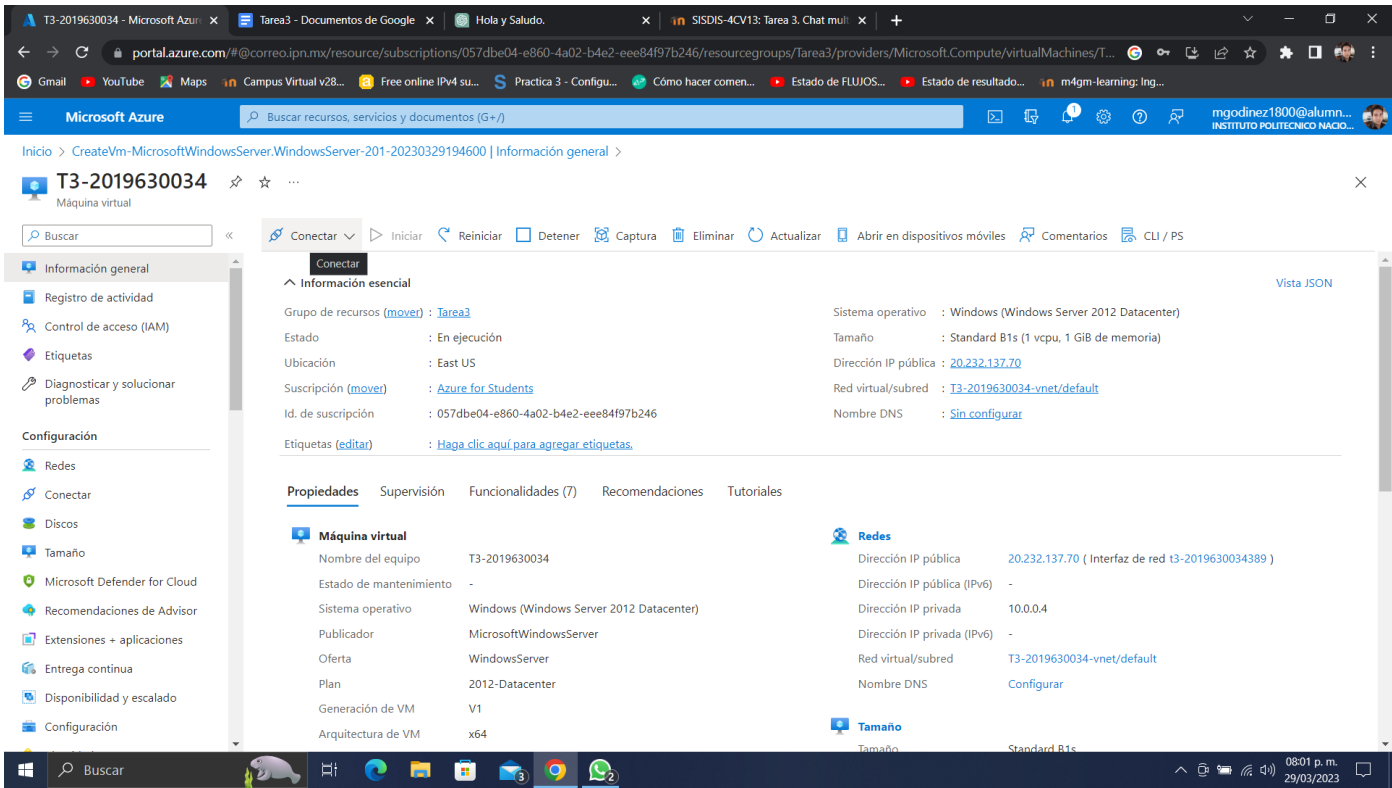


imagen 11

Luego seleccionamos RDP y verificamos que la ip pública y el puerto estén bien, y posteriormente descargamos el archivo RDP como se muestra en la imagen 12.

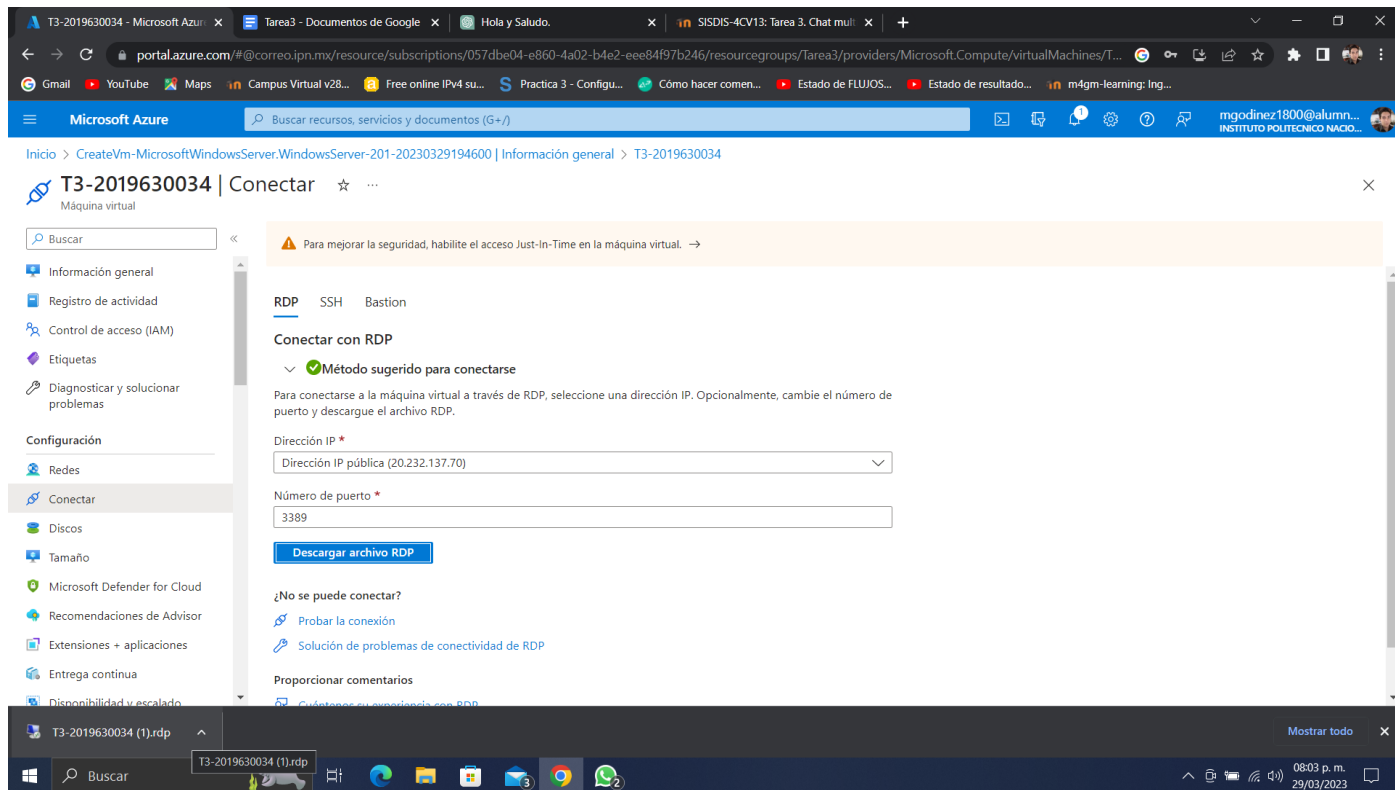


imagen 12

El archivo tendrá por nombre el nombre de la VM.rdp, en nuestro caso es T3-2019630034.rdp

Dicho archivo se abre con la aplicación de escritorio remoto incluida en windows, al darle click nos muestra lo siguiente:

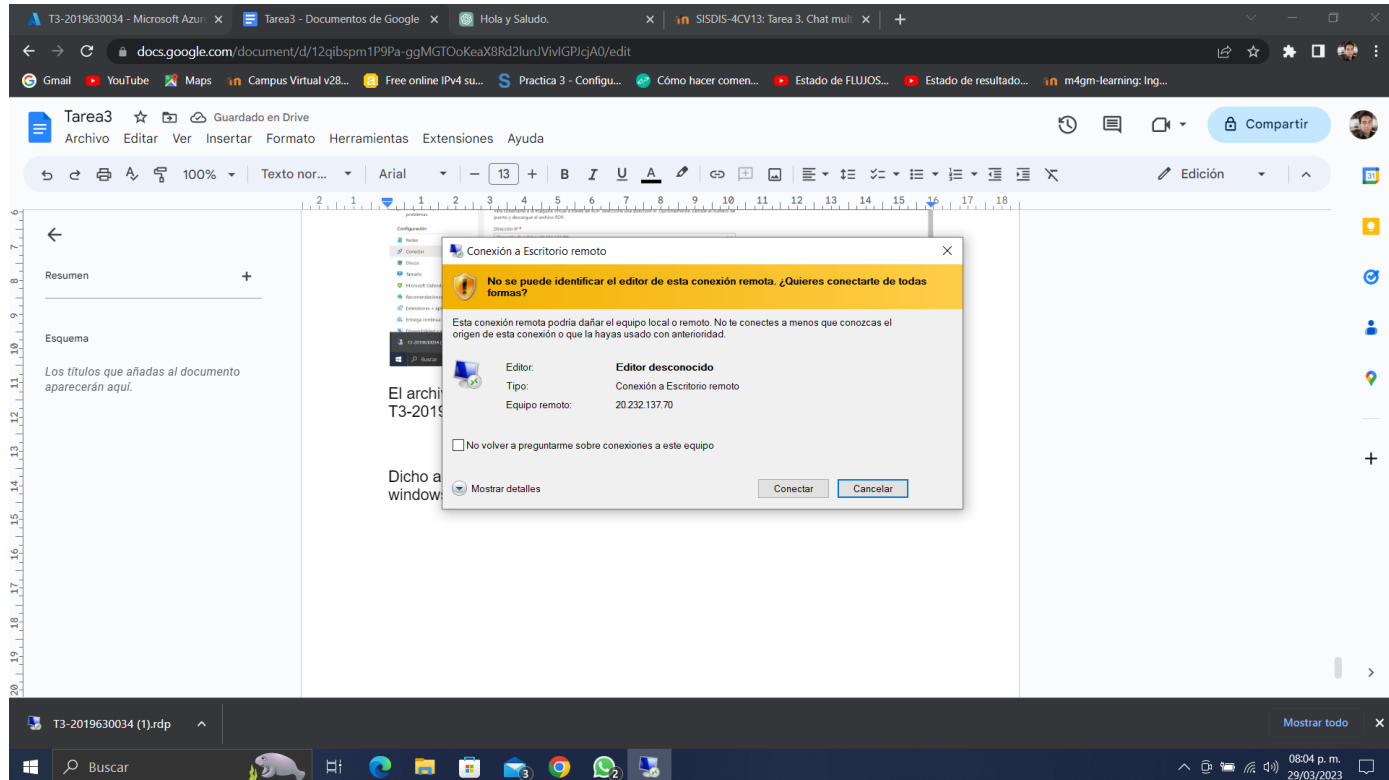


imagen 13

Damos click en conectar.

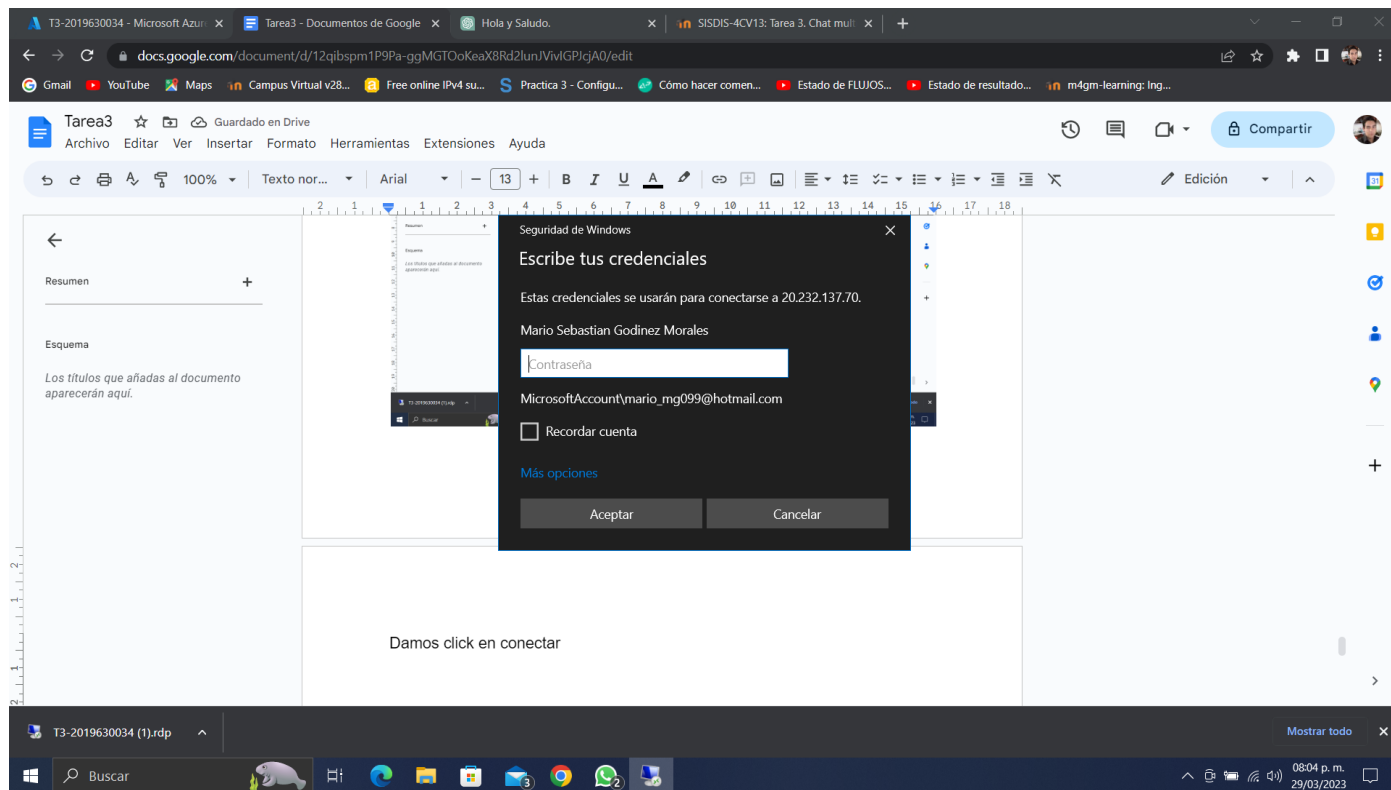


imagen 14

Nos pedirá la contraseña de la VM que introducimos en el apartado de la información básica.

Al introducirla nos pedirá confirmar la conexión y nos mostrará el nombre de la VM como se muestra en la imagen 15:

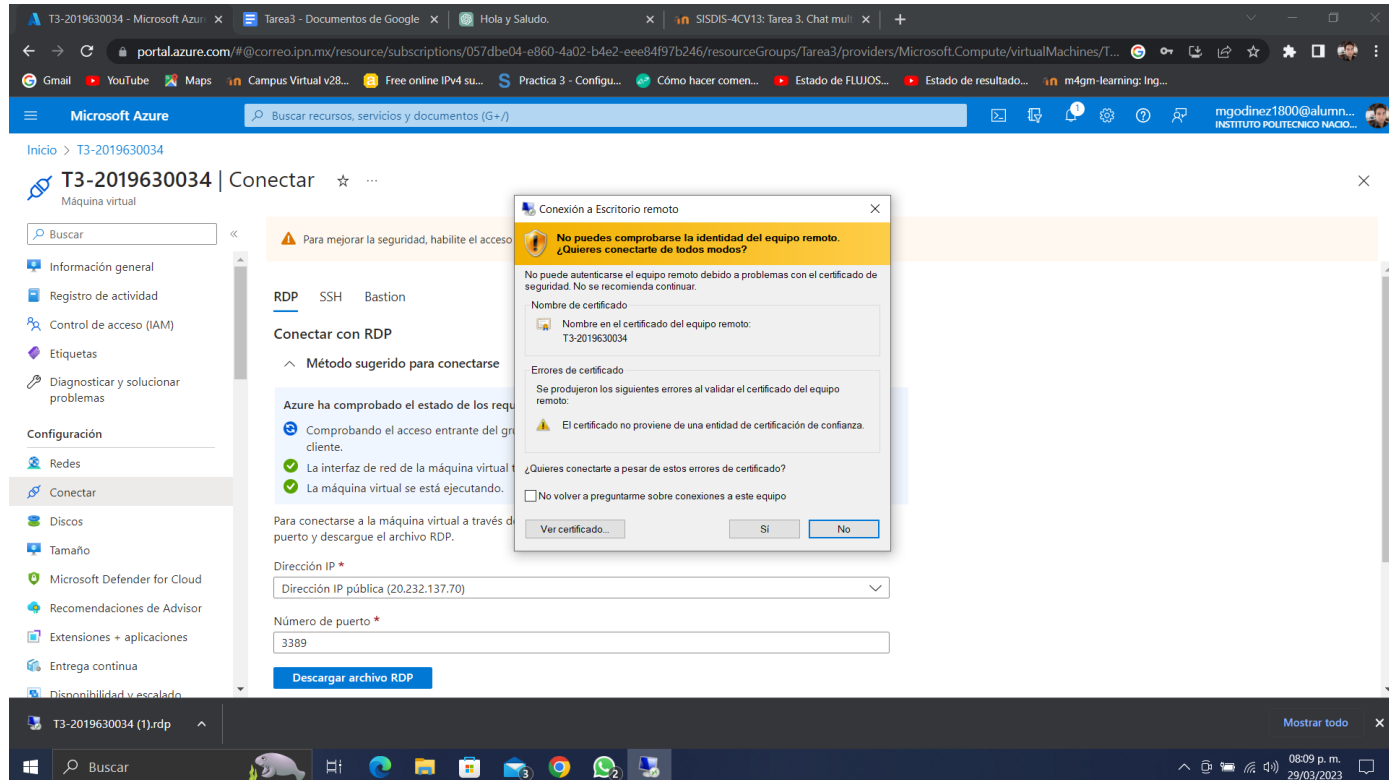


imagen 15



Damos click en si y ya entraremos a nuestra VM.

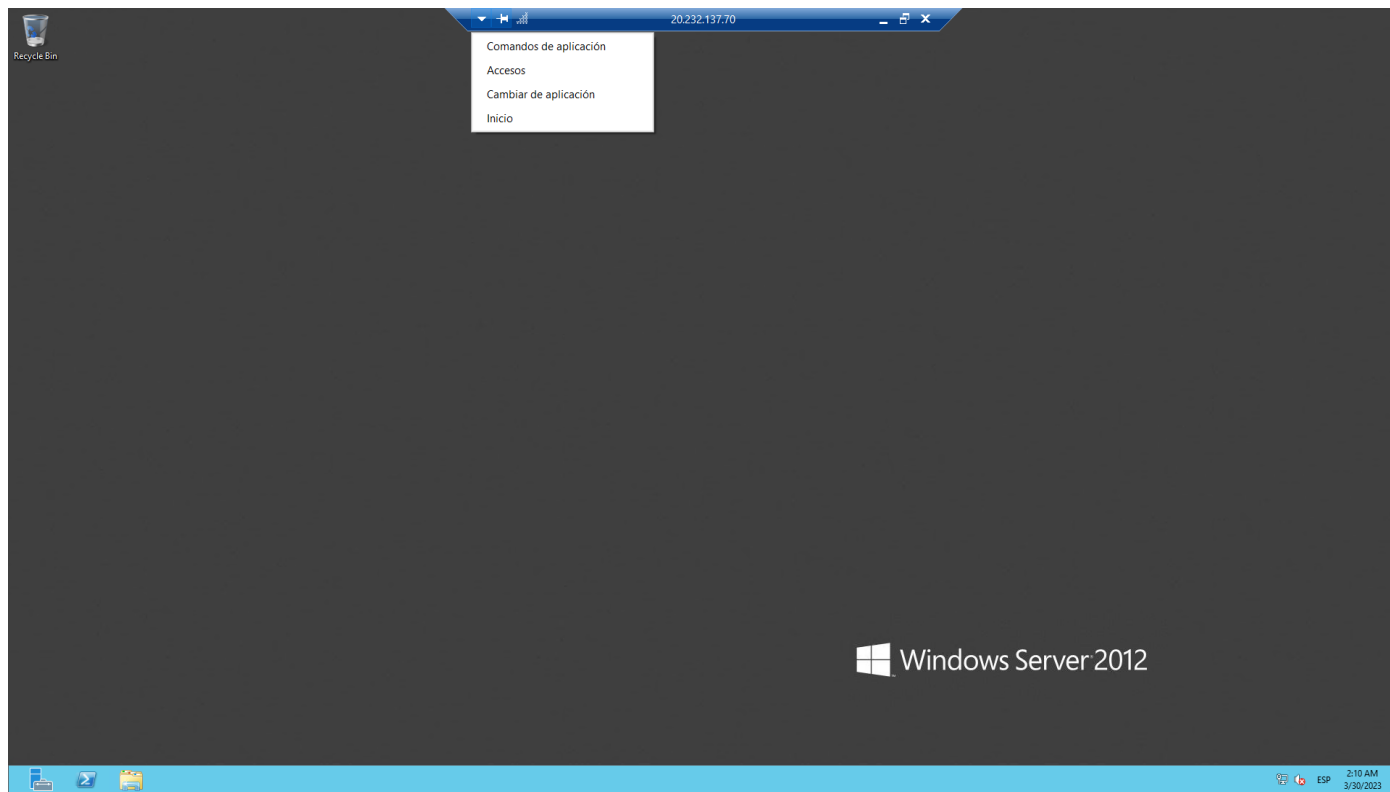


imagen 16

# Instalación de Java en la VM

Buscamos Java SE en el navegador de la VM y descargamos la versión 17.

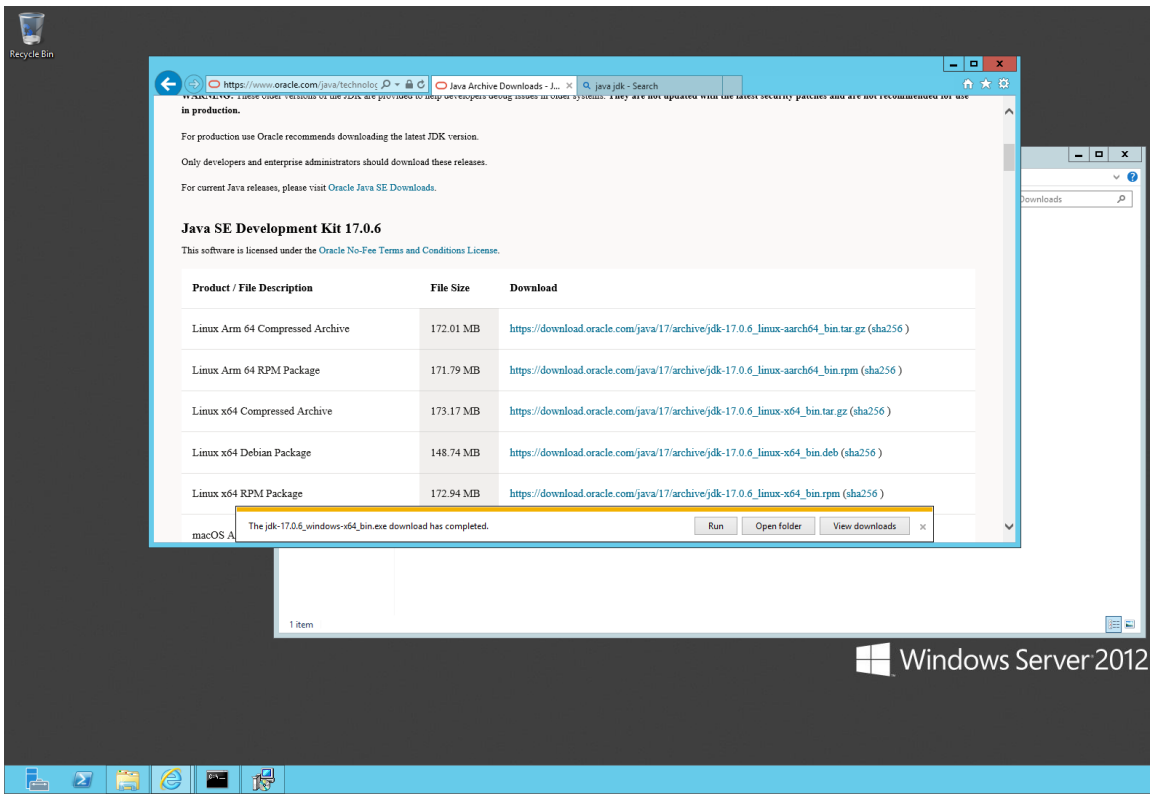


imagen 18

Ejecutamos el instalador como se muestra en la imagen 19.

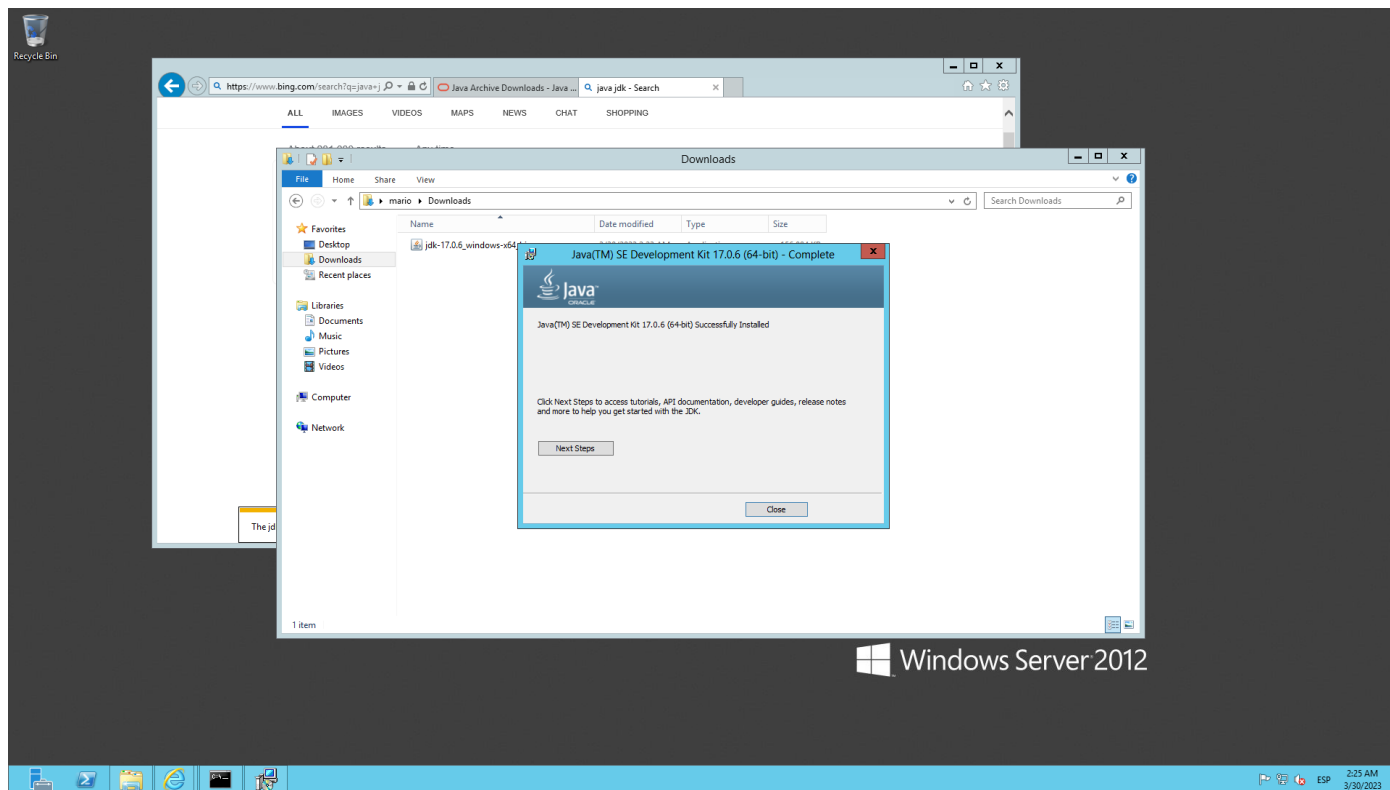


imagen 19

Agregamos la variable al path para poder ejecutar comandos en consola, esto lo hacemos abriendo el explorador de archivos, seleccionar el equipo con click derecho y seleccionar propiedades.

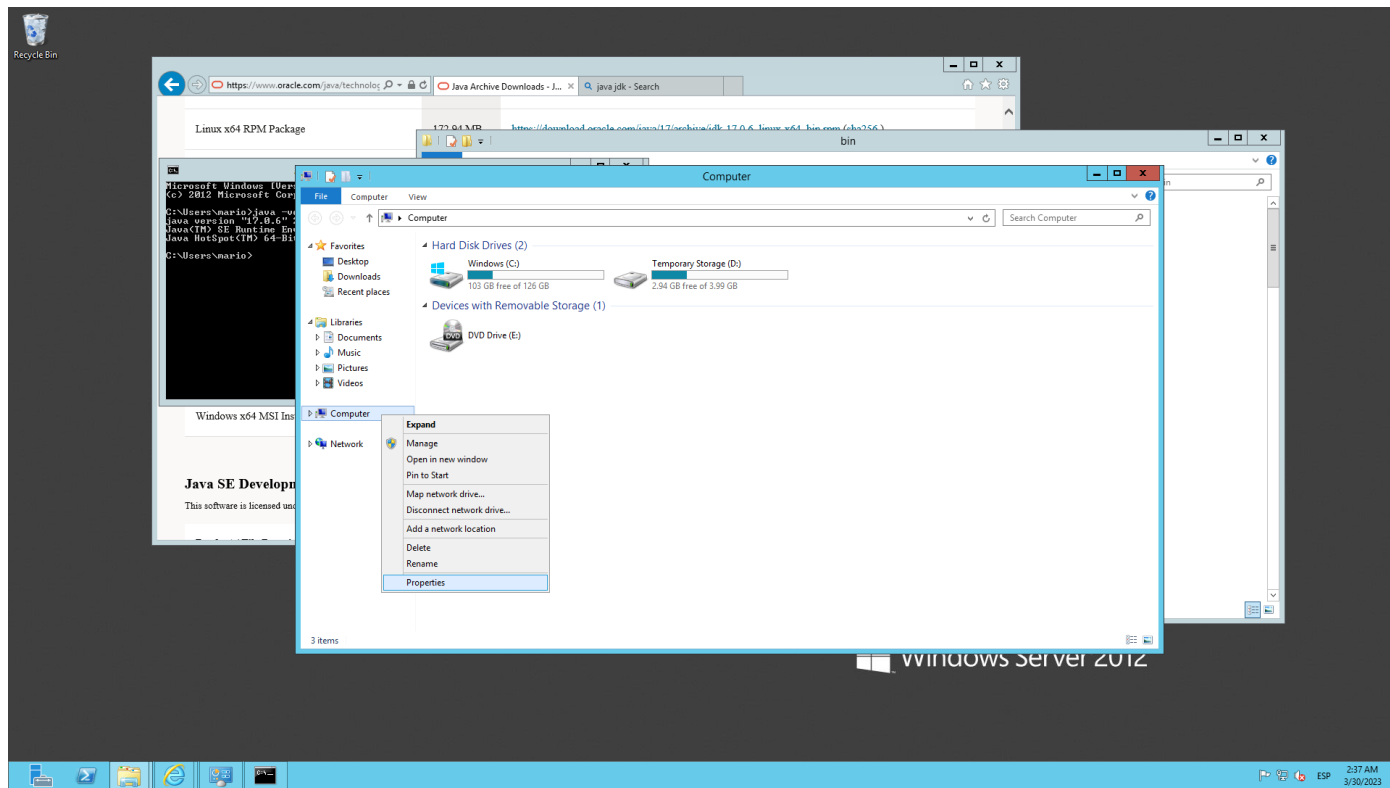


imagen 20

Luego seleccionamos configuración avanzada.

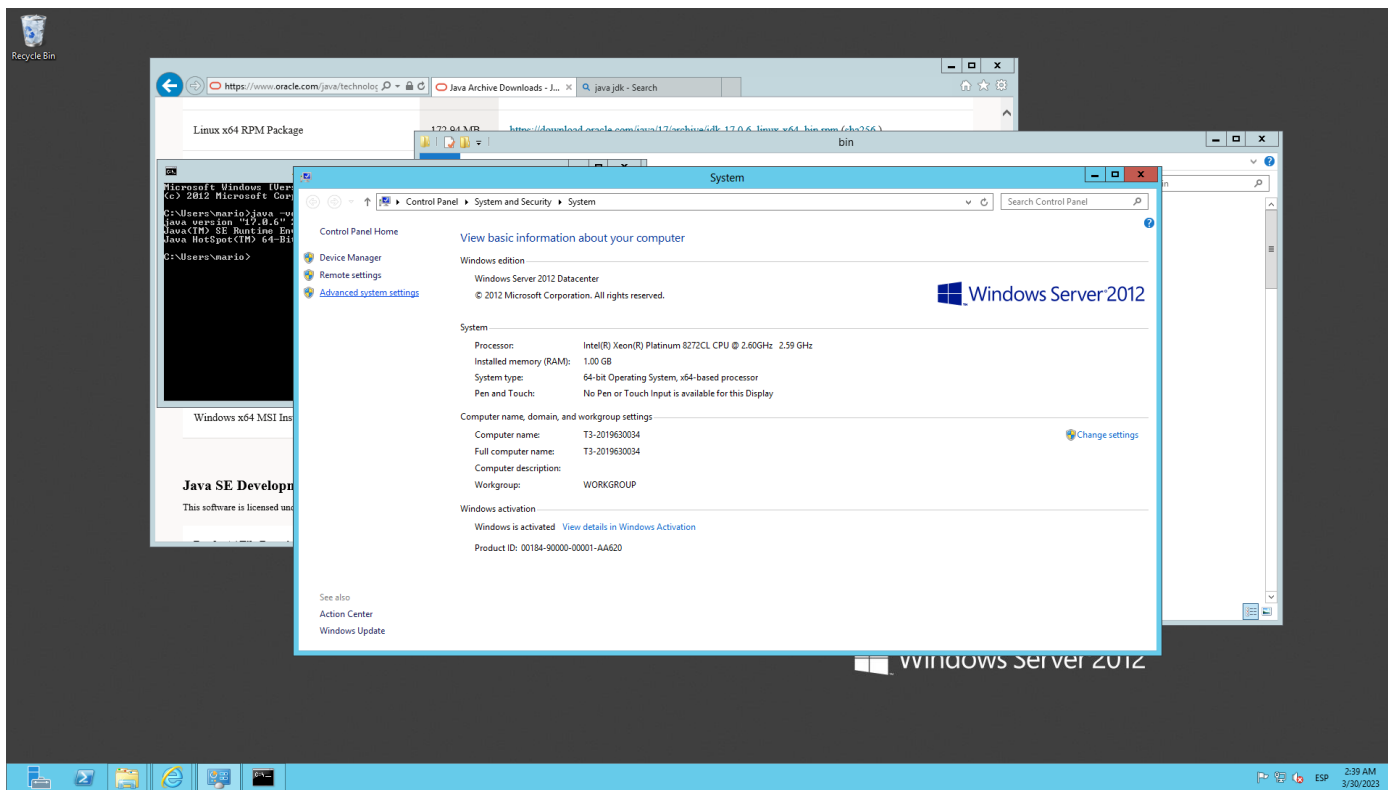


imagen 21

Luego seleccionamos variables de entorno.

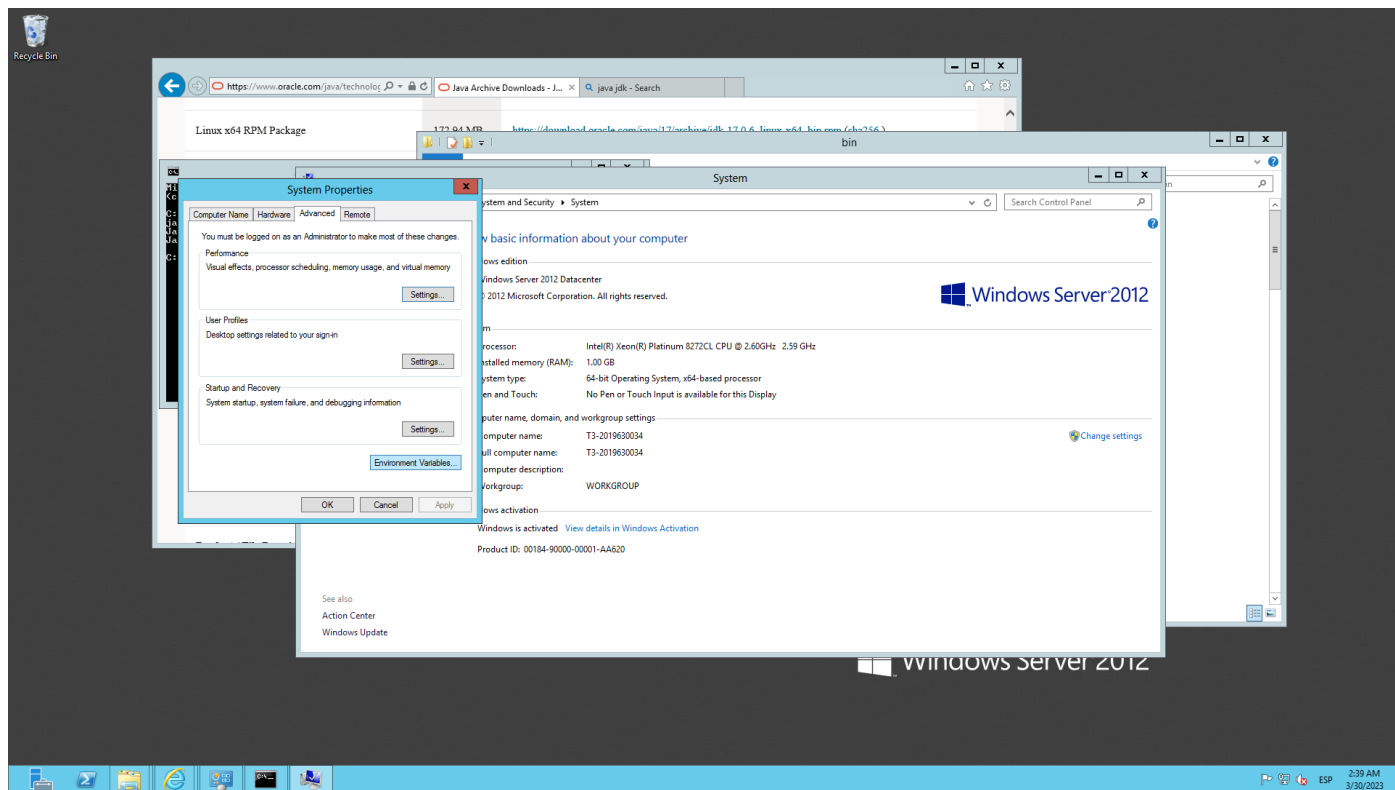


imagen 22

Finalmente seleccionamos variable de entorno y agregamos al path la ruta en donde se instaló el JDK.

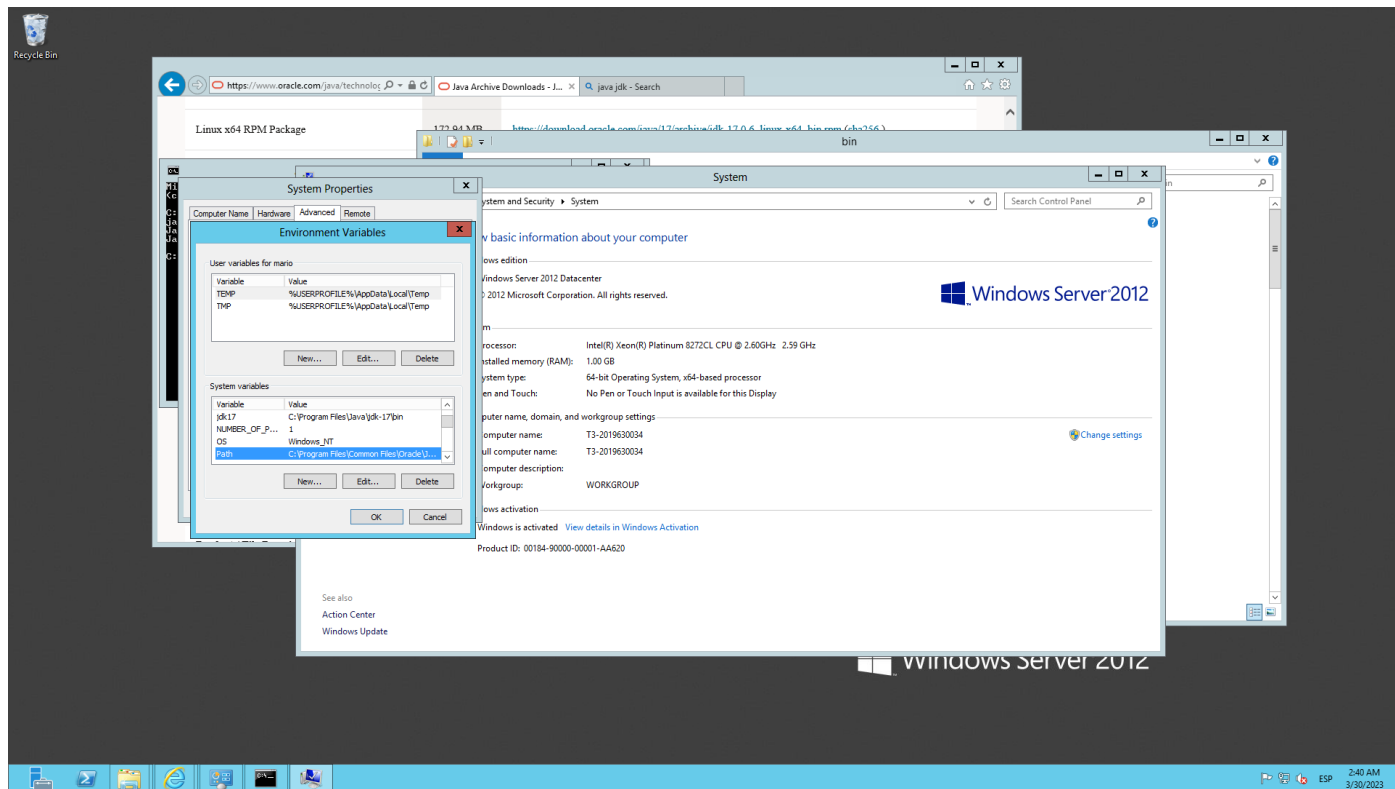


imagen 23

Comprobamos que se puedan correr comandos de java en terminal como se muestra en la imagen 24.

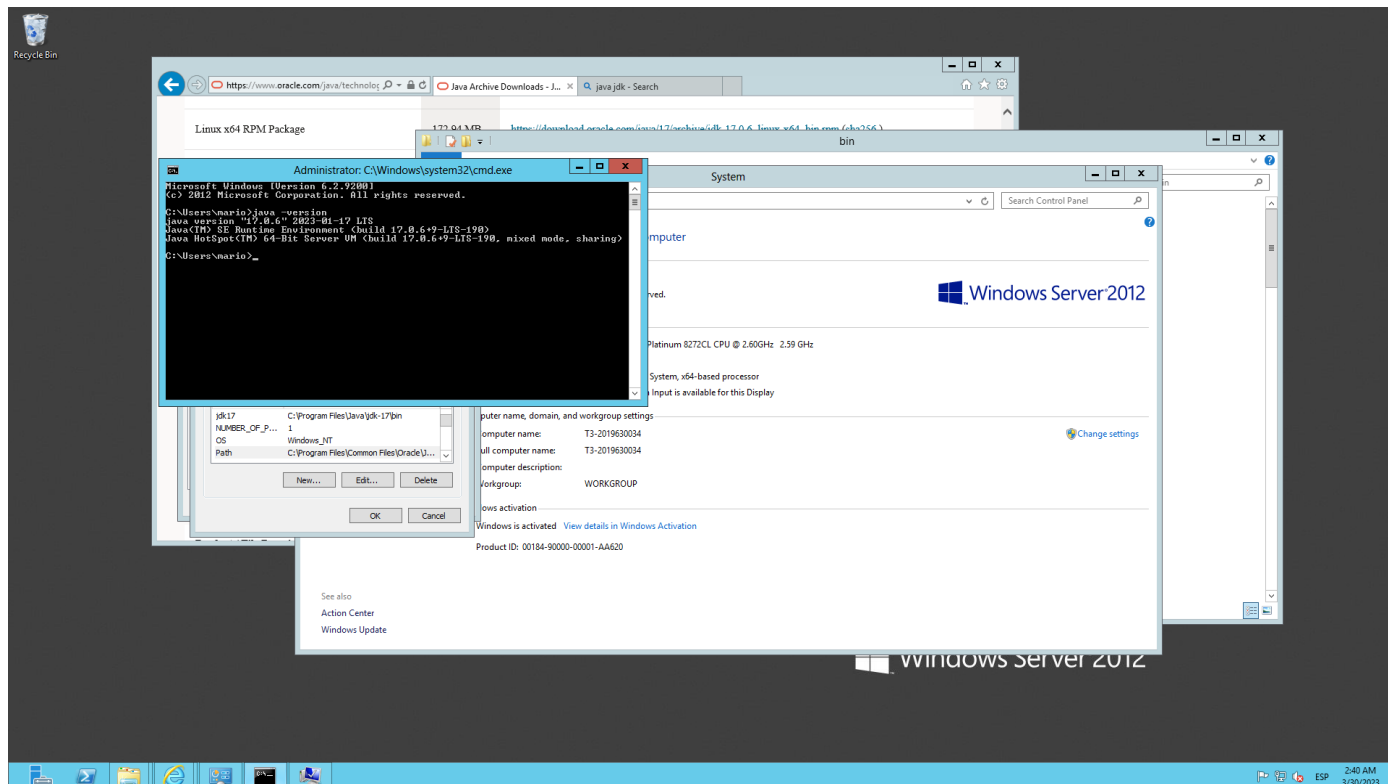


imagen 24

Ya tenemos configurada la VM para ejecutar los programas

Para correr el programa copiamos el código en un notepad en la máquina local y lo pegamos en el notepad de la VM y lo guardamos con extensión .java, posteriormente lo compilamos en la ubicación para generar el .class



## Ejecución del programa

Para esta parte corremos 4 terminales pasando como argumento el nombre de usuario, la prueba se hará conforme a lo especificado en la tarea 3.

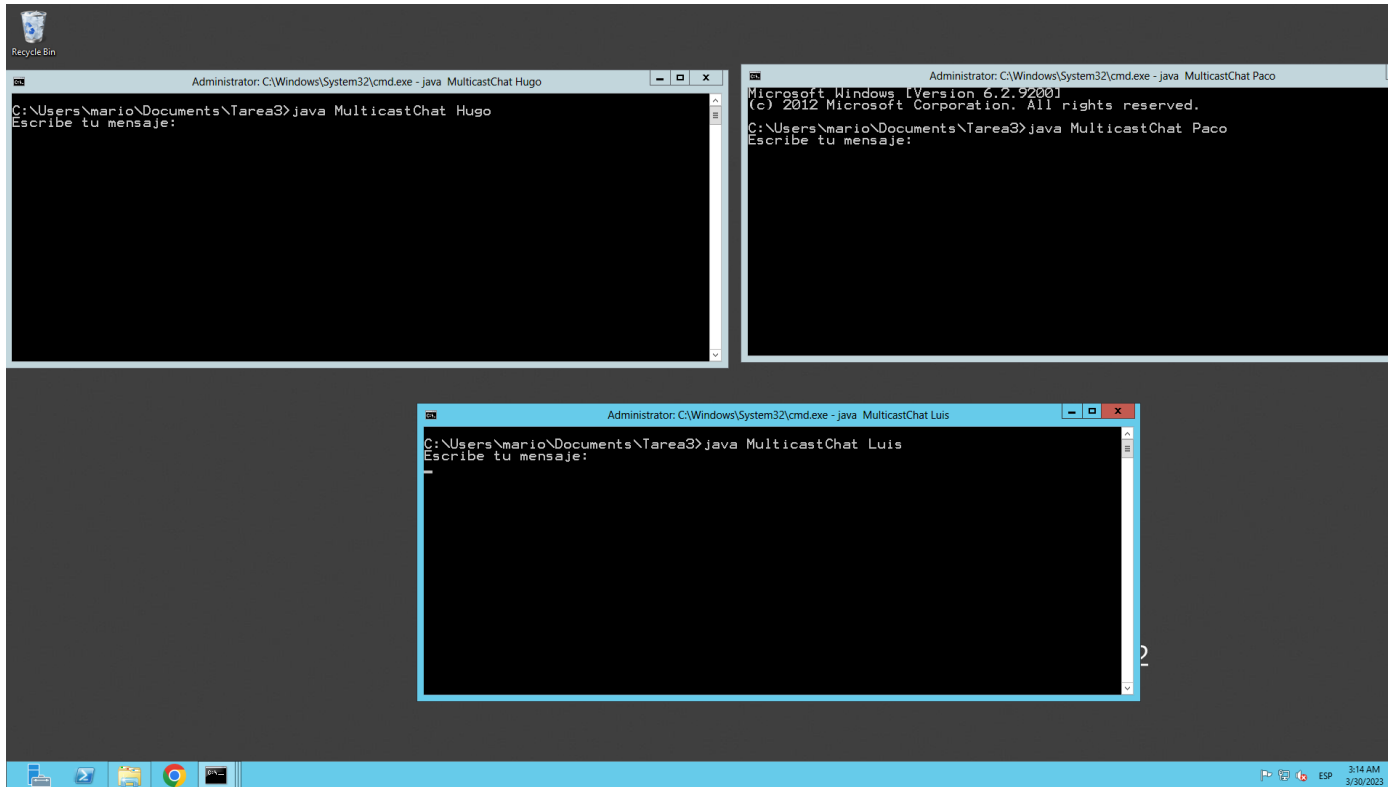


imagen 25

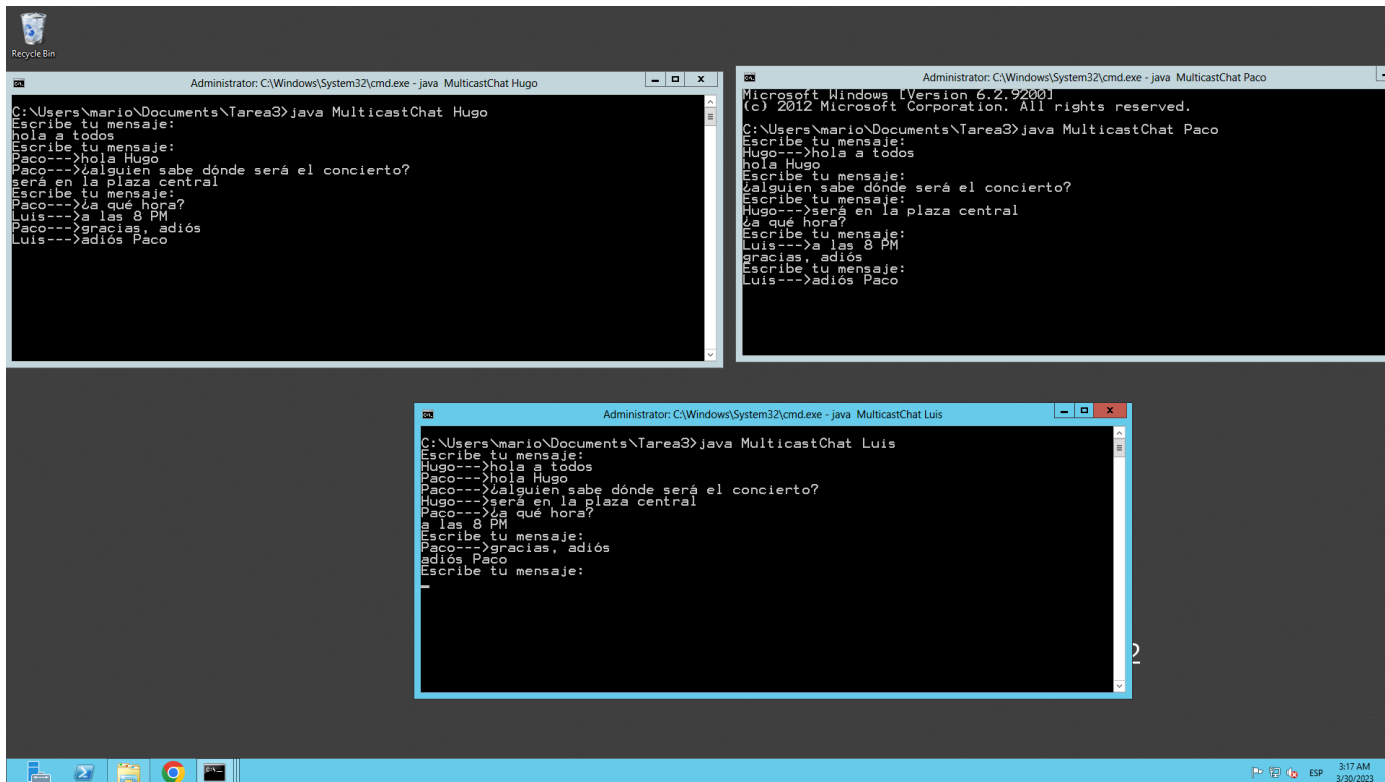


imagen 26

## Conversación sostenida con ChatGPT

**Mario Sebastian**

hola

**ChatGPT**

¡Hola! ¿En qué puedo ayudarte hoy?

**Mario Sebastian**

por favor haz un programa en java que haga lo siguiente: El programa creará un thread 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: "Escribe tu mensaje: " (sin las comillas), entonces se leerá del teclado el mensaje.

Se deberá enviar el mensaje a los nodos que pertenecen al grupo identificado por la IP 239.0.0.0 a través del puerto 50000. El mensaje a enviar en el datagrama deberá tener la siguiente forma:

`nombre_usuario--->mensaje_ingresado`

Dónde `nombre_usuario` es el nombre del usuario que pasó como parámetro al programa, `--->` es un separador y `mensaje_ingresado` es el mensaje que el usuario ingresó por el teclado.

Debido a que el protocolo UDP no garantiza el orden en que se reciben los mensajes, no se deberá enviar la longitud del mensaje y luego el mensaje; se deberá utilizar una longitud de mensaje fija. Al desplegar el mensaje recibido se deberán eliminar los espacios adicionales a la derecha (si los hay).

## ChatGPT

Por supuesto, aquí te dejo el código en Java que cumple con los requisitos que mencionaste:

```
import java.io.IOException;
```

```
import java.net.*;
```

```
public class MulticastChat {
```

```
    private static final int MAX_MESSAGE_LEN = 100; // longitud máxima  
del mensaje
```

```
    private String username;
```

```
    private InetAddress groupAddress;
```

```
    private MulticastSocket socket;
```

```
private Thread receiverThread;

public MulticastChat(String username) {
    this.username = username;

    try {
        this.groupAddress = InetAddress.getByName("239.0.0.0");
        this.socket = new MulticastSocket(50000);
        this.socket.joinGroup(this.groupAddress);
    } catch (IOException e) {
        e.printStackTrace();
    }

    // Thread para recibir mensajes
    this.receiverThread = new Thread(() -> {
        while (true) {
            byte[] buffer = new byte[MAX_MESSAGE_LEN];
            DatagramPacket packet = new DatagramPacket(buffer,
buffer.length);

            try {
                socket.receive(packet);

                String received = new String(packet.getData()).trim();
                String[] parts = received.split("---->");

                // Despliega el mensaje solo si no lo envió el usuario actual
```

```
        if (!parts[0].equals(username)) {  
            System.out.println(received);  
        }  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}  
});
```

```
    this.receiverThread.start();  
}
```

```
public void send(String message) {  
    String formattedMessage = username + "--->" + message;  
    byte[] buffer = formattedMessage.getBytes();  
  
    try {  
        DatagramPacket packet = new DatagramPacket(buffer,  
buffer.length, groupAddress, 50000);  
        socket.send(packet);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

```
public static void main(String[] args) {
```

```
    if (args.length != 1) {  
        System.out.println("Se debe especificar el nombre de usuario  
como argumento");  
        System.exit(1);  
    }
```

```
    MulticastChat chat = new MulticastChat(args[0]);
```

```
    while (true) {  
        System.out.print("Escribe tu mensaje: ");  
        String message = System.console().readLine().trim();  
        chat.send(message);  
    }  
}
```

Para probar el programa, simplemente compila el código y ejecútalo con el nombre de usuario que quieras utilizar como argumento:

```
javac MulticastChat.java
```

```
java MulticastChat <nombre_de_usuario>
```

El programa solicitará un mensaje por cada iteración del ciclo while y lo enviará a todos los nodos en el grupo identificado por la dirección IP 239.0.0.0 y el puerto 50000. Los mensajes recibidos serán desplegados en la pantalla, excepto los que fueron enviados por el mismo nodo.

## **Conclusión**

Esta práctica fue más rápida a diferencia de la práctica 2 debido a que con un solo prompt salió bien el código y no se tuvo que modular nada, he notado que si haces prompts puntuales y específicos puedes obtener una respuesta acertada y con poca variación de lo que se pide. Ahora en azure se creó una VM con windows server 2012 y cambió la forma en que mandé los archivos a la VM, además es interesante saber el principio de los chats con esta práctica que se realizó.