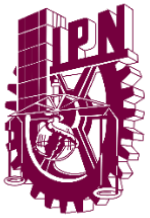


Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas distribuidos

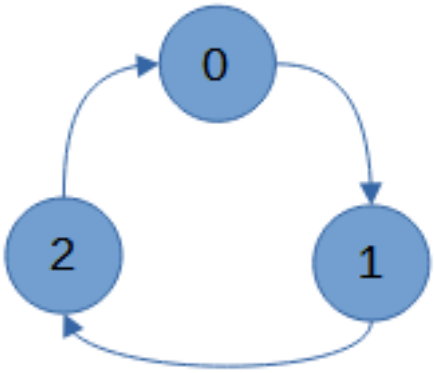
Profesor: Pineda Guerrero Carlos

Tarea 5. Chat multicast

Alumnos:
Osornio Zambrano Alberto Aacini

4CV2

Índice



Implementación de chat multicast

esarrollar 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 programa **deberá** utilizar las siguientes funciones para enviar y recibir los mensajes multicast:

```
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();
}
```

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.
- Para probar el programa, se **deberá** ejecutar la siguiente conversación en tres ventanas de comandos (cmd) en la máquina virtual con Windows 10. En la primera ventana escribirá hugo, en la segunda ventana escribirá paco y en la tercera ventana escribirá luis:
- hugo debe escribir:
  hola a todos

  paco debe escribir:
  hola hugo

  luis debe escribir:
  hola hugo

  hugo debe escribir:
  ¿alguien sabe dónde será la fiesta el sábado?

  paco debe escribir:
  será en la casa de donald

  hugo debe escribir:
  ¿a qué hora?

luis debe escribir:
a las 8 PM

hugo debe escribir:
adios

paco debe escribir:
adios hugo

luis debe escribir:
adios hugo

-
- Notar que los signos de interrogación y las letras acentuadas deberán desplegarse correctamente en la ventana de comandos de Windows.
-
- Se **deberá** subir a la plataforma un archivo texto con el código fuente del programa desarrollado y un reporte de la tarea en formato PDF con portada, desarrollo y conclusiones como mínimo. El archivo PDF deberá incluir las capturas de pantalla de la compilación y ejecución del programa, se deberá incluir la captura de pantalla correspondiente a **cada paso** de la creación de la máquina virtual. No se admitirá la tarea si no incluye las pantallas correspondientes a cada paso del procedimiento de creación de la máquina virtual.
- El nombre de la máquina virtual deberá ser el número de boleta del alumno, si el número de boleta del alumno es 12345678, entonces la máquina virtual deberá llamarse: B12345678. **No se admitirá la tarea** si la máquina virtual no se nombra como se indicó anteriormente.
- Recuerden que deben **eliminar la máquina virtual** cuando no la usen, con la finalidad de ahorrar el saldo de sus cuentas de Azure.
- No se admitirá la tarea si se envían archivos en formato RAR o en formato WORD.
- Valor de la tarea: 20% (1.2 puntos de la segunda evaluación parcial)

Desarrollo

Se crearan maquinas virtuales mediante la plataforma azure deacuerdo a las especificaciones vista en la guía de la clase, que compartan el mismo grupo de trabajo para que se le puedan cambiar políticas en conjunto.. Una ves inicializadas entraremos en ellas por medio de RDP lo cual nos dara control via remoto al equipo . Procederemos a intalarle el sdk para poder compilar y ejecutar el código descrito en la tarea. Pasaremos a pasarle el código estopuede ser simplemente copiarlo desde nuestro escritorio dentro de un editor de texto como notepad. Compilaremos y ejecutaremos pasando lso parámetros el nombre de huho, paco y luis correspondientemente y posteriormente los mensajes especificados

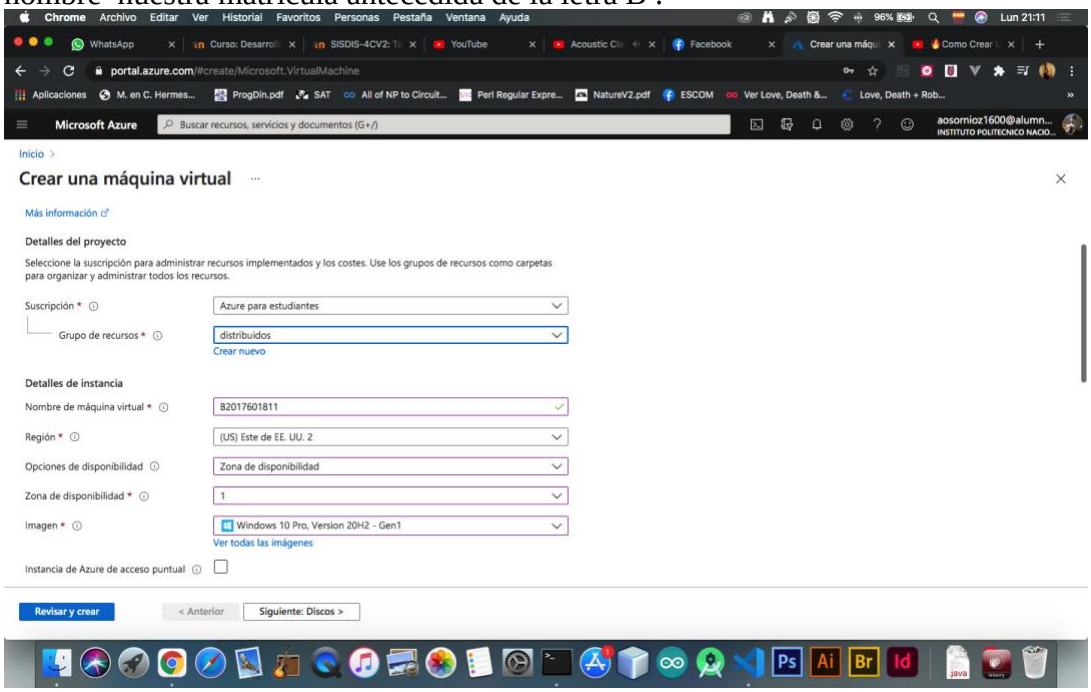
Codigo

```
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.nio.ByteBuffer;

class Chat{


   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();
   }


   static class Worker extends Thread{
      @Override
      public void run(){
         try{
            while(true){
               InetAddress group = InetAddress.getByName("230.0.0.0");
               MulticastSocket socket = new MulticastSocket(50000);
               socket.joinGroup(group);
               byte msj[] = recibe_mensaje_multicast(socket,1024);
               System.out.print("\n");
               System.out.println(new String(msj,"Windows-1252"));
               System.out.println("");
               socket.leaveGroup(group);
```

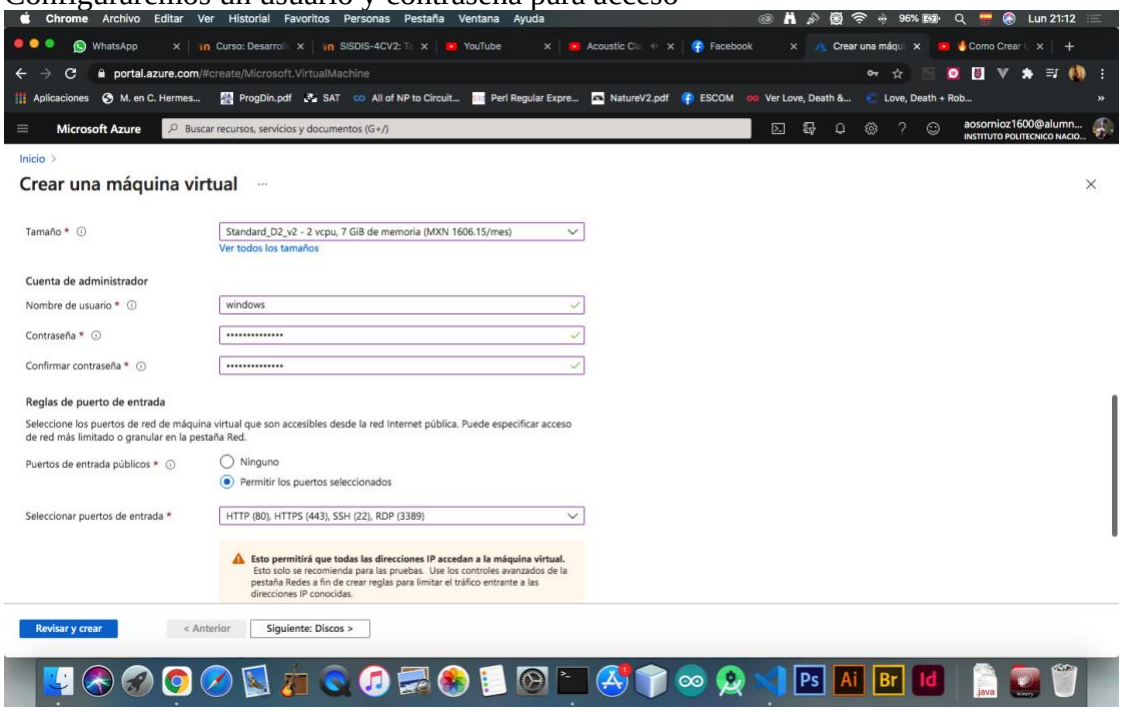
```
                socket.close();
            }
        }catch(Exception except){
            System.err.println(except.toString());
        }
    }
}

public static void main(String args[])throws Exception{
    Worker w = new Worker();
    w.start();
    String nombre = args[0];
    BufferedReader buffer = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("Ingrese el mensaje a enviar: ");
    while(true){
        String message = nombre + ":" + buffer.readLine();
        envia_mensaje_multicast(message.getBytes(), "230.0.0.0", 50000);
    }
}


}
```

Creacion de las maquinas virtuales

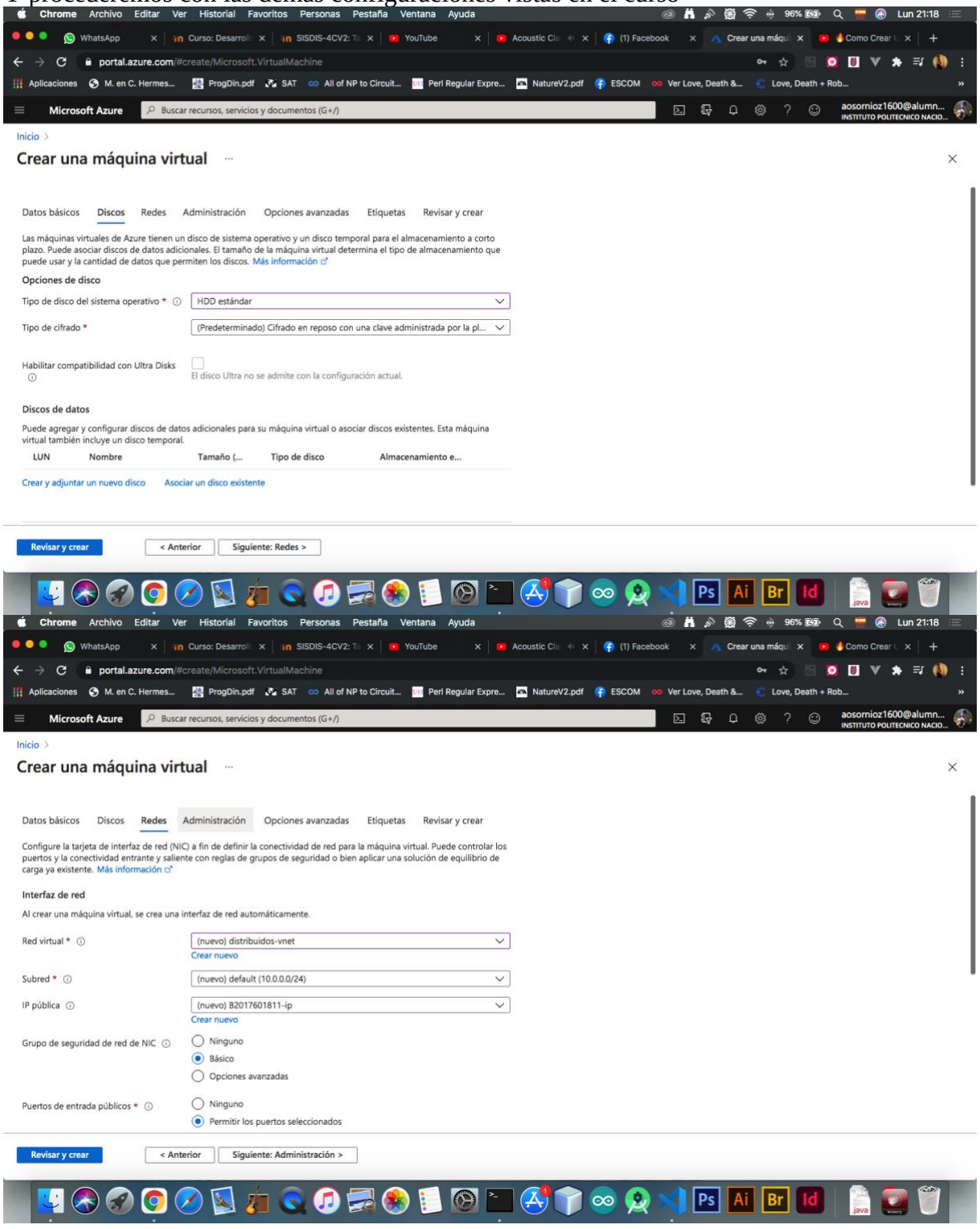
Crearemos una maquina virtual con el sistema operativo windows 10 y le pondermos como nombre nuestra matricula antecedida de la letra B :

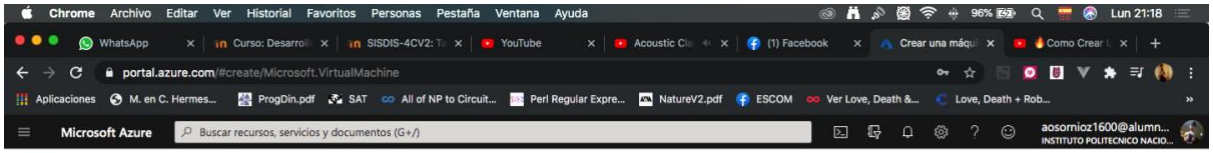


Configuraremos un usuario y contraseña para acceso



Y procederemos con las demas configuraciones vistas en el curso





Inicio >

Crear una máquina virtual ...

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

Configure las opciones de supervisión y administración de la VM.

Azure Security Center

Azure Security Center proporciona características unificadas de administración de la seguridad y protección contra amenazas en todas las cargas de trabajo de nube híbrida. Más información

La suscripción está protegida por el plan básico de Azure Security Center.

Supervisión

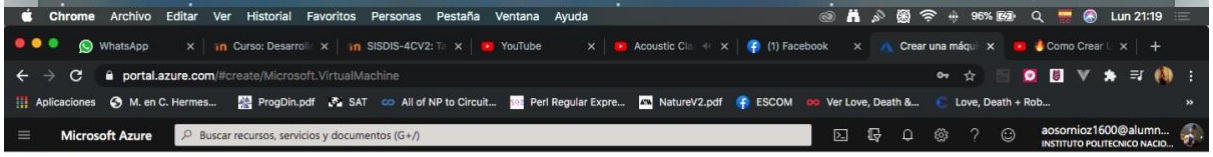
Diagnósticos de arranque
- ○ Habilitar con la cuenta de almacenamiento administrada (recomendado)
- ○ Habilitar con la cuenta de almacenamiento personalizada
- ◉ Deshabilitar

Habilitar diagnósticos del SO invitado ☐

Identidad

Identidad administrada asignada por el sistema ☐

Revisar y crear | < Anterior | Siguiente: Opciones avanzadas >



Inicio >

Crear una máquina virtual ...

Validación superada

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

DETALLES DEL PRODUCTO

Estándar D2 v2
por Microsoft
Términos de uso | Directiva de privacidad

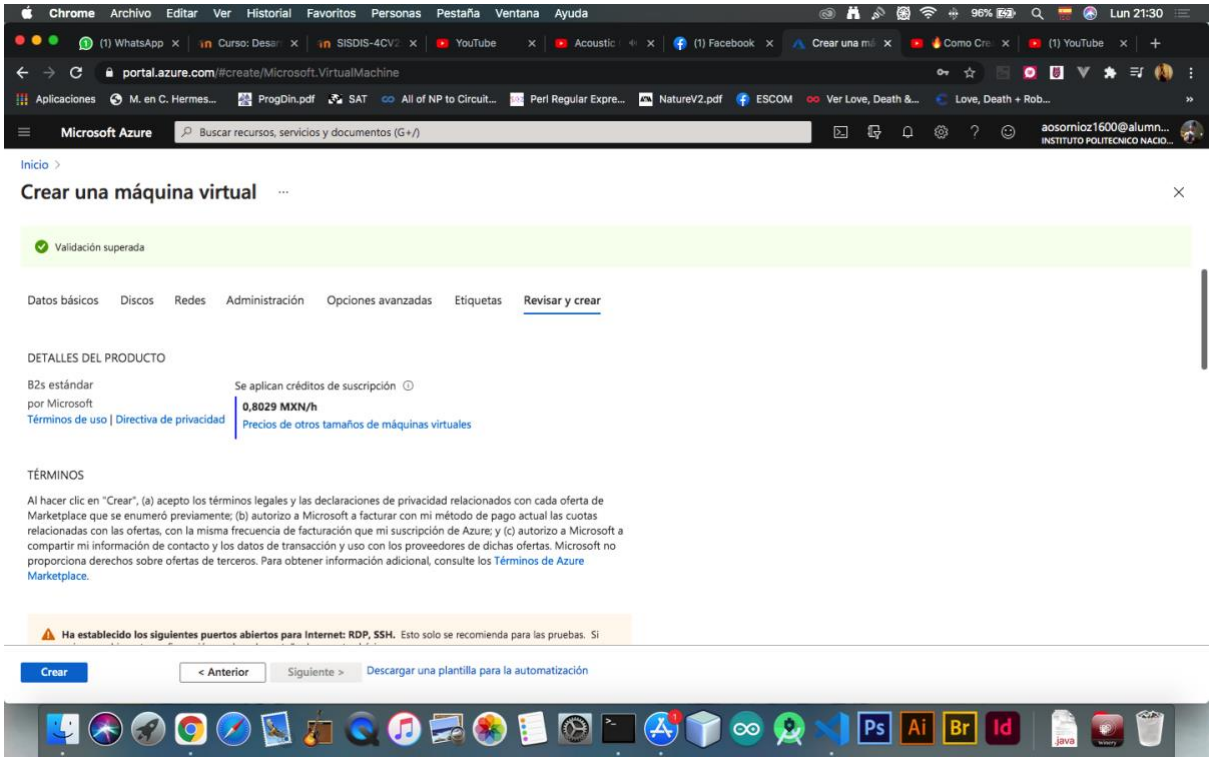
Se aplican créditos de suscripción
2,2002 MXN/h
Precios de otros tamaños de máquinas virtuales

TÉRMINOS

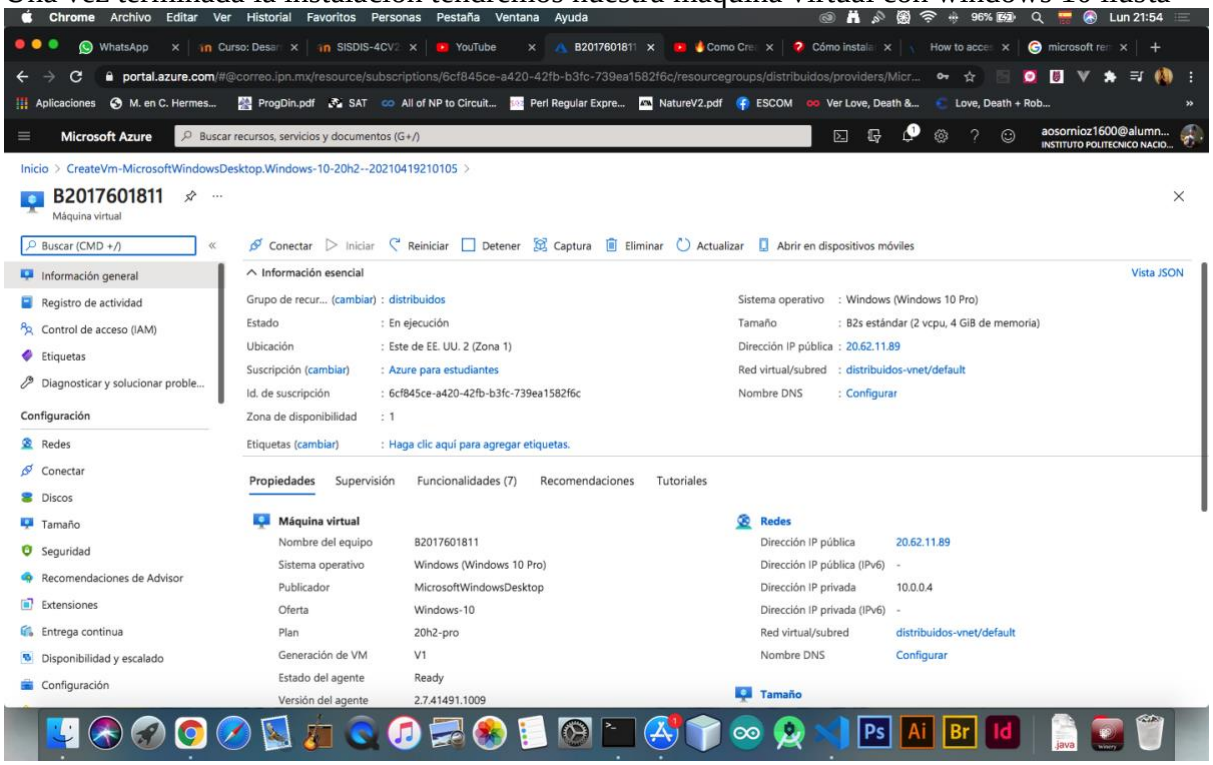
Al hacer clic en "Crear", (a) acepto los términos legales y las declaraciones de privacidad relacionados con cada oferta de Marketplace que se enumeró previamente; (b) autorizo a Microsoft a facturar con mi método de pago actual las cuotas relacionadas con las ofertas, con la misma frecuencia de facturación que mi suscripción de Azure; y (c) autorizo a Microsoft a compartir mi información de contacto y los datos de transacción y uso con los proveedores de dichas ofertas. Microsoft no proporciona derechos sobre ofertas de terceros. Para obtener información adicional, consulte los Términos de Azure Marketplace.

⚠ **Ha establecido los siguientes puertos abiertos para Internet: RDP, SSH.** Esto solo se recomienda para las pruebas. Si

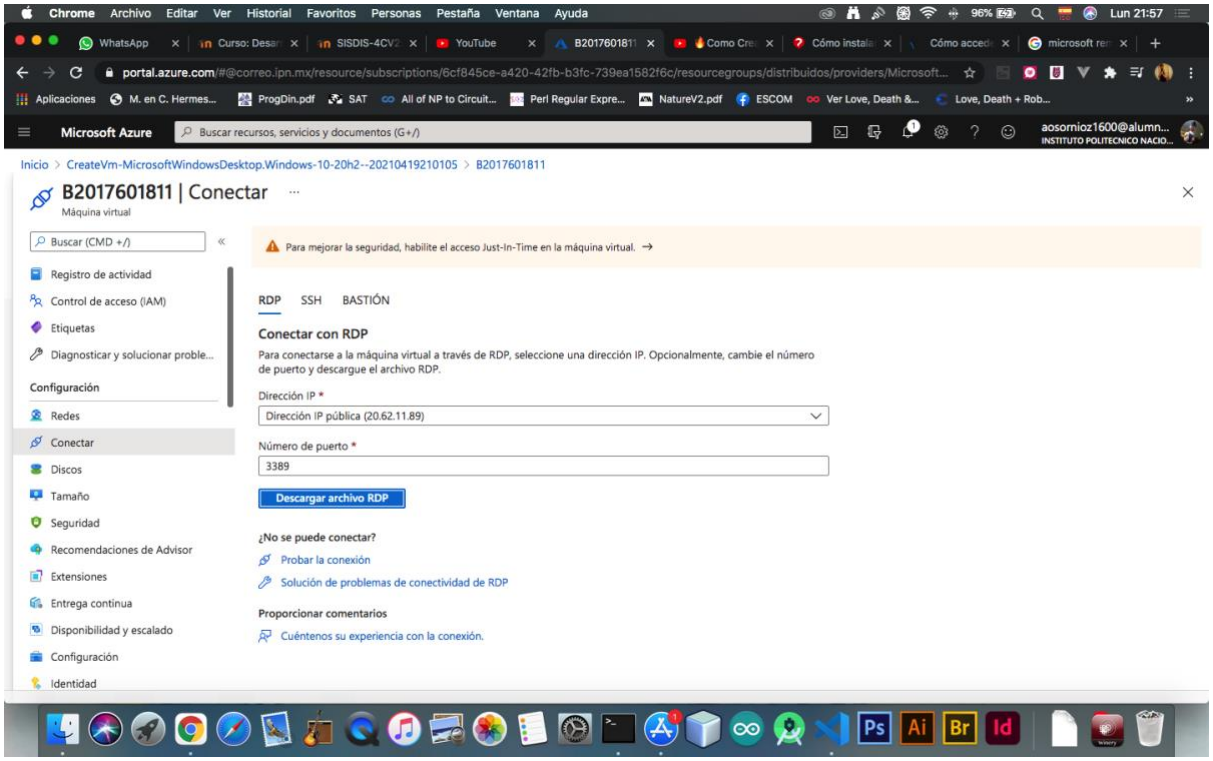
Crear | < Anterior | Siguiente > | Descargar una plantilla para la automatización



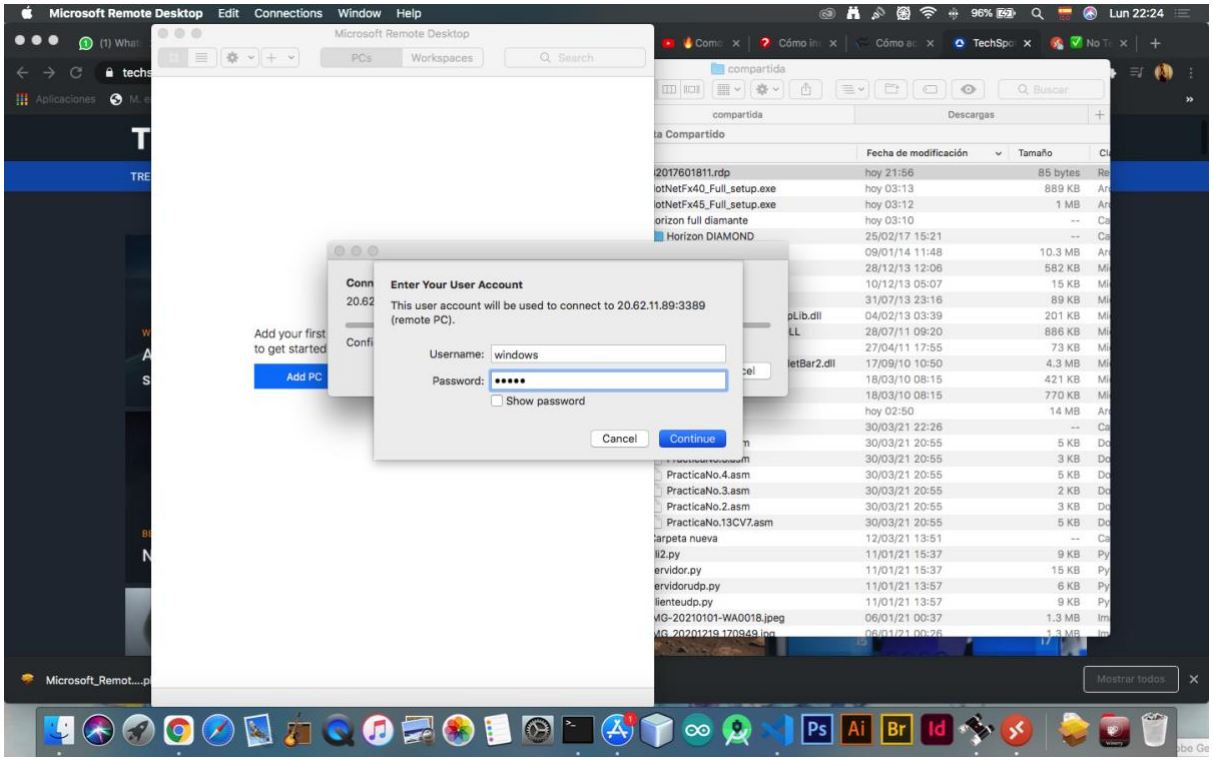
Una vez terminada la instalacion tendremos nuestra maquina virtual con windows 10 liusta

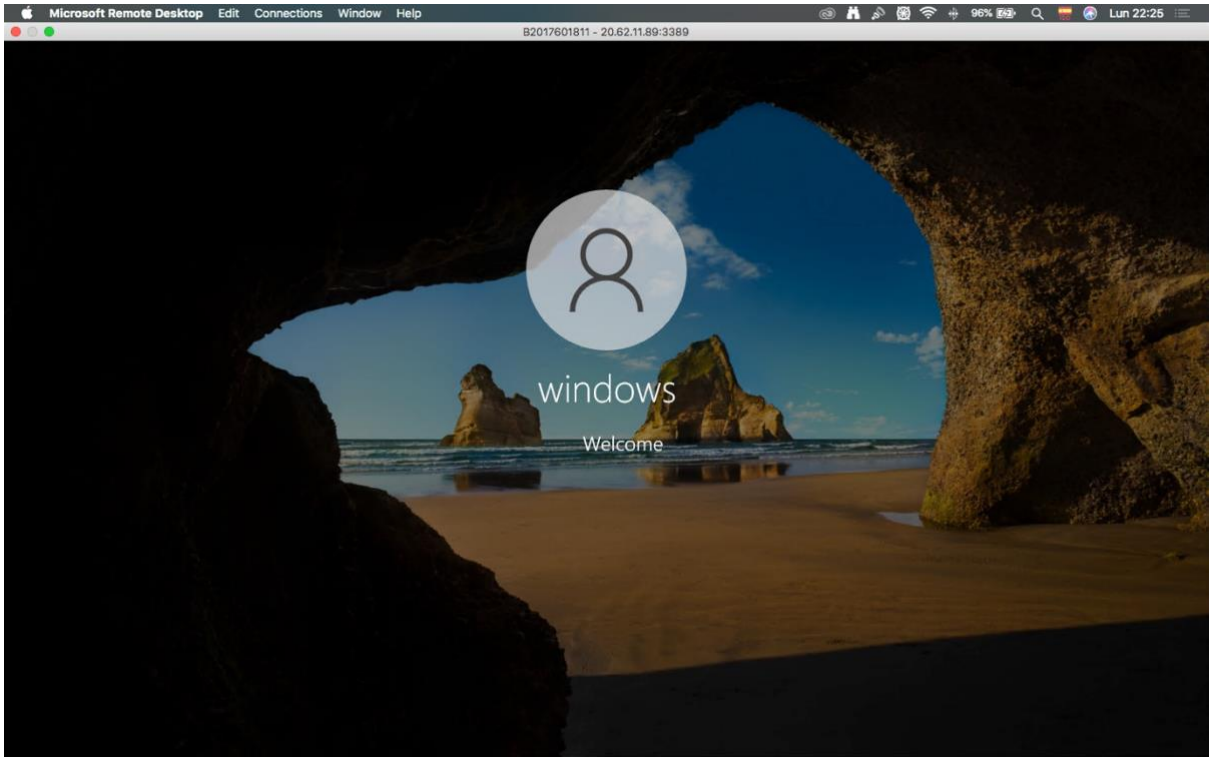


En el apartado de conexión encotnraremos RDP donde nos permetira descargar un archivo rdp para conectarnos via remota

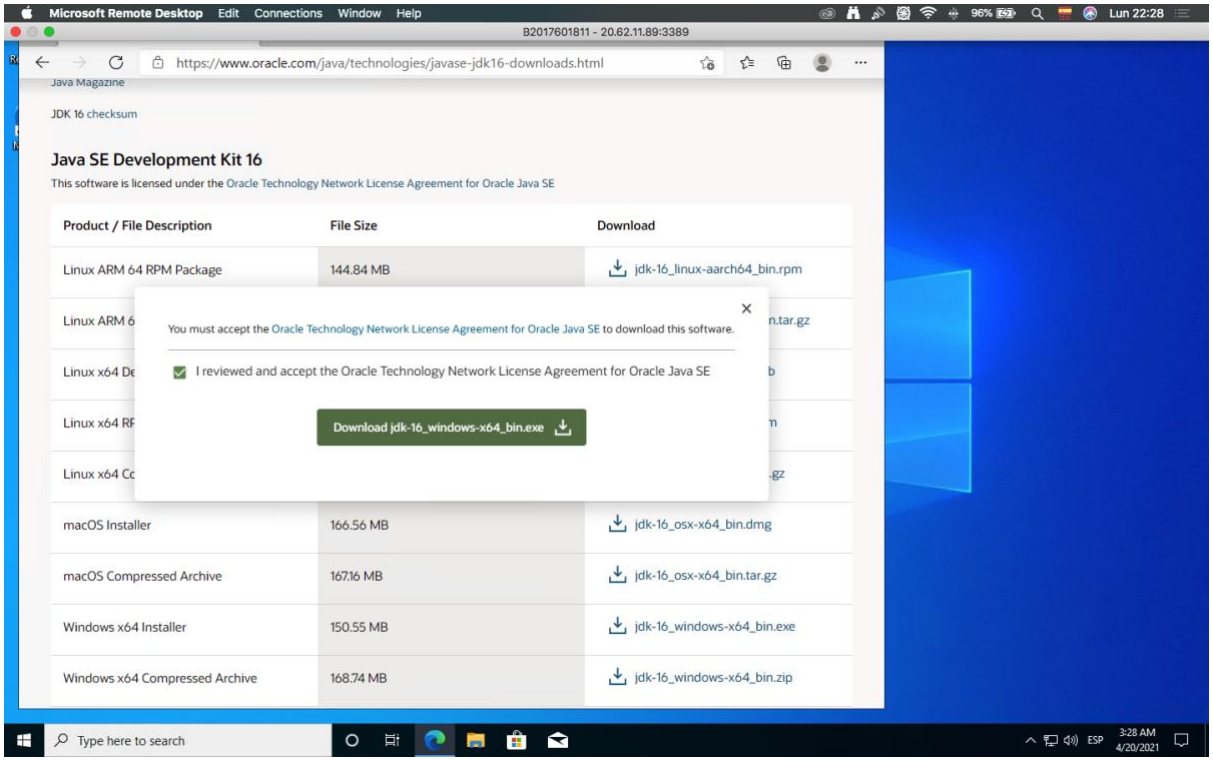


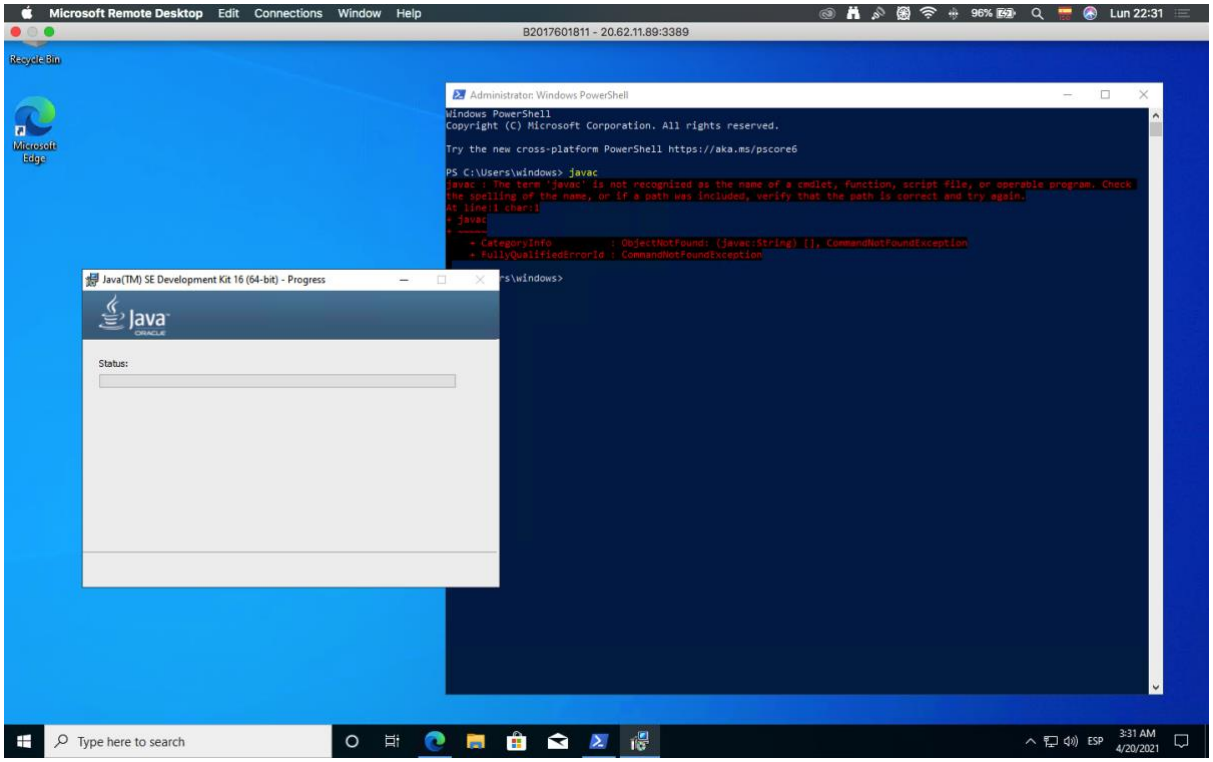
Una vez descargado nos permitira usarlo en Microsoft remote descktop para acceder a nuestra amquina virtual



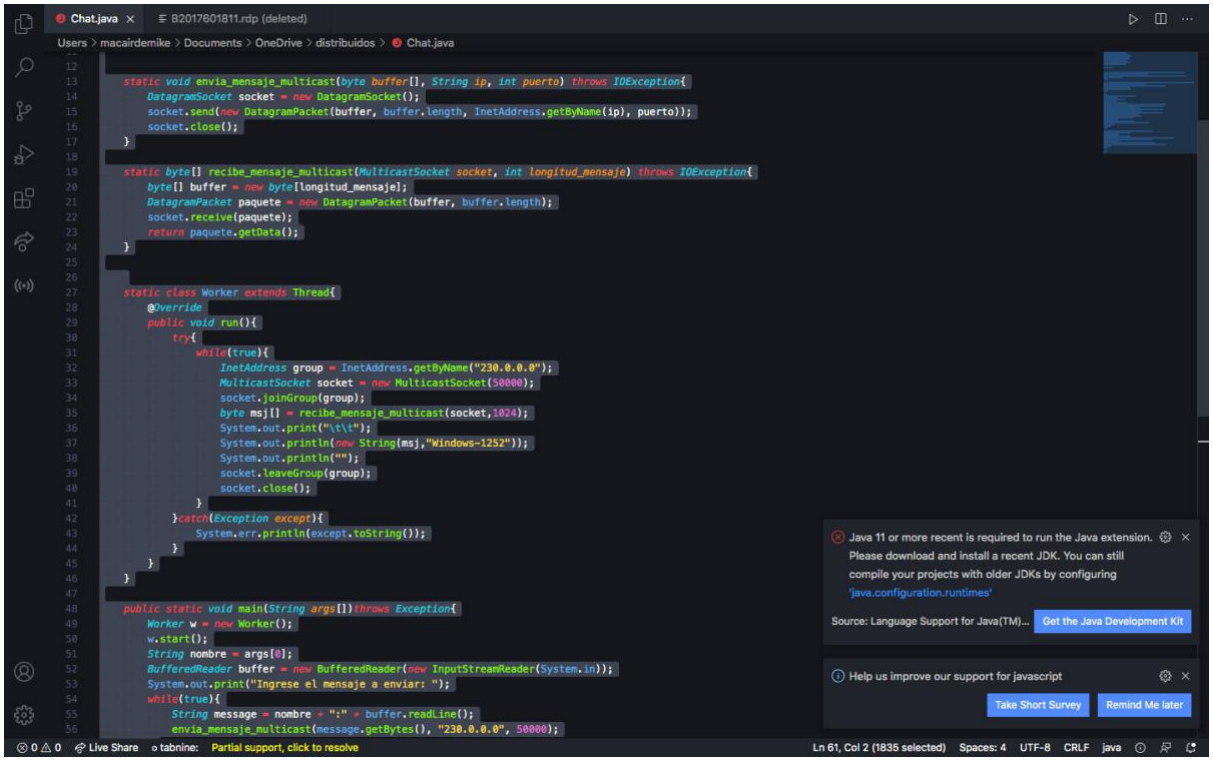


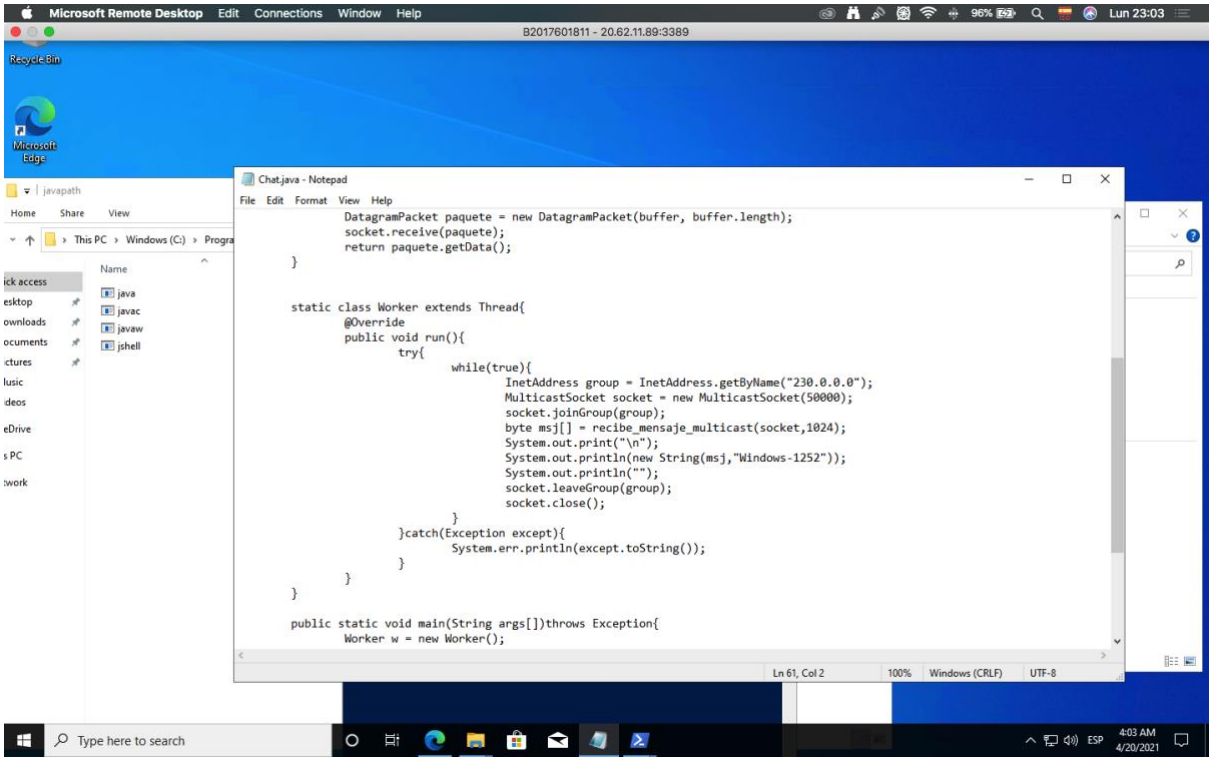
Instalamos el jdk de java para poder compilar y. correr nuestro código



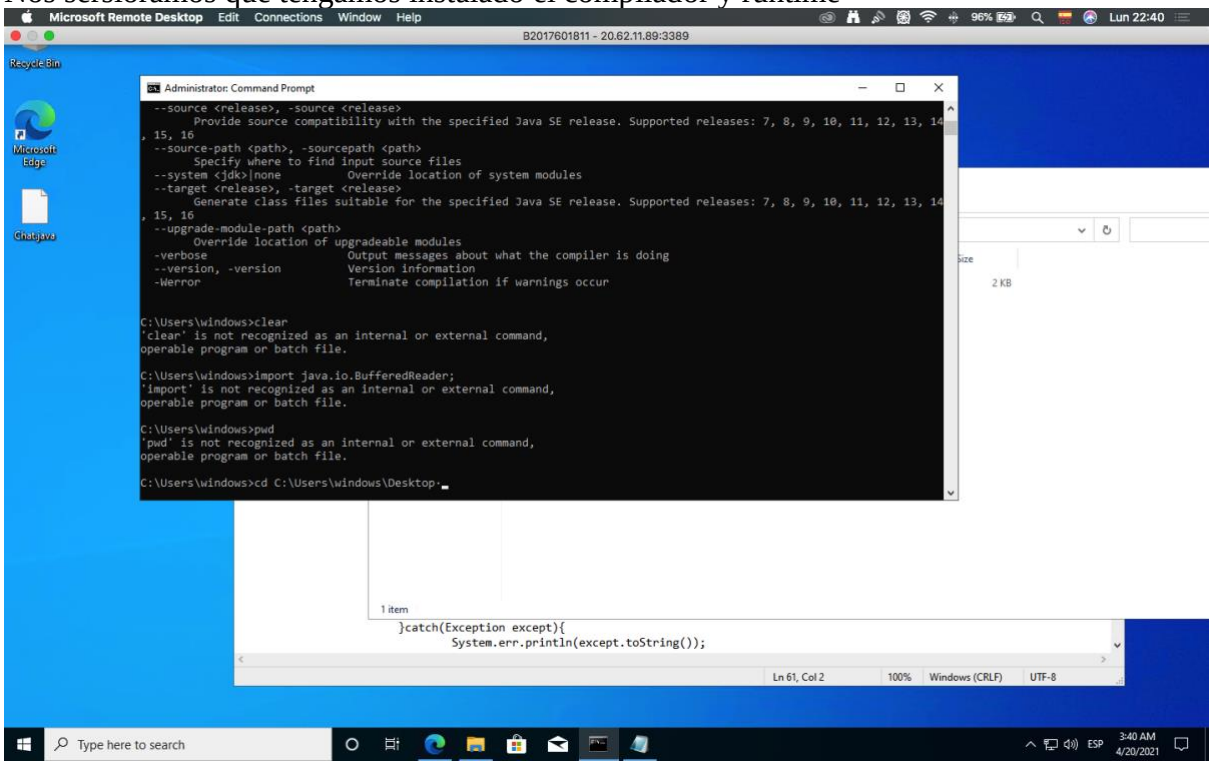


Copiamos nuestro codigo desde la computadora local a un editor de texto en nuestra maquina remota

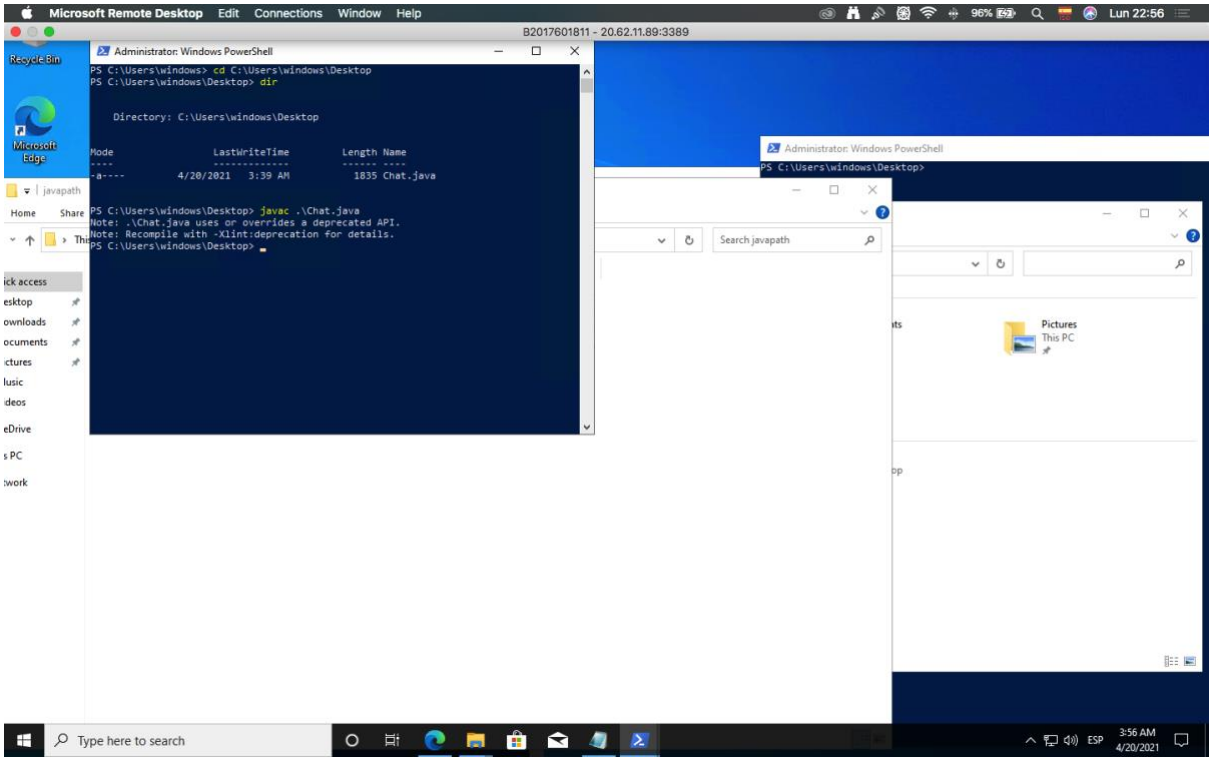

Chat.java
B2017601811.rdp (deleted)
Users > macairdemike > Documents > OneDrive > distribuidos > Chat.java
static void envia_mensaje_multicast(byte buffer[], String ip, int puerto) throws IOException{
DatagramSocket socket = new DatagramSocket();
socket.send(new DatagramPacket(buffer, buffer.length, 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();
}
static class Worker extends Thread{
@Override
public void run(){
try{
while(true){
InetAddress group = InetAddress.getByName("230.0.0.0");
MulticastSocket socket = new MulticastSocket(50000);
socket.joinGroup(group);
byte msj[] = recibe_mensaje_multicast(socket,1024);
System.out.print("\t\t");
System.out.println(new String(msj,"Windows-1252"));
System.out.println("");
socket.leaveGroup(group);
socket.close();
}
}catch(Exception except){
System.err.println(except.toString());
}
}
}
public static void main(String args[])throws Exception{
Worker w = new Worker();
w.start();
String nombre = args[0];
BufferedReader buffer = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Ingrese el mensaje a enviar: ");
while(true){
String message = nombre + ":" + buffer.readLine();
envia_mensaje_multicast(message.getBytes(), "230.0.0.0", 50000);
Java 11 or more recent is required to run the Java extension.
Please download and install a recent JDK. You can still compile your projects with older JDKs by configuring 'java.configuration.runtimes'
Source: Language Support for Java(TM)...
Get the Java Development Kit
Help us improve our support for javascript
Take Short Survey
Remind Me later
Live Share
tabnine:
Partial support, click to resolve
Ln 61, Col 2 (1835 selected)
Spaces: 4
UTF-8
CRLF
java



Nos sersioramos que tengamos instalado el compilador y runtime

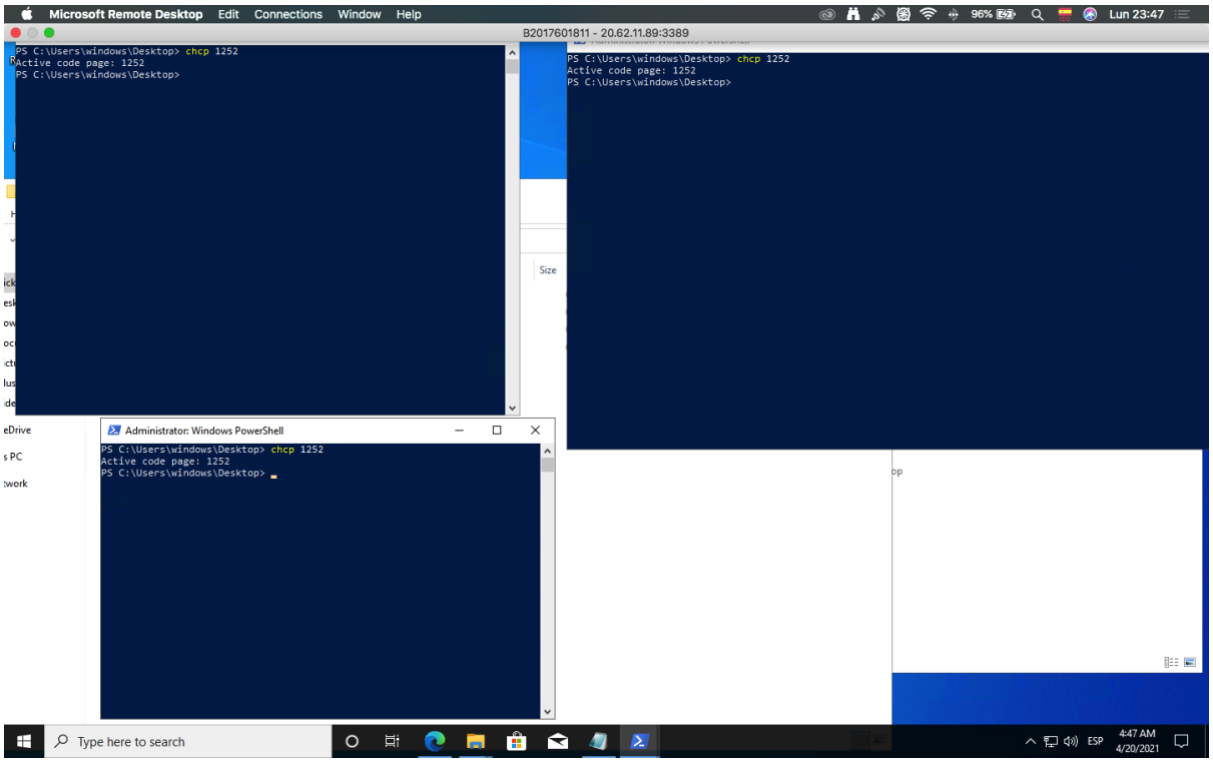


Nos dirijimos a donde guardamos nuestro codigo (en este caso escritorio) y compilamos nuestro codigo

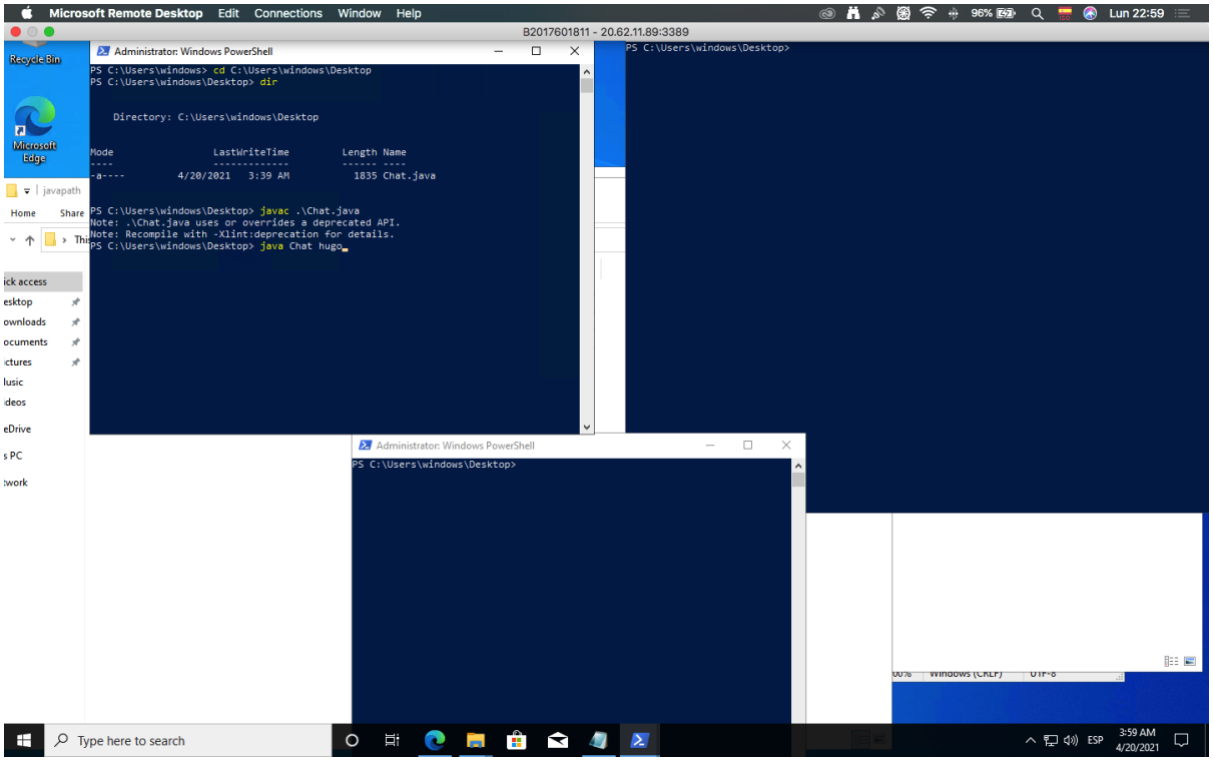


Abrimos otras dos terminales y las situamso en la misma carpeta y les enviamos el comando

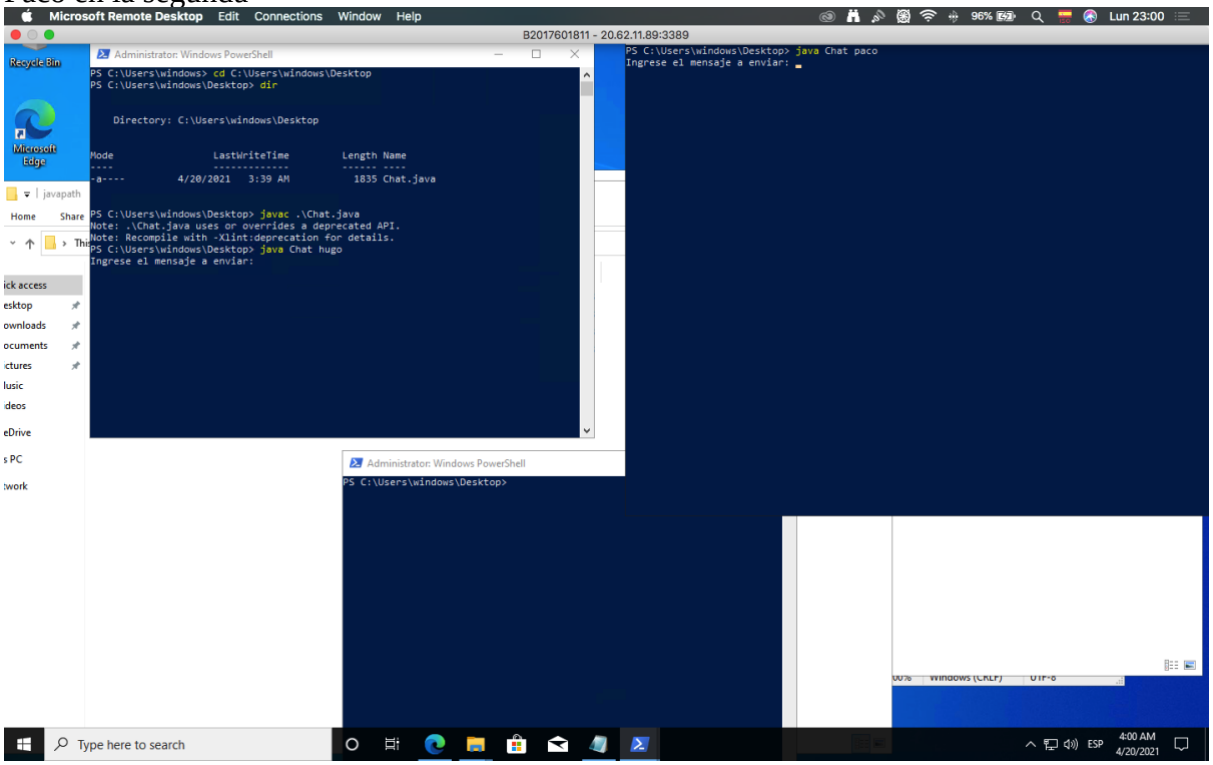
chcp 1252 para que windows nos reconosca los caracteres



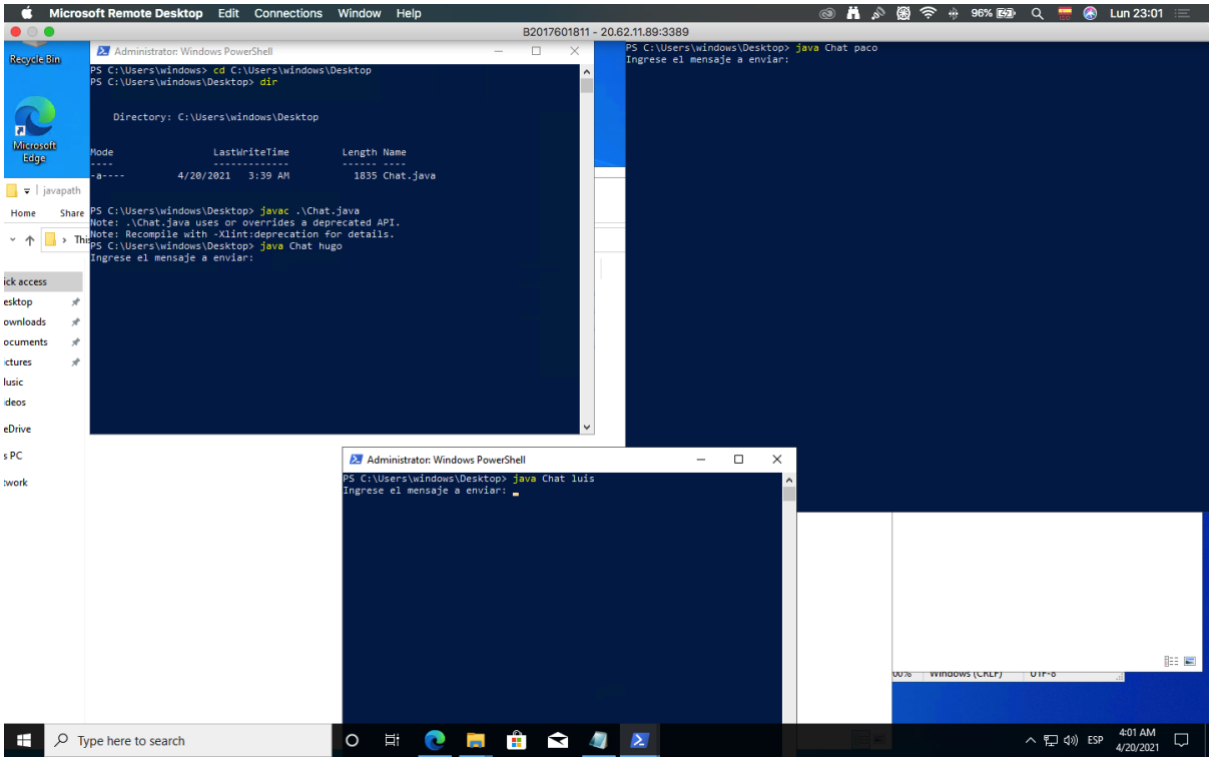
Luego procedemos a ejecutar el chat pasandoles el nombre correspondiente
Hugo para la primer terminal :



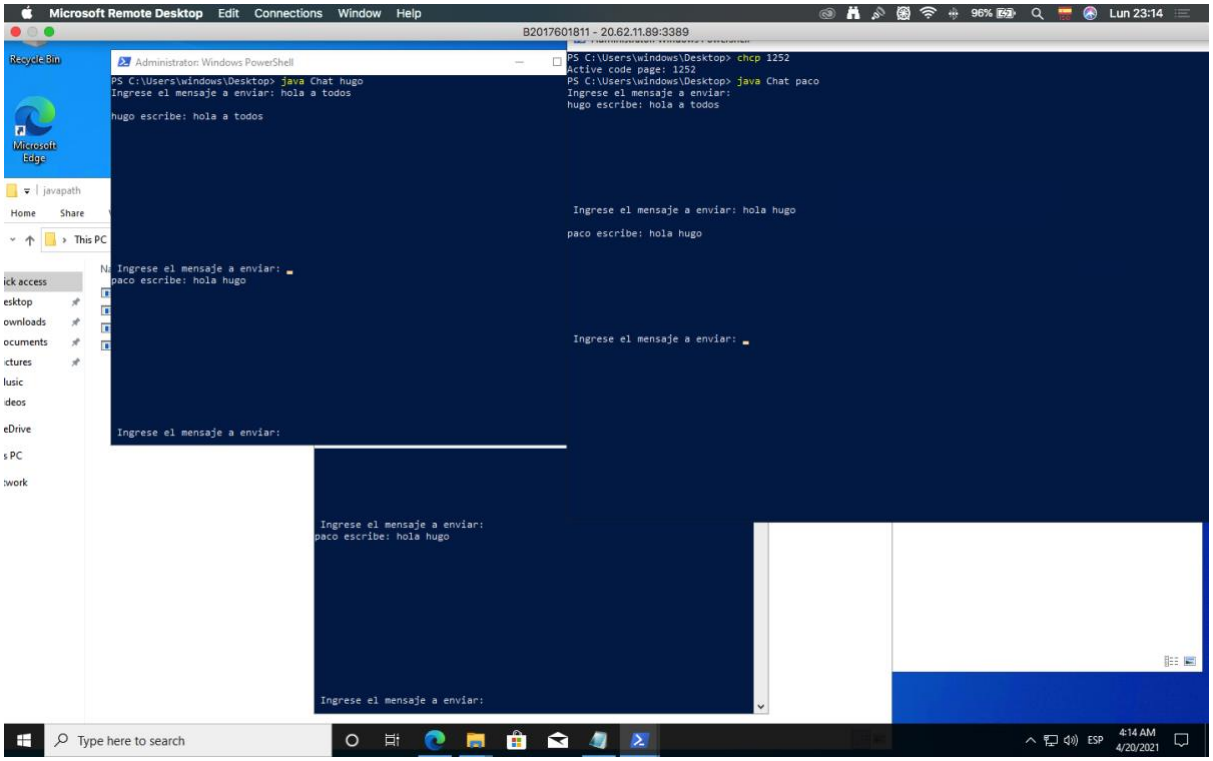
Paco en la segunda



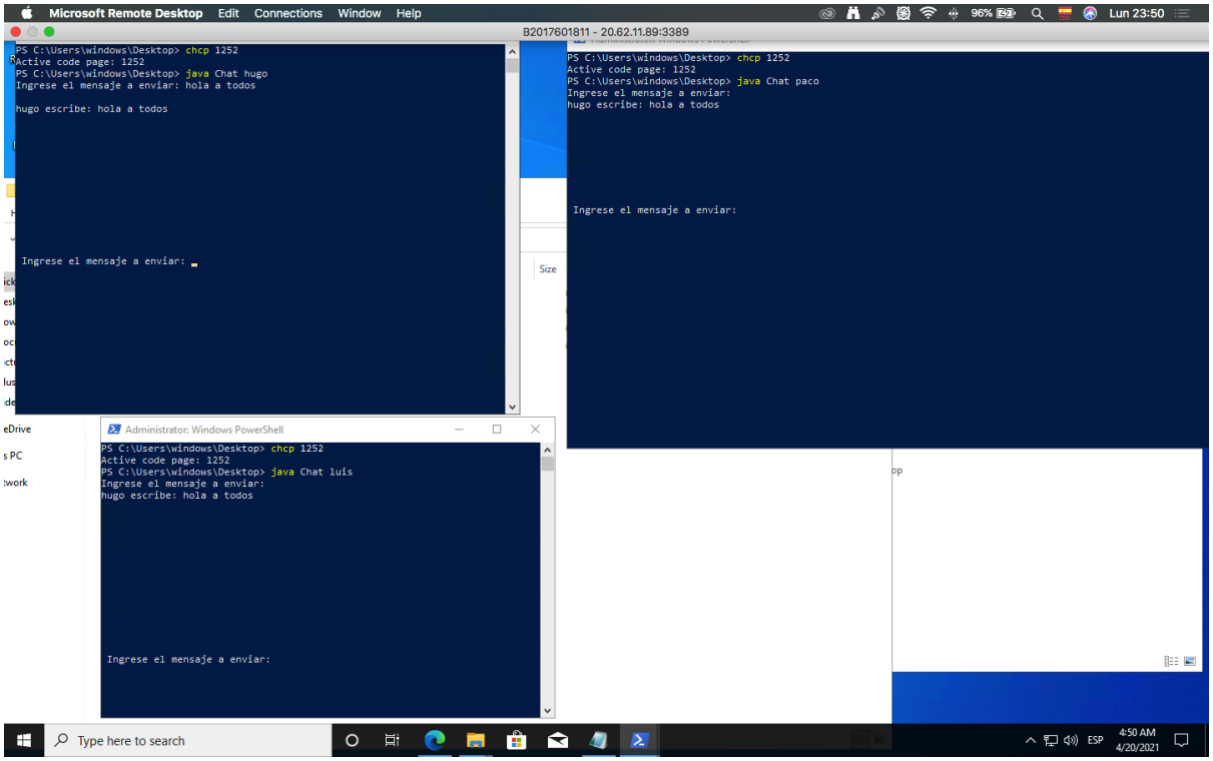
Luis:



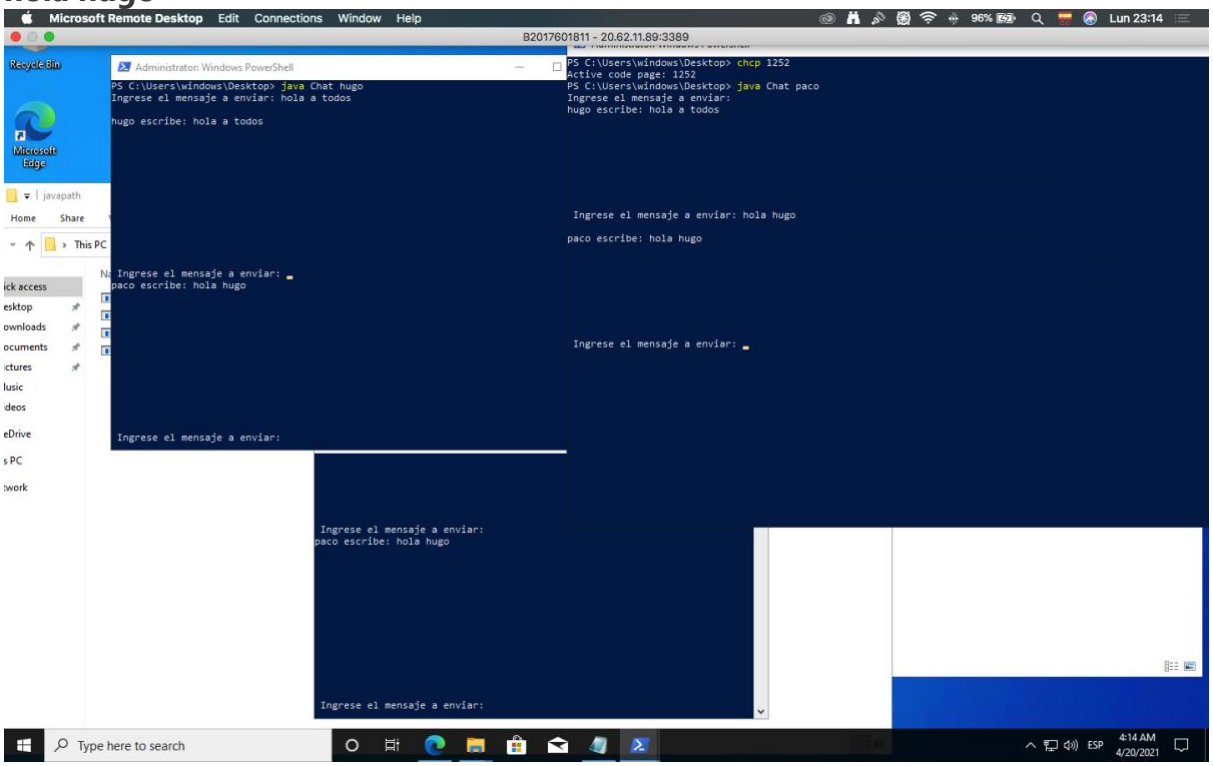
Procesedemos a pasarles los mensajes
Al momento de mandar el primer mensaje windows nos mandara un mensaje de firewall para permitir abrir el puerto indicado (olvide tomar captura xD)



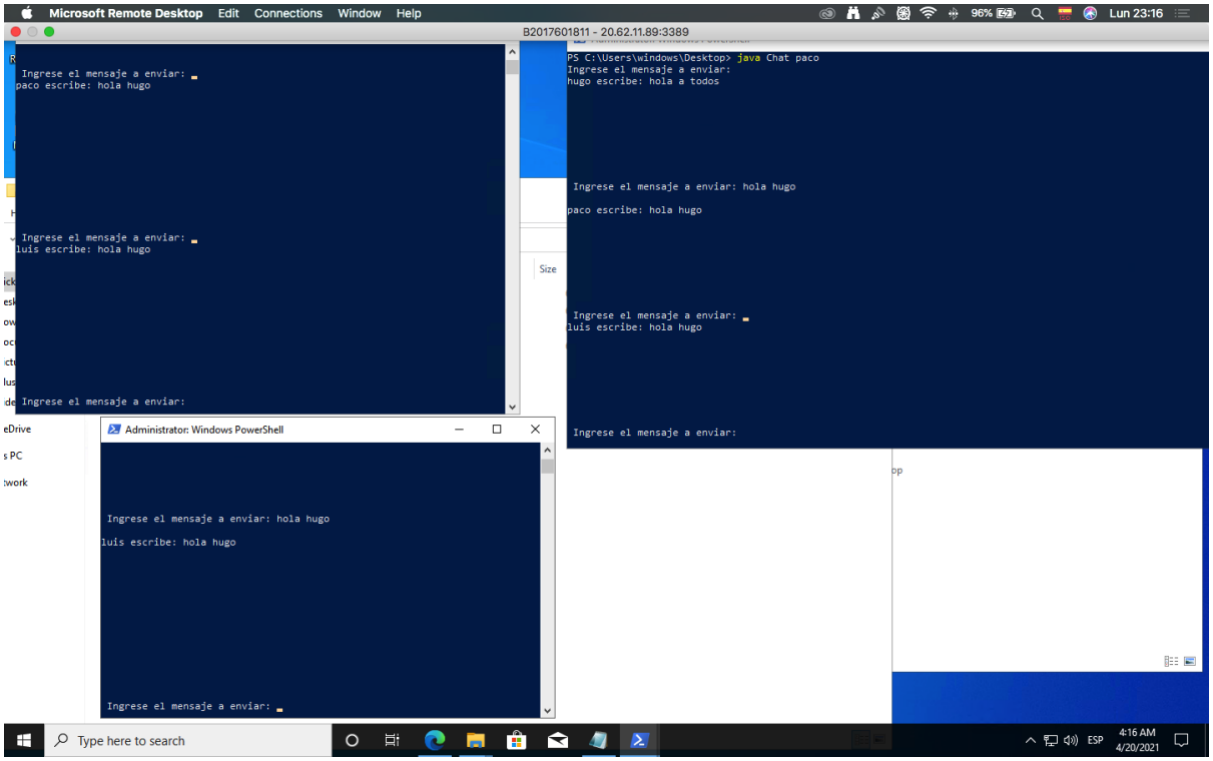
hugo debe escribir:
hola a todos



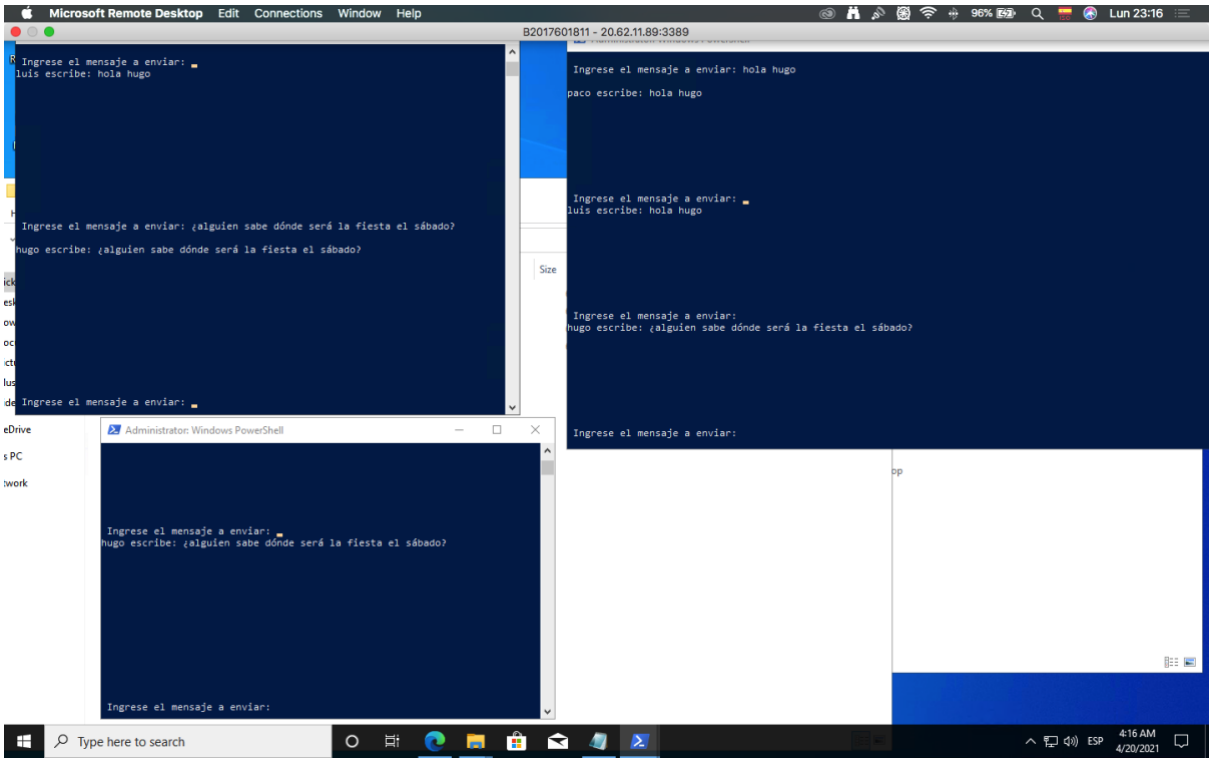
paco debe escribir:
hola hugo



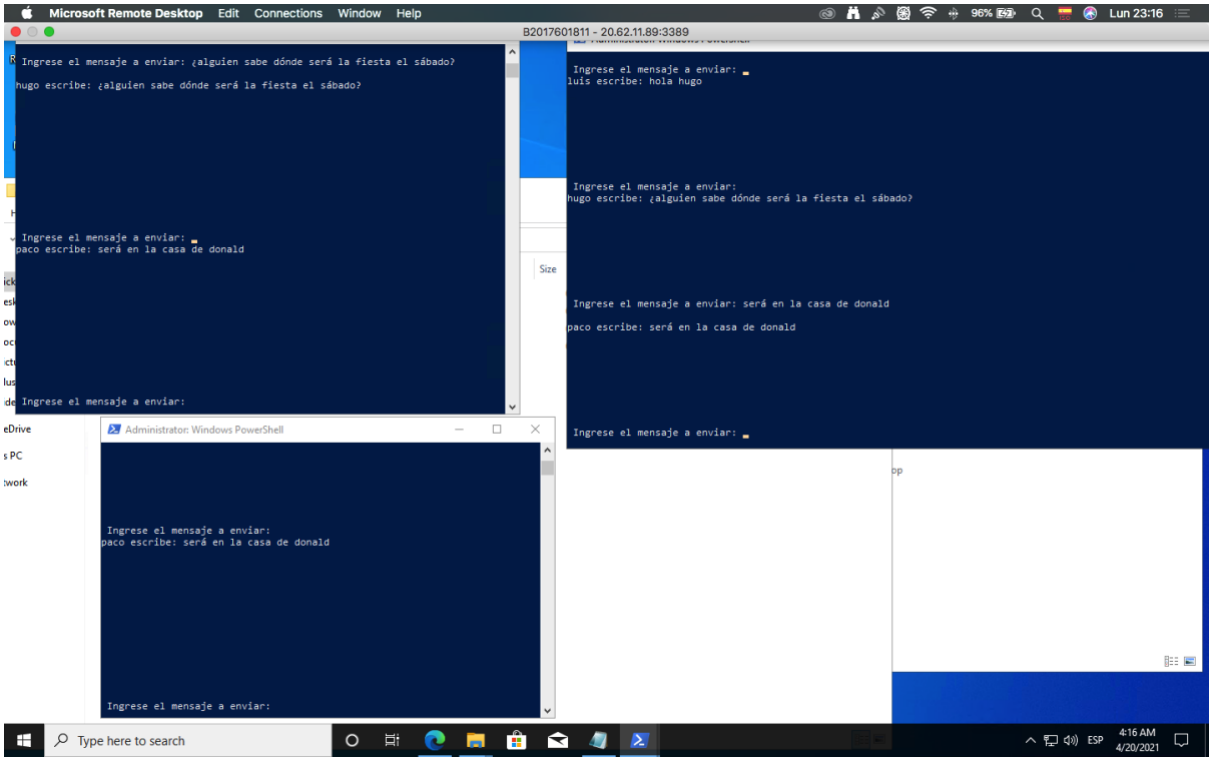
luis debe escribir:
hola hugo



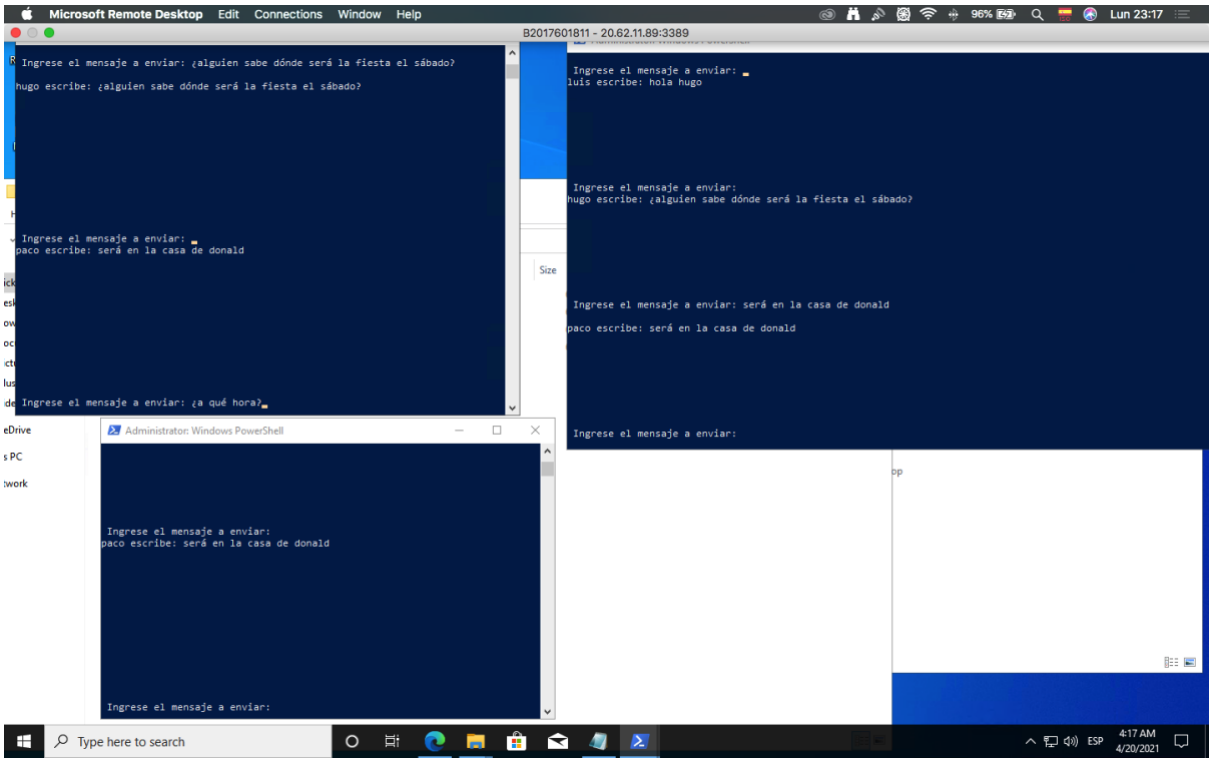
hugo debe escribir:
¿alguien sabe dónde será la fiesta el sábado?



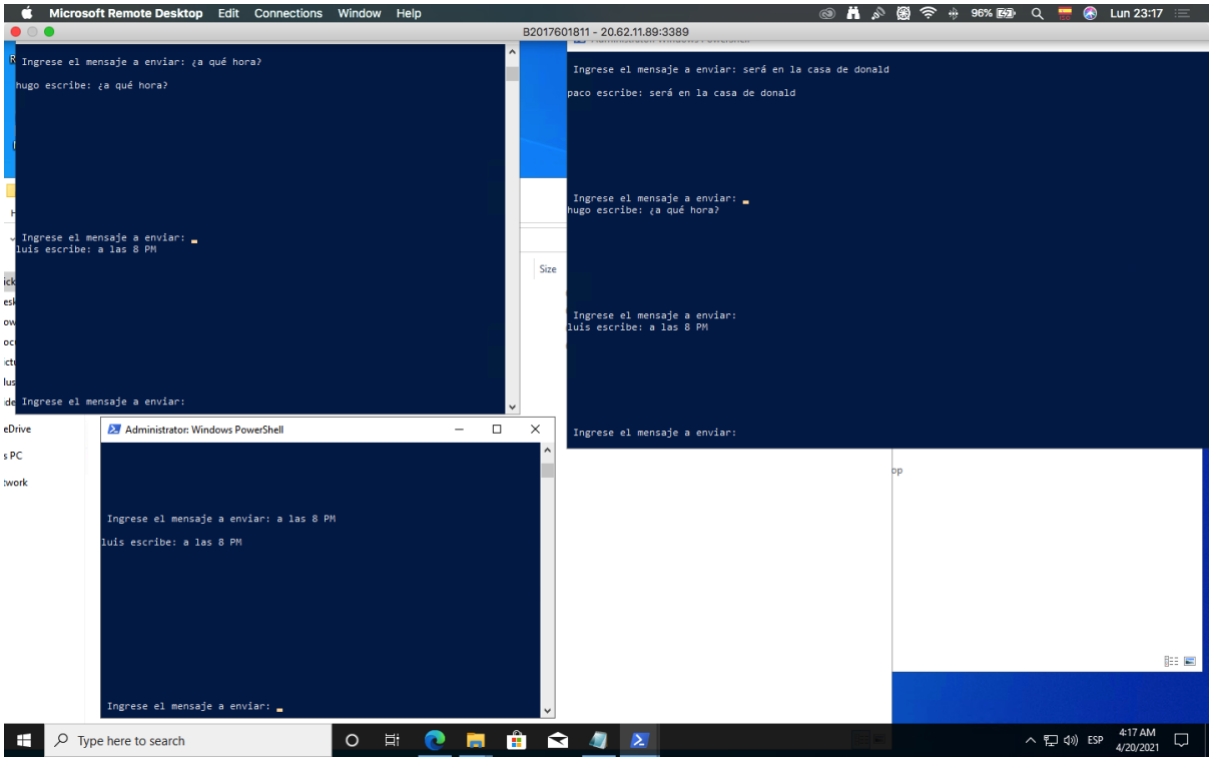
paco debe escribir:
será en la casa de donald



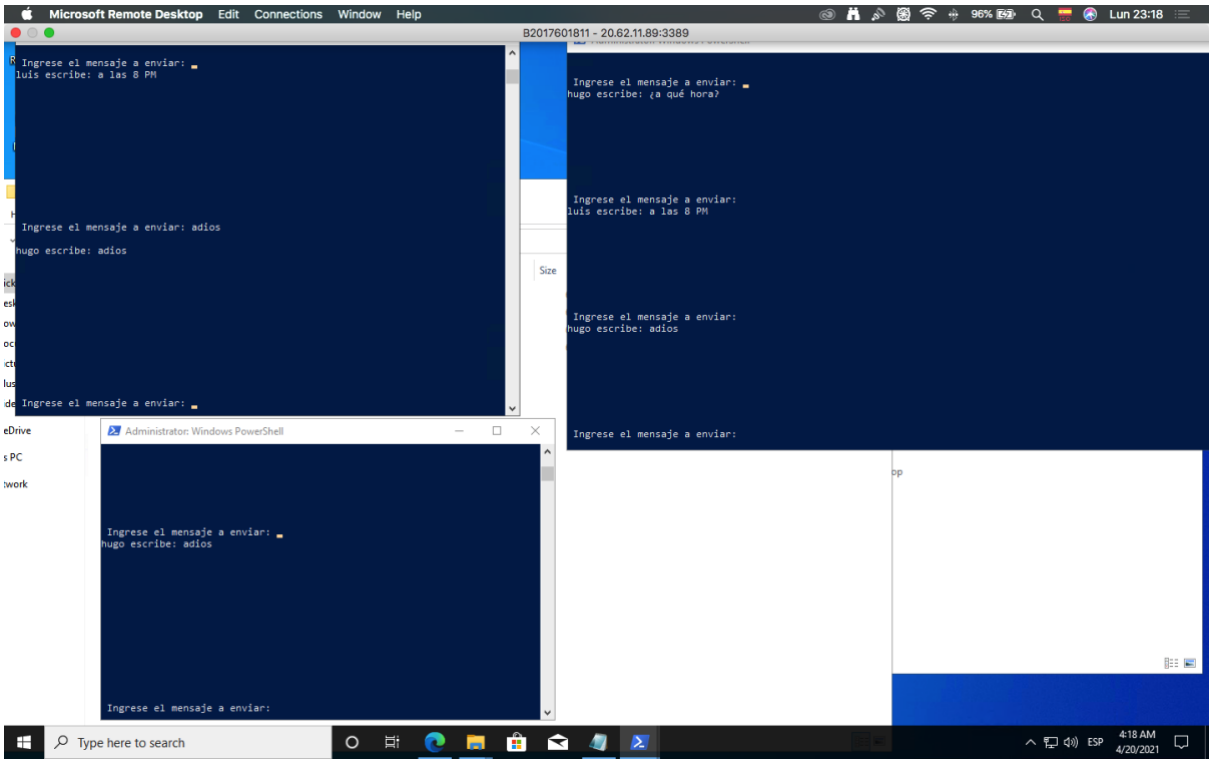
hugo debe escribir:
¿a qué hora?



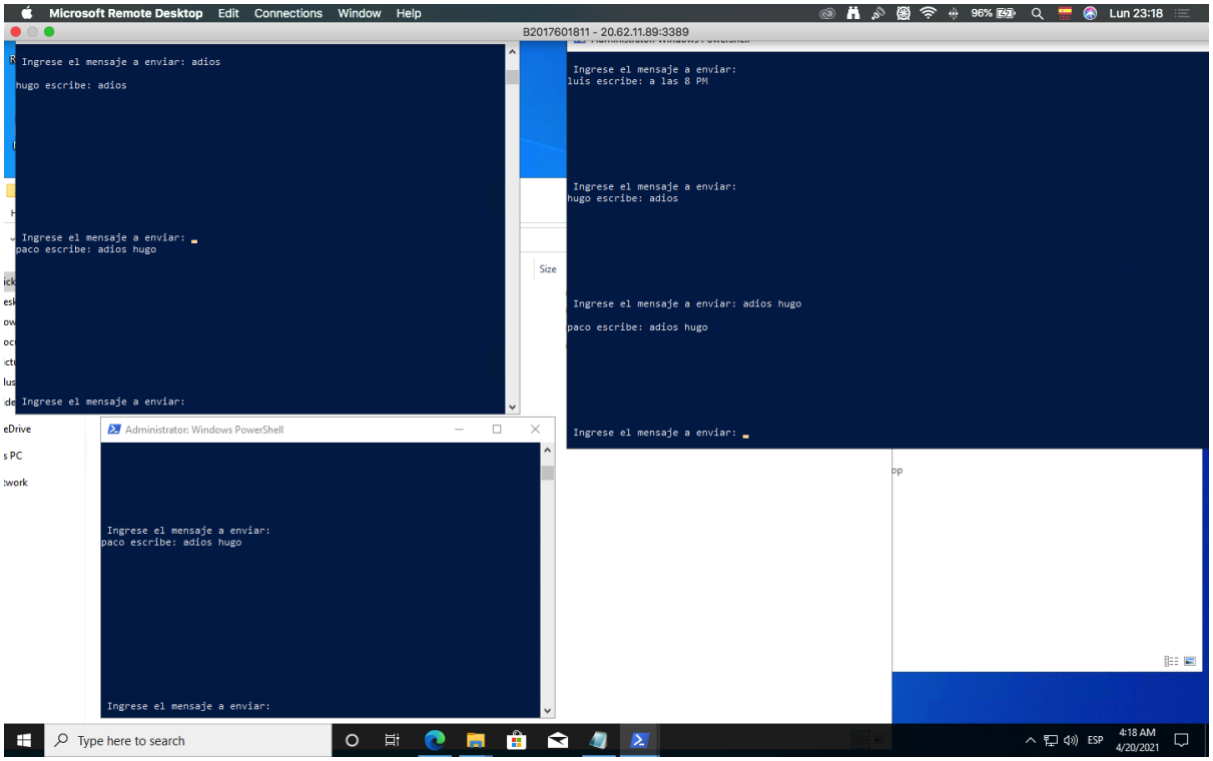
luis debe escribir:
a las 8 PM



hugo debe escribir:
adios

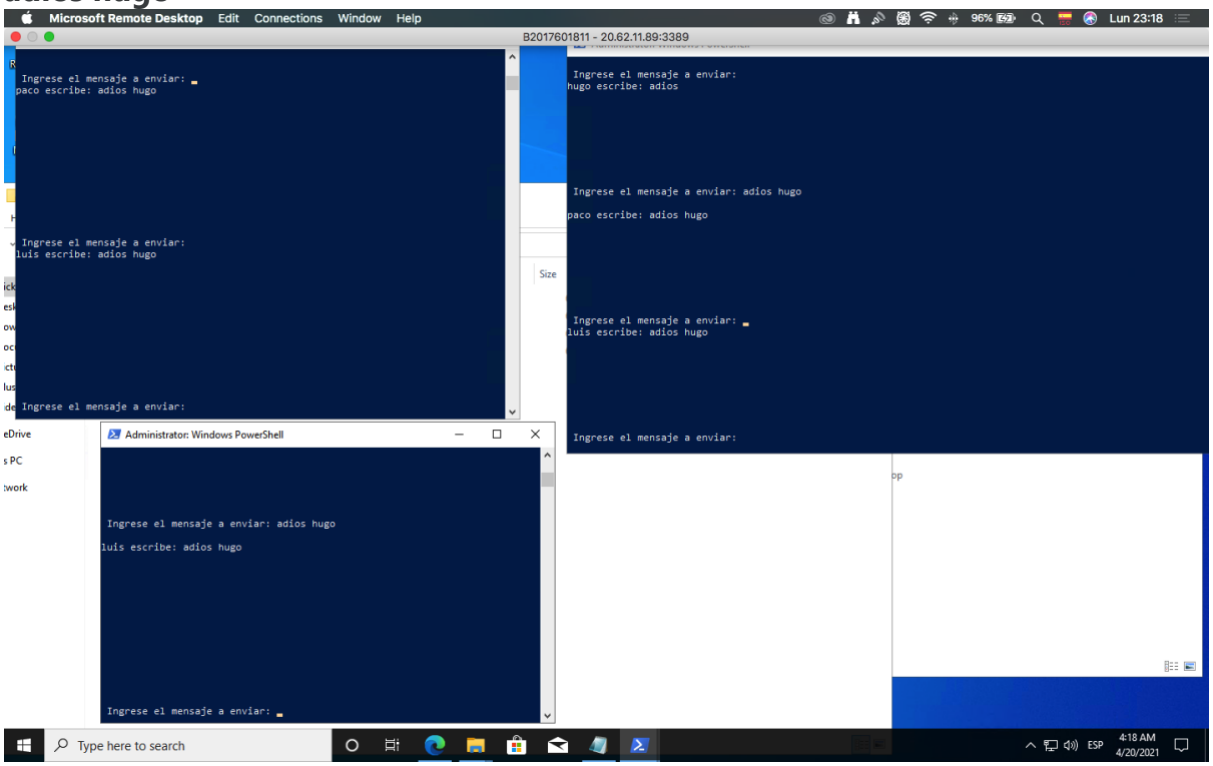


paco debe escribir:
adios hugo



luis debe escribir:

adios hugo



Conclusión

Azure es una plataforma excelente para la ejecución de sistemas distribuidos ya que brinda todas las herramientas para ejecutar un servidor en remoto conocer su estatus, dar una configuración completa y acceso. Asi pudimos ejecutar un sistema distribuido sin usar recursos de nuestra computadora personal y aun precio accesible y el multi cast es una herramienta para comunicación que nos brinda la posibilidad de no tener un nodo centralizado