



INSTITUTO POLITÉCNICO NACIONAL



"ESCUELA SUPERIOR DE CÓMPUTO"

Integrantes:

- **Velasco Huerta Ángel Eduardo**

Práctica No. 3

Chat Multicast

Materia:

Desarrollo de sistemas distribuidos

Profesor:

Pineda Guerrero Carlos

Grupo:

4CV11

Desarrollo

Creación de máquina virtual

Máquina Virtual:

Crear una máquina virtual - Micr...New chatSISDI5-4CV11: Tarea 3. Chat multi...Recibidos (22.916) - angeleduar... (1) WhatsAppCopia de _Práctica 6 DM_ CRISP +

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

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

avelascoh1600@alumn...INSTITUTO POLITECNICO NACIO...

Inicio > Todos los recursos > Crear un recurso >

Crear una máquina virtual

Suscripción * Azure for Students

Grupo de recursos * (Nuevo) Práctica3

Crear nuevo

Detalles de instancia

Nombre de máquina virtual * T3-2020630482

Región * (US) East US

Opciones de disponibilidad No se requiere redundancia de la infraestructura

Tipo de seguridad Estándar

Imagen * Windows Server 2012 Datacenter - x64 Gen2

Ver todas las imágenes | Configurar la generación de máquinas virtuales

Arquitectura de VM ☐ Arm64 ☒ x64

Arm64 no es compatible con la imagen seleccionada.

Ejecución de Azure Spot con descuento ☐

Tamaño * Standard_D2s_v3 - 2 vcpu, 8 GiB de memoria (\$137.24/mes)

Ver todos los tamaños

Cuenta de administrador

Nombre de usuario * windows

Contraseña *

Confirmar contraseña *

Reglas de puerto de entrada

Revisar y crear

< Anterior

Siguiente: Discos >

Enviar comentarios

paso 1 asignamos nombre, grupo de recursos, S.O., disco y recursos (procesadores y RAM)

Crear una máquina virtual - Micr...New chatSISDI5-4CV11: Tarea 3. Chat multi...Recibidos (22.916) - angeleduar... (1) WhatsAppCopia de _Práctica 6 DM_ CRISP +

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

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

avelascoh1600@alumn...INSTITUTO POLITECNICO NACIO...

Inicio > Todos los recursos > Crear un recurso >

Crear una máquina virtual

Ver todas las imágenes | Configurar la generación de máquinas virtuales

Arquitectura de VM ☐ Arm64 ☒ x64

Arm64 no es compatible con la imagen seleccionada.

Ejecución de Azure Spot con descuento ☐

Tamaño * Standard_D2s_v3 - 2 vcpu, 8 GiB de memoria (\$137.24/mes)

Ver todos los tamaños

Cuenta de administrador

Nombre de usuario * windows

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.

Puertos de entrada públicos * ☐ Ninguno ☒ Permitir los puertos seleccionados

Seleccionar puertos de entrada * RDP (3389)

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.

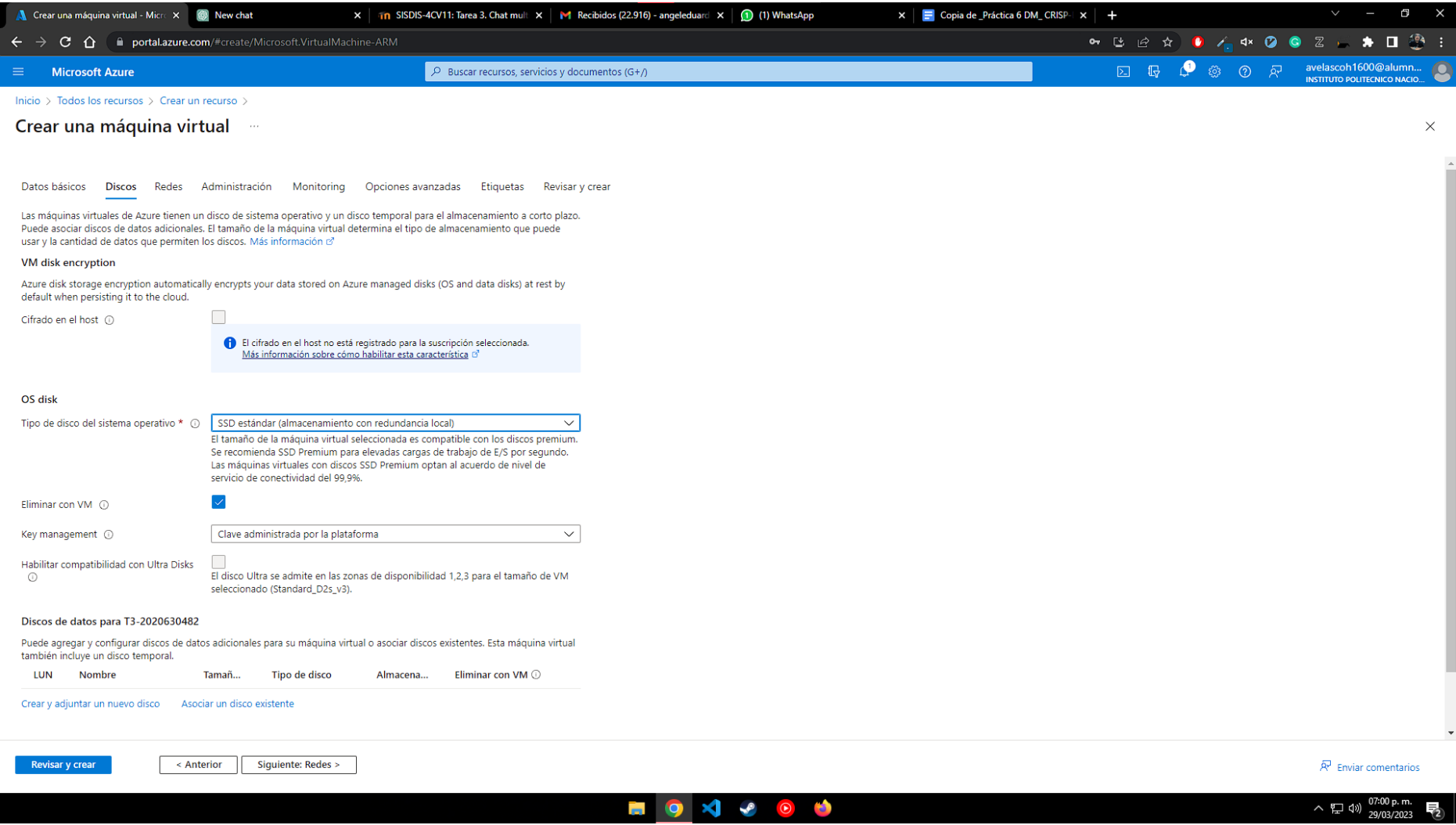
Revisar y crear

< Anterior

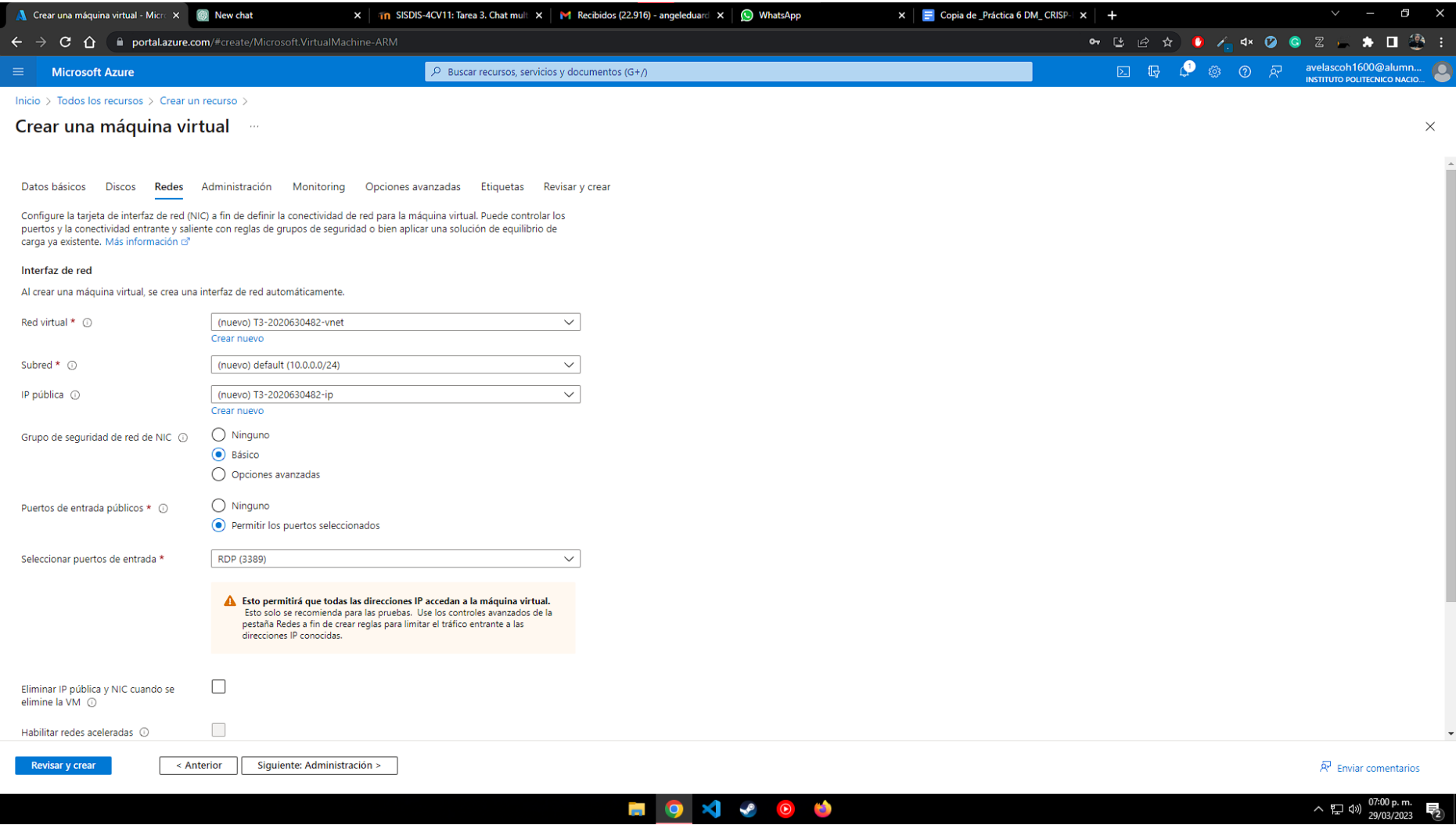
Siguiente: Discos >

Enviar comentarios

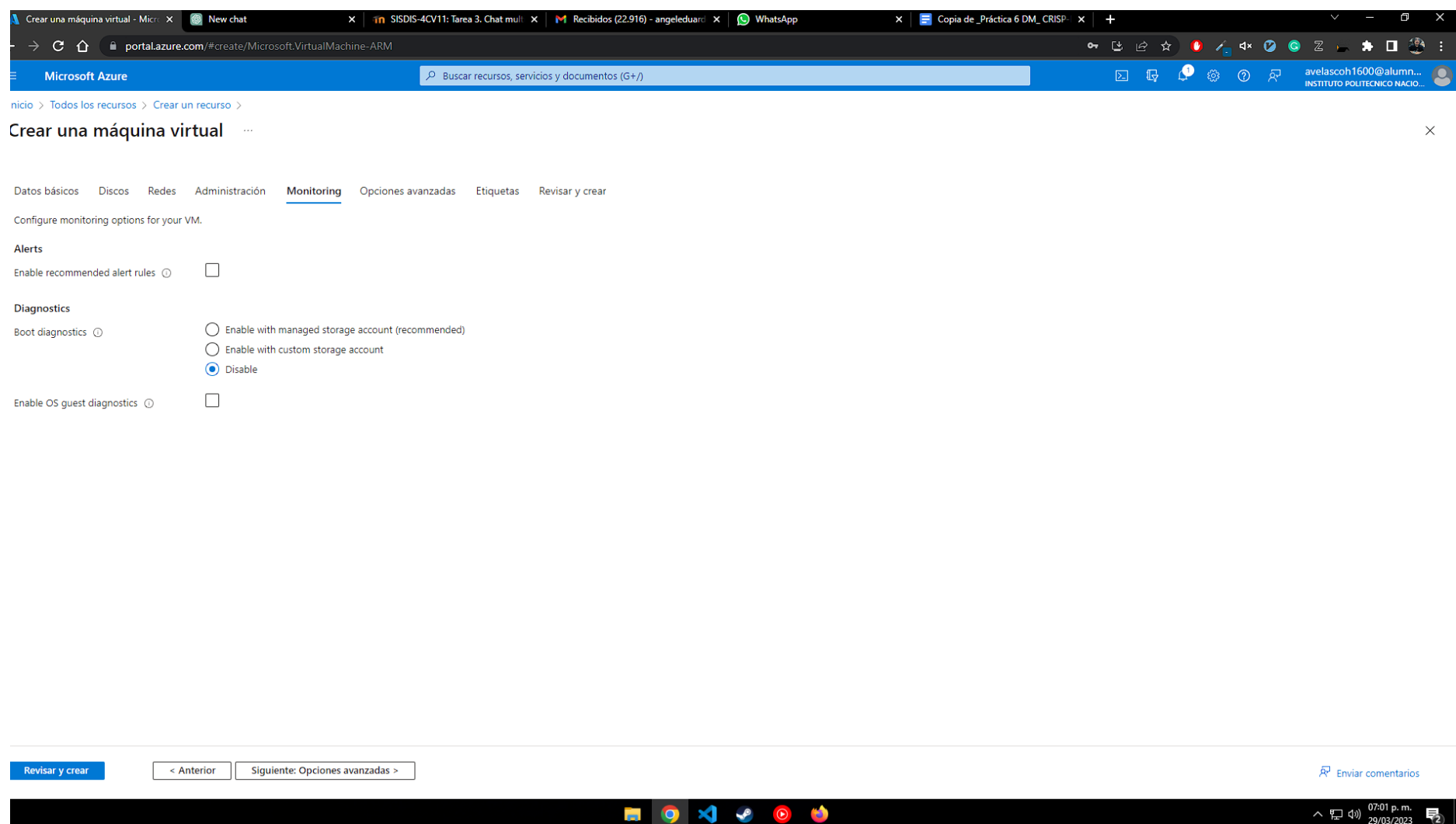
paso 2 asignamos contraseña y usuario así como el puerto RDP



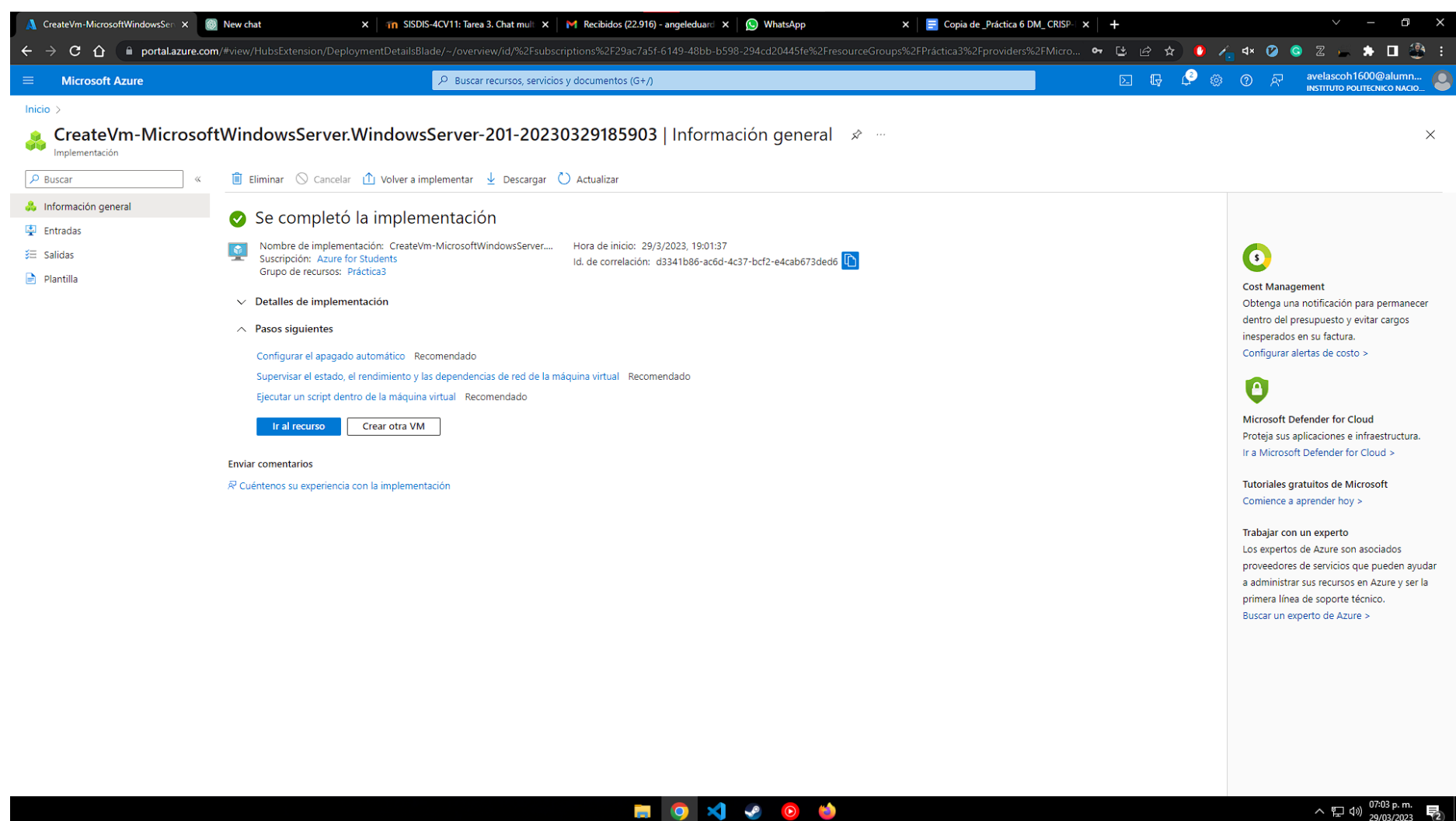
paso 3, seleccionamos un Disco apropiado para la práctica.



paso 4, definios la red virtual, y nos aseguramos de que si hay multiples máquinas, pertenezcan a la misma red



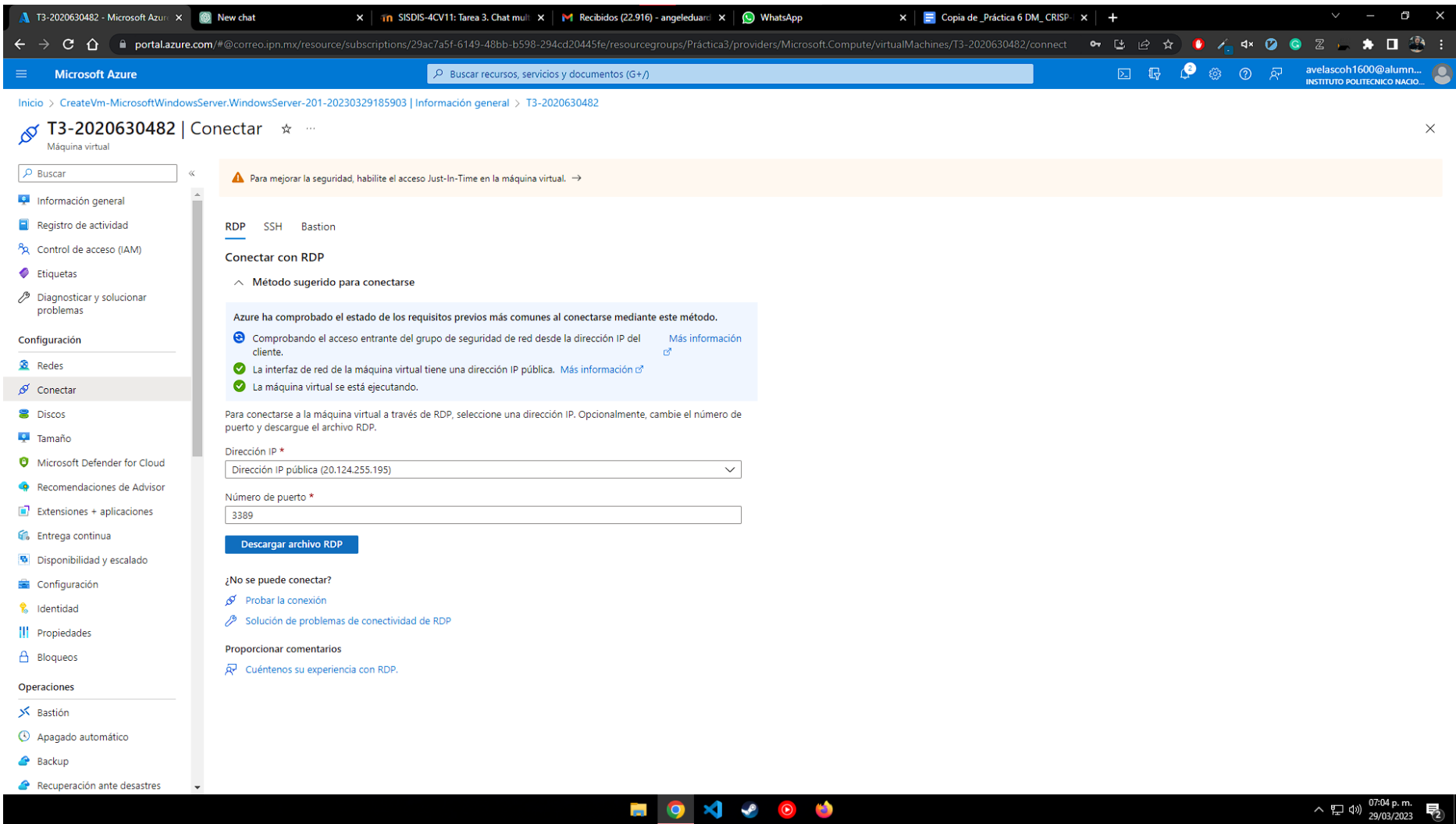
paso 5, deshabilitamos el diagnostico de arranque para hacer más rápida su ejecución.



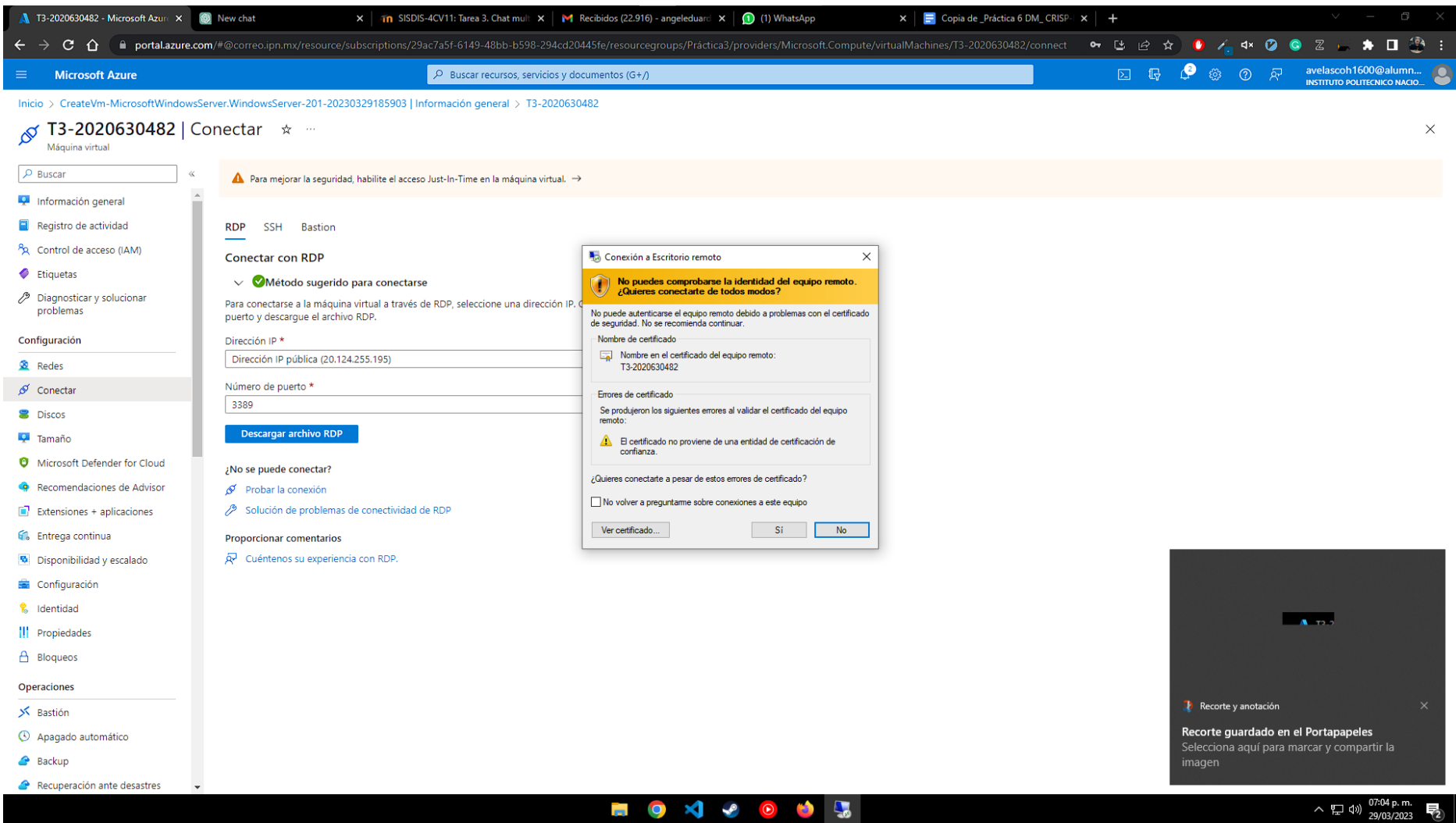
paso 6 Finalizamos la creación de la máquina virtual, y creamos el recurso

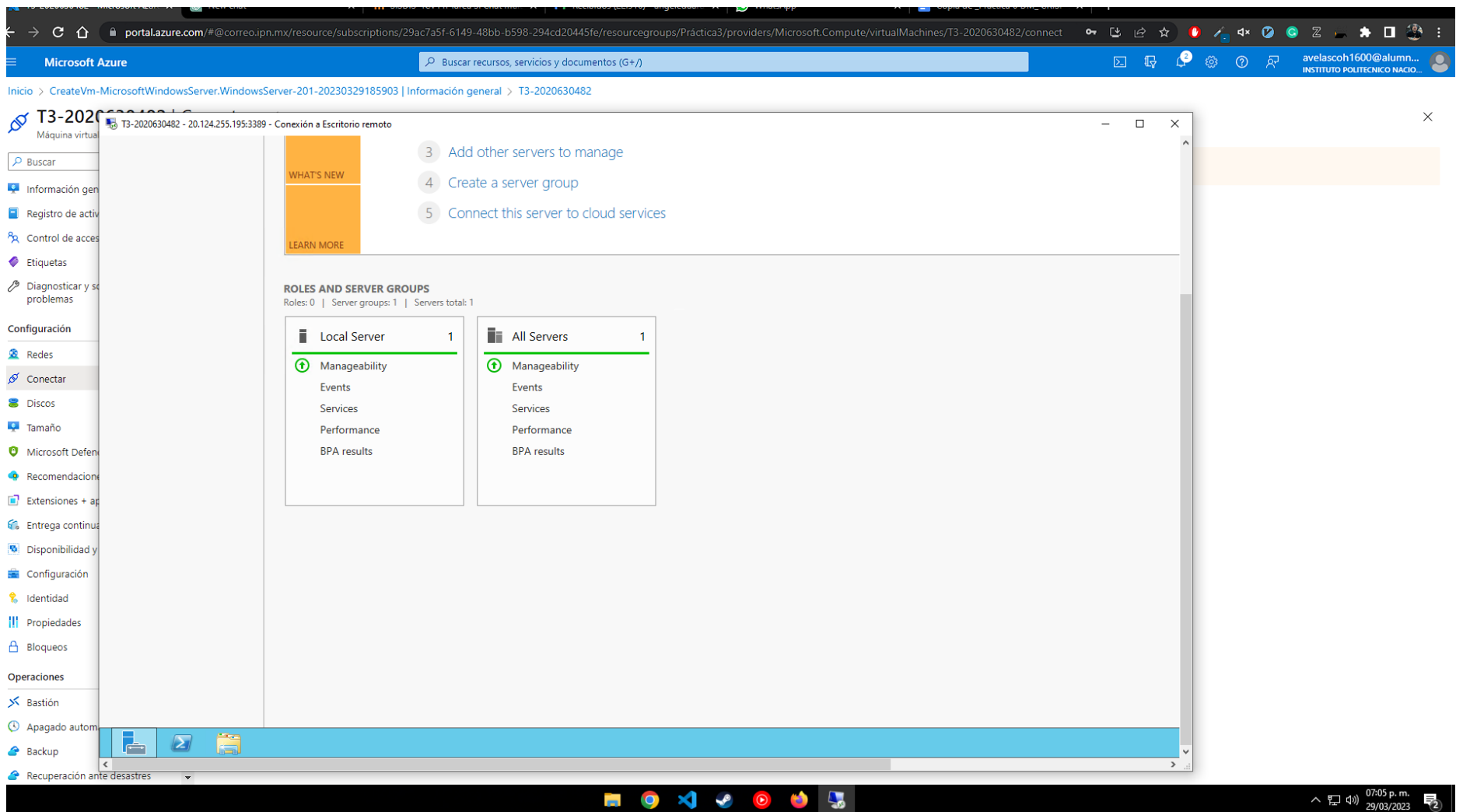
Conexión RDP

Para conectarnos por RDP, basta con descargar el archivo correspondiente al dar clic en conectar → RDP.



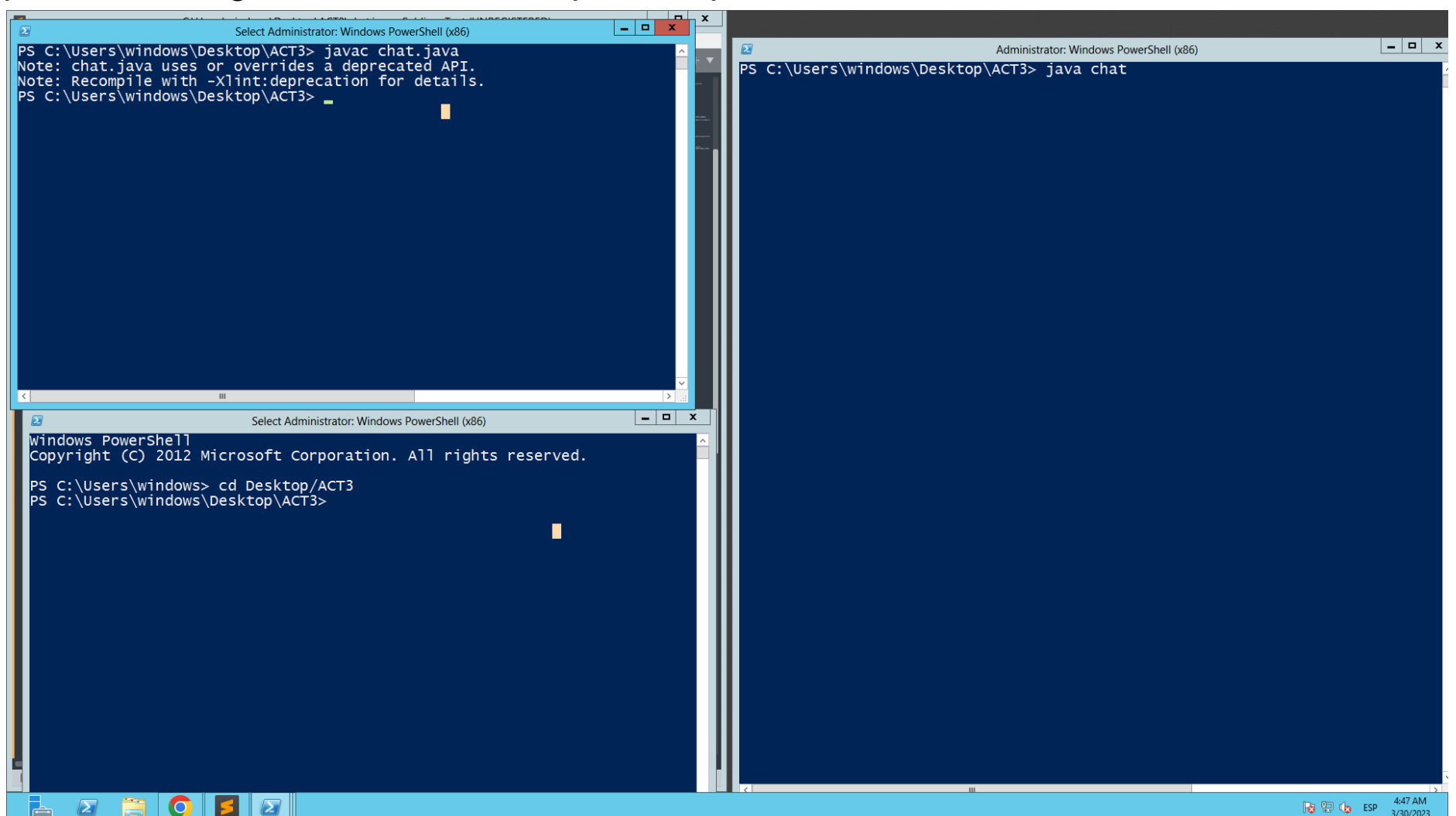
Y nos conectaremos abriéndolo:



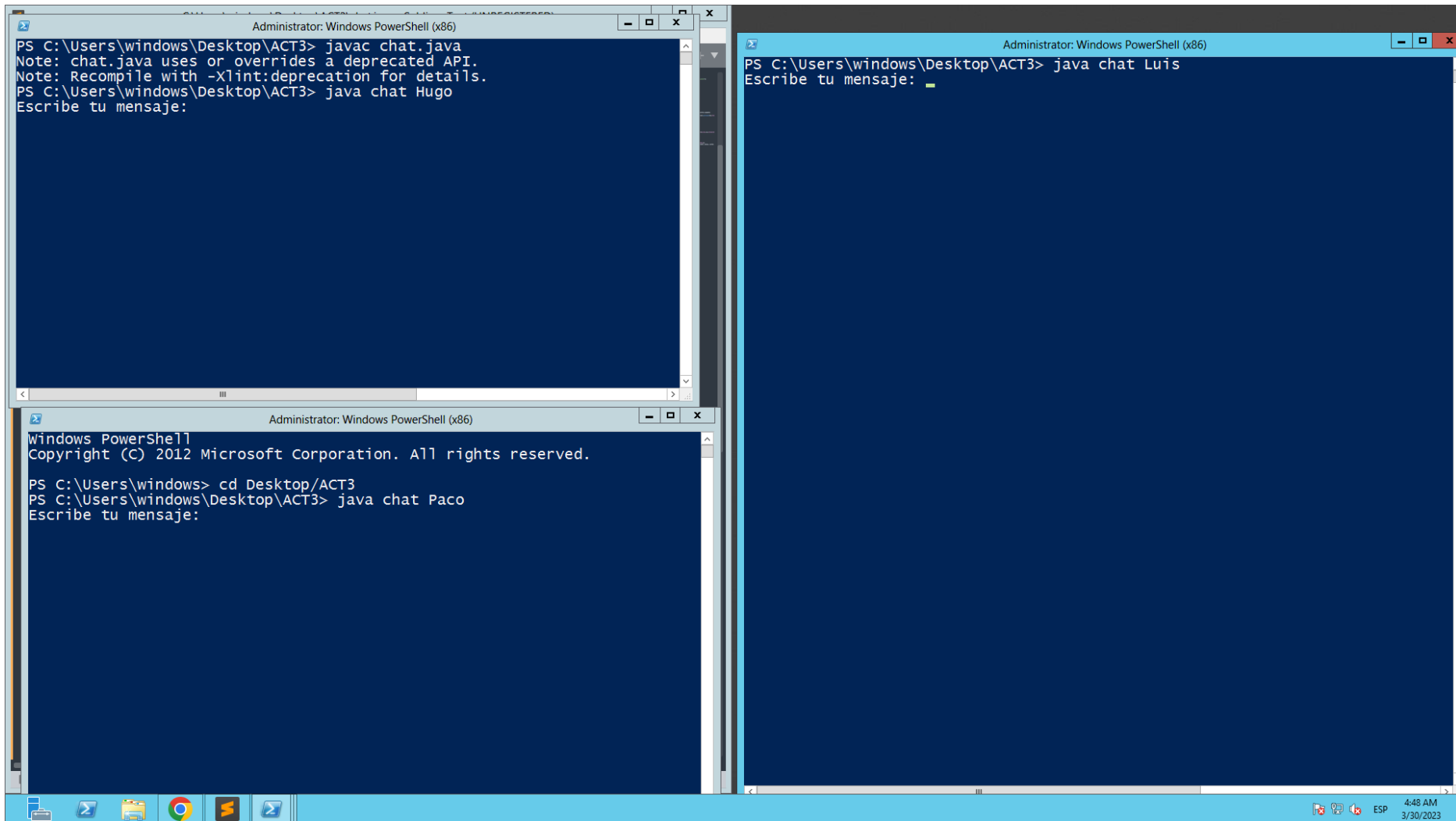


Ya estamos en la máquina virtual de Azure, podemos comenzar a trabajar en ellas.

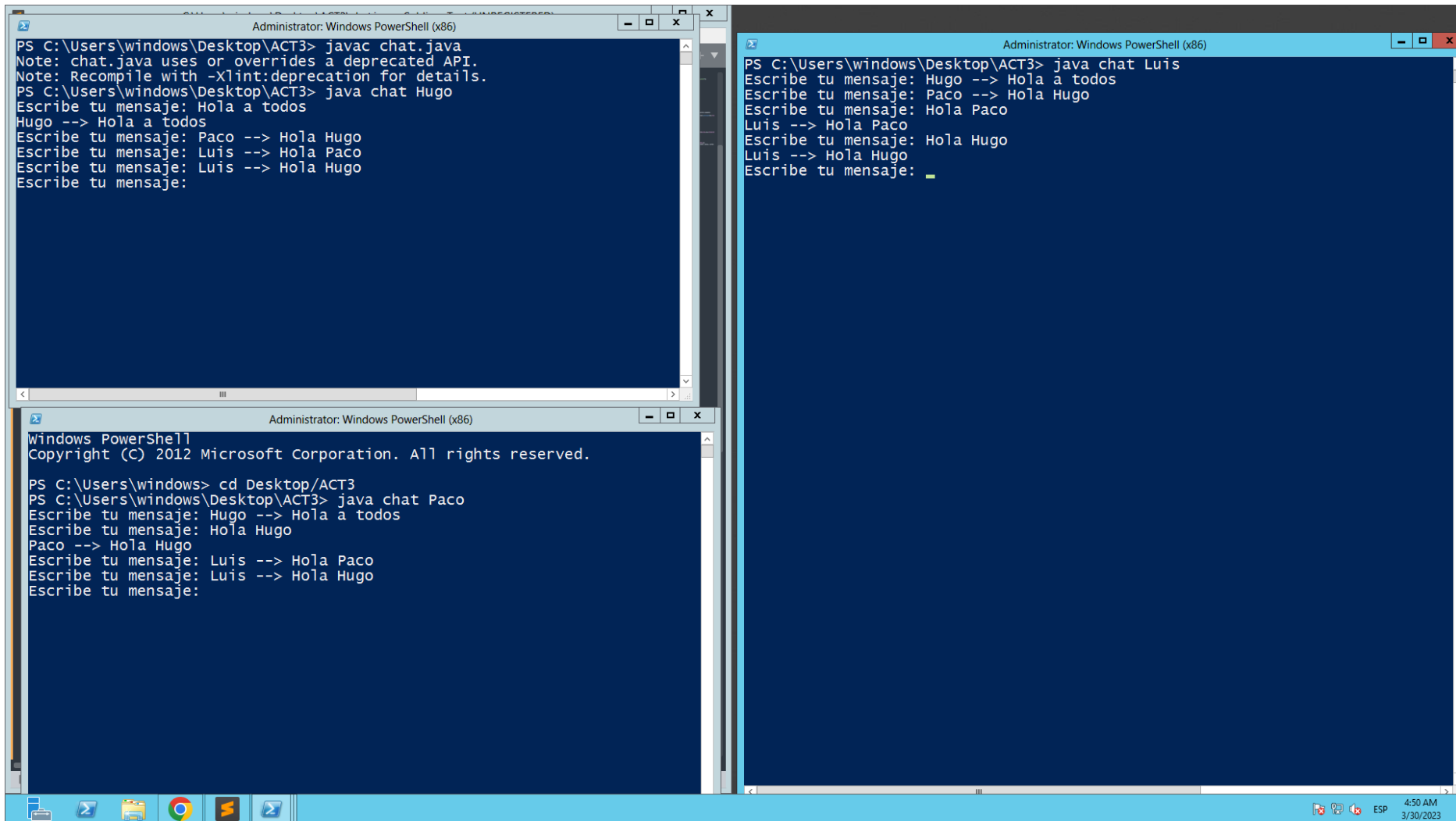
Una vez cargado el programa en la máquina virtual, abrimos una consola de CMD para probar el código, entonces, lo compilamos primero



Ahora, procedemos a iniciar el chat de cada consola con su respectivo usuario.



Ya conectados, iniciamos la conversación:



Como podemos ver, la comunicación se está cumpliendo, seguimos con la conversación:


```
PS C:\Users\windows\Desktop\ACT3> java chat Hugo
Escribe tu mensaje: Hola a todos
Hugo --> Hola a todos
Escribe tu mensaje: Paco --> Hola Hugo
Escribe tu mensaje: Luis --> Hola Paco
Escribe tu mensaje: Luis --> Hola Hugo
Escribe tu mensaje: Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje:

PS C:\Users\windows\Desktop\ACT3> java chat Luis
Escribe tu mensaje: Hugo --> Hola a todos
Escribe tu mensaje: Paco --> Hola Hugo
Escribe tu mensaje: Hola Paco
Luis --> Hola Paco
Escribe tu mensaje: Hola Hugo
Luis --> Hola Hugo
Escribe tu mensaje: Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje:

PS C:\Users\windows\Desktop\ACT3> java chat Paco
Escribe tu mensaje: Hugo --> Hola a todos
Escribe tu mensaje: Hola Hugo
Paco --> Hola Hugo
Escribe tu mensaje: Luis --> Hola Paco
Escribe tu mensaje: Luis --> Hola Hugo
Escribe tu mensaje: ¿Alguien sabe dónde será el concierto?
Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje:
```

Como podemos ver, al escribir con tildes y signos propios del español, se muestran correctamente en la consola, importante aclarar que para que esto sirva correctamente, se debe de primero escribir chcp 1252 en la consola, pues eso permite que se muestre correctamente con caracteres de ISO.

Seguimos:

```
PS C:\Users\windows\Desktop\ACT3> java chat Hugo
Escribe tu mensaje: Hola a todos
Hugo --> Hola a todos
Escribe tu mensaje: Paco --> Hola Hugo
Escribe tu mensaje: Luis --> Hola Paco
Escribe tu mensaje: Luis --> Hola Hugo
Escribe tu mensaje: Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje: Será en la Plaza Central
Hugo --> Será en la Plaza Central
Escribe tu mensaje: Paco --> ¿A qué hora?
Escribe tu mensaje: Luis --> A las 8 PM
Escribe tu mensaje: Paco --> Gracias, adiós.
Escribe tu mensaje: Luis --> adiós Paco
Escribe tu mensaje:

PS C:\Users\windows\Desktop\ACT3> java chat Luis
Escribe tu mensaje: Hugo --> Hola a todos
Escribe tu mensaje: Paco --> Hola Hugo
Escribe tu mensaje: Hola Paco
Luis --> Hola Paco
Escribe tu mensaje: Hola Hugo
Luis --> Hola Hugo
Escribe tu mensaje: Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje: Hugo --> Será en la Plaza Central
Escribe tu mensaje: Paco --> ¿A qué hora?
Escribe tu mensaje: A las 8 PM
Luis --> A las 8 PM
Escribe tu mensaje: Paco --> Gracias, adiós.
Escribe tu mensaje: adiós Paco
Luis --> adiós Paco
Escribe tu mensaje:

PS C:\Users\windows\Desktop\ACT3> java chat Paco
Escribe tu mensaje: Hugo --> Hola a todos
Escribe tu mensaje: Hola Hugo
Paco --> Hola Hugo
Escribe tu mensaje: Luis --> Hola Paco
Escribe tu mensaje: Luis --> Hola Hugo
Escribe tu mensaje: ¿Alguien sabe dónde será el concierto?
Paco --> ¿Alguien sabe dónde será el concierto?
Escribe tu mensaje: Hugo --> Será en la Plaza Central
Escribe tu mensaje: ¿A qué hora?
Paco --> ¿A qué hora?
Escribe tu mensaje: Luis --> A las 8 PM
Escribe tu mensaje: Gracias, adiós.
Paco --> Gracias, adiós.
Escribe tu mensaje: Luis --> adiós Paco
Escribe tu mensaje:
```

Y así finaliza la conversación entre Hugo, Paco y Luis, corta pero suficiente para demostrar las características del programa solicitado.

Conversación con CHAT GPT 3.5

En este caso, el chat fue capaz de concluir los códigos solicitados, por lo que el programa se le pidió como tal a chatGPT

Angel:

Hola chat, por favor, crea un chat multicast en java, el chat será ejecutado en el cmd, así que hay que asegurarse que acentos y caracteres de UNICODE o UTF8 sean perfectamente visibles, como parametro al programa se le debe de pasar el nombre de usuario del chat, necesito que solo cuando se vaya a escribir un mensaje, se despliegue el prompt "Escribe tu mensaje" y que cuando se reciba un mensaje, se vea con el siguiente formato: Nombre de usuario --> Mensaje.

CHAT:

Para crear un chat multicast en Java, podemos utilizar la clase MulticastSocket y DatagramPacket para enviar y recibir mensajes multicast. Aquí está el código:

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

public class MulticastChat {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Escriba su nombre de usuario, java chat [usuario]");
            return;
        }

        String username = args[0];
        int port = 12345;
        InetAddress group;
        try {
            group = InetAddress.getByName("230.0.0.0");
        } catch (UnknownHostException e) {
            e.printStackTrace();
            return;
        }

        try {
            MulticastSocket socket = new MulticastSocket(port);
            socket.joinGroup(group);

            Thread receiveThread = new Thread(() -> {
                while (true) {
                    try {
                        byte[] buffer = new byte[1024];
                        DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
                        socket.receive(packet);
```

```

        String message = new String(packet.getData(), 0,
packet.getLength(), "UTF-8");

        System.out.println(message);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

});
receiveThread.start();

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Bienvenido al chat. Escribe tu mensaje:");
while (true) {
    String line = reader.readLine();
    if (line == null) break;
    String fullMessage = username + " --> " + line;
    byte[] buffer = fullMessage.getBytes("UTF-8");
    DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group,
port);

    socket.send(packet);
}

socket.leaveGroup(group);
socket.close();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

Angel:

Sobre ese mismo código necesito algunas correcciones, la codificación necesito que sea ISO, pues se learan acentos en cmd, además necesito que el prompt de Escribe tu Mensaje: se lea cada que se termina de recibir un mensaje, para que así se vea esto antes de teclear el mensaje, además, necesito el puerto 50000 y la dirección 239.0.0.0.

CHAT:

```

import java.io.*;
import java.net.*;

public class MulticastChat {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Escriba su nombre de usuario, java chat [usuario]");
            return;
        }

        String username = args[0];
        int port = 50000;
        InetAddress group;
    }
}

```

```

    try {
        group = InetAddress.getByName("239.0.0.0");
    } catch (UnknownHostException e) {
        e.printStackTrace();
        return;
    }

    try {
        MulticastSocket socket = new MulticastSocket(port);
        socket.joinGroup(group);

        Thread receiveThread = new Thread(() -> {
            while (true) {
                try {
                    byte[] buffer = new byte[1024];
                    DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
                    socket.receive(packet);
                    String message = new String(packet.getData(), 0,
packet.getLength(), "ISO-8859-1");
                    System.out.println(message);
                    System.out.print("Escribe tu mensaje: ");
                    System.out.flush();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
        receiveThread.start();

        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Escribe tu mensaje: ");
        System.out.flush();
        while (true) {
            String line = reader.readLine();
            if (line == null) break;
            String fullMessage = username + " --> " + line;
            byte[] buffer = fullMessage.getBytes("ISO-8859-1");
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group,
port);

            socket.send(packet);
            System.out.print("Escribe tu mensaje: ");
            System.out.flush();
        }

        socket.leaveGroup(group);
        socket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Aquí están las modificaciones que se hicieron:

- Se cambió la codificación de UTF-8 a ISO-8859-1 para que se puedan leer los acentos en la consola de comandos.
- Se agregó una llamada a `System.out.flush()` después de cada `System.out.print()` para asegurarse de que se escriba inmediatamente en la consola.
- Se movió la línea `System.out.print("Escribe tu mensaje: ");` dentro del hilo que recibe mensajes para que se muestre después de cada mensaje recibido.

Con estas correcciones, el programa debería mostrar el prompt "Escribe tu mensaje:" después de cada mensaje recibido y permitir el uso de acentos y otros caracteres ISO en la consola de comandos.

ANGEL:

Me puedes dar alguna forma de garantizar que los caracteres ISO se visualicen en la consola de Windows?

CHAT:

Sí, hay varias formas de garantizar que los caracteres ISO se visualicen correctamente en la consola de Windows:

1. Cambiar la página de códigos de la consola: La consola de Windows utiliza una página de códigos para determinar cómo se muestran los caracteres. Para garantizar que los caracteres ISO se visualicen correctamente, debes cambiar la página de códigos de la consola a una que soporte esos caracteres. Puedes hacerlo ejecutando el siguiente comando en la consola de Windows:

```
chcp 1252
```

Este comando establece la página de códigos en "Windows 1252", que es una página de códigos que soporta los caracteres ISO.

2. Usar una fuente que soporte los caracteres ISO: La fuente que uses en la consola de Windows también puede afectar la visualización de los caracteres ISO. Si la fuente que estás usando no soporta esos caracteres, es posible que no se muestren correctamente. Puedes cambiar la fuente de la consola de Windows haciendo clic con el botón derecho en la barra de título de la ventana de la consola, seleccionando "Propiedades" y luego seleccionando una fuente que soporte los caracteres ISO.
3. Usar una herramienta de terceros: Si las opciones anteriores no funcionan, puedes usar una herramienta de terceros para visualizar los caracteres ISO en la consola de Windows. Por ejemplo, puedes utilizar "ConEmu" o "Cmder", que son emuladores de terminal que soportan múltiples páginas de códigos y fuentes, incluyendo aquellas que soportan los caracteres ISO.

Angel:

Creo que una posible solución a los caracteres que no se ven es usar STANDARDCHARSETS no?

CHAT:

¡Correcto! Usar la clase `StandardCharsets` de Java es otra forma de solucionar el problema de los caracteres que no se ven en la consola de comandos. Esta clase proporciona constantes para las codificaciones de caracteres más comunes, incluyendo UTF-8 e ISO-8859-1.

Aquí está el código modificado para usar `StandardCharsets` en lugar de pasar las codificaciones como cadenas de texto:

```
import java.io.*;
import java.net.*;
import java.nio.charset.StandardCharsets;
public class chat {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Escriba su nombre de usuario, java chat [usuario]");
            return;
        }
    }
}
```

```

String username = args[0];
int port = 50000;
InetAddress group;
try {
    group = InetAddress.getByName("239.0.0.0");
} catch (UnknownHostException e) {
    e.printStackTrace();
    return;
}

try {
    MulticastSocket socket = new MulticastSocket(port);
    socket.joinGroup(group);

    Thread receiveThread = new Thread(() -> {
        while (true) {
            try {
                byte[] buffer = new byte[1024];
                DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
                socket.receive(packet);
                String message = new String(packet.getData(), 0,
packet.getLength(), StandardCharsets.ISO_8859_1);
                System.out.println(message.trim());
                System.out.print("Escribe tu mensaje: ");
                System.out.flush();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    });
    receiveThread.start();

    BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in, StandardCharsets.ISO_8859_1));
    System.out.print("Escribe tu mensaje: ");
    System.out.flush();
    while (true) {
        String line = reader.readLine();
        if (line == null) break;
        String fullMessage = username + " --> " + line;
        byte[] buffer = fullMessage.getBytes(StandardCharsets.ISO_8859_1);
        DatagramPacket packet = new DatagramPacket(buffer, buffer.length, group,
port);

        socket.send(packet);

        System.out.flush();
    }

    socket.leaveGroup(group);
    socket.close();
} catch (IOException e) {
    e.printStackTrace();
}

```

```
}  
}  
}
```

ANGEL:

Gracias!

Conclusiones:

En este programa hemos aprendido acerca de cómo usar sockets de multicast en Java para implementar un chat de grupo en red. Además, hemos visto cómo manejar la entrada y salida de caracteres codificados en diferentes conjuntos de caracteres o codificaciones.

Al utilizar datagramas para el envío y recepción de mensajes, hemos aprendido que esta técnica es adecuada para aplicaciones que no requieren conexiones persistentes entre los extremos, como un chat en el que los mensajes son enviados y recibidos a medida que se producen. Los datagramas son paquetes independientes que no tienen un flujo continuo de datos, lo que los hace útiles para aplicaciones que requieren una comunicación rápida y eficiente.

También hemos visto cómo la codificación de caracteres es importante en la comunicación de datos, especialmente en ambientes multi-idioma o multi-alfabeto. Es importante tener en cuenta la codificación utilizada por el sistema y por las diferentes aplicaciones, para evitar problemas de visualización de caracteres especiales o acentos.

Por último, el uso de multicast en este programa nos ha permitido enviar mensajes a múltiples destinatarios al mismo tiempo, lo que es esencial en un chat de grupo en el que los mensajes deben ser vistos por todos los participantes.

En conclusión, el uso de datagramas, codificación de caracteres y multicast son elementos clave en la implementación de aplicaciones de red eficientes y escalables, como el chat de grupo que hemos creado en este programa. Aprender a utilizar estas técnicas y herramientas puede ser de gran utilidad para los desarrolladores que trabajan en proyectos de comunicación de red.