



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Tarea 4. Chat multicast

Unidad de aprendizaje: Desarrollo de Sistemas Distribuidos

Grupo: 4CV11

Alumno:
Sanchez Mendez Edmundo Josue

Profesor:
Pineda Guerrero Carlos

23 de septiembre de 2021

Índice

1. Introducción	2
2. Desarrollo	2
2.1. Configuración de maquina virtual	5
2.2. Compilación del código	15
2.3. Ejecución del programa	18
3. Conclusiones	18
Referencias	18

1. Introducción

Mediante el desarrollo de un solo programa en Java en modo consola se implementara un chat utilizando comunicación multicast mediante datagramas. Se deberá ejecutar el programa en una máquina virtual con Windows Server 2012 en Azure. 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 10000. El paquete a enviar deberá tener la siguiente forma: nombre_usuario:mensaje_ingresado donde 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, separados por el carácter “:” sin espacios.

2. Desarrollo

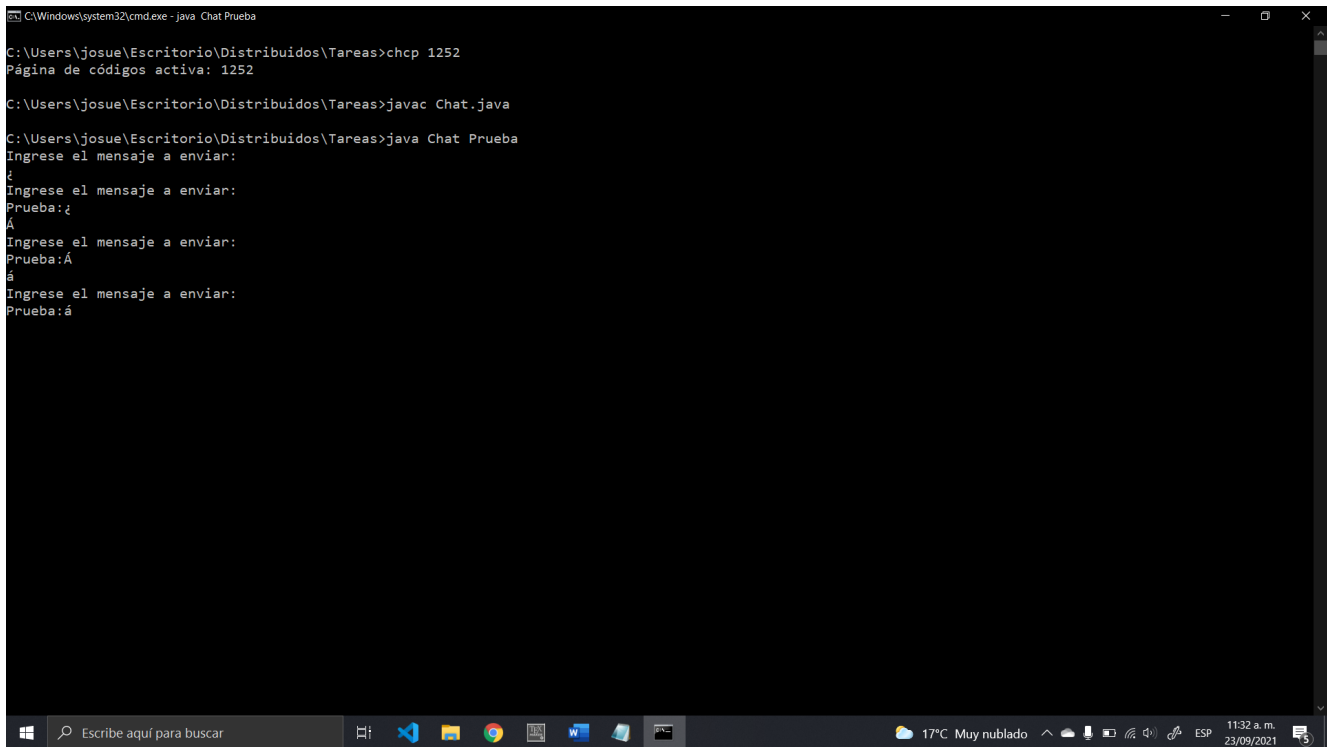
El desarrollo de esta tarea no fue nada complicada, ya que habíamos hecho un ejercicio similar en clase y prácticamente fue solo implementar el código visto, sin embargo, una de las cosas más interesantes fue el cómo imprimir lo signos de interrogación y las letras acentuadas en CMD, esto fue un gran conflicto ya que Windows puede usar diferentes páginas de códigos, estas páginas nos define el conjunto de caracteres que se está utilizando, usualmente Windows tiene por defecto la pagina 850, 437 o la 1252, las cuales son diferentes en cuanto a codificación de los caracteres, para poder saber que numero de página utiliza nuestro CMD se usa el comando: chcp, para cambiarlo usamos chcp n, donde n es el número de página.

La página 437 que Java reconoce como IBM437, la cual es usada en la región de los Estados Unidos, esta página es el conjunto de caracteres que incorporaba la primera versión del Personal Computer (PC) de IBM, hay pequeñas diferencias con las otras dos páginas de códigos que usa Windows, de hecho, las otras dos se crearon para poder complementar esta página.

La página 850 fue creada para varios idiomas por lo que se le llama Multilingual (Latin I) a veces denominada Europa occidental, esta página a diferencia de la página 437 es que varios caracteres empleados para el dibujo de cuadros, así como letras griegas y otros fueron reemplazados por caracteres latinos con diacríticos requeridos por idiomas como el español, Java la reconoce como IBM850.

Por último, la pagina 1252 que Java reconoce como windows-1252 es una codificación de caracteres del alfabeto latino, usada por defecto cuando unicode no se usa en los componentes oficiales de Microsoft Windows en inglés y en algunos lenguajes occidentales. Esta codificación es un superconjunto de ISO 8859-1

Ahora el desarrollo de la práctica se hizo en mi laptop la cual usa la pagina 850, esta página tiene los caracteres especiales que se usaran en las pruebas de esta práctica. Ahora al querer usar esta pagina al insertar el carácter “¿” el resultado al mandar el mensaje era el carácter “ lo cual es erróneo, al empezar el desarrollo de la práctica lo cual fue hace unos días me di cuenta de que si usamos la pagina 1252 y usamos UTF-8 esto era solucionado y la impresión era correcta, como podemos ver en la figura 1, esto lo logre ya que decidí ver como se codificaban los caracteres que ingresaba el chat, las lineas de código están comentadas.

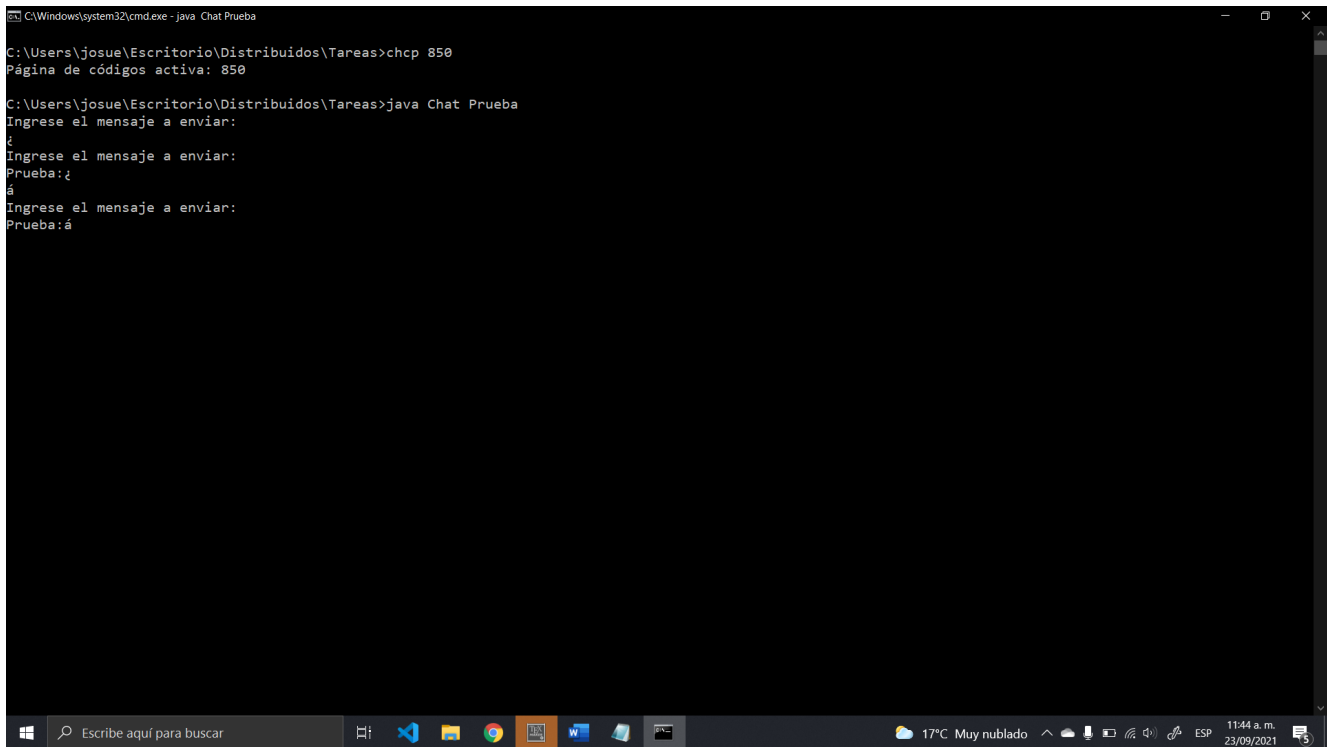


```
C:\Windows\system32\cmd.exe - java Chat Prueba
C:\Users\josue\Escritorio\Distribuidos\Tareas>chcp 1252
Página de códigos activa: 1252
C:\Users\josue\Escritorio\Distribuidos\Tareas>javac Chat.java
C:\Users\josue\Escritorio\Distribuidos\Tareas>java Chat Prueba
Ingrese el mensaje a enviar:
¿
Prueba:¿
Á
Prueba:Á
á
Prueba:á
```

Figura 1: Prueba usando la página 1252 y UTF-8.

Ahora me doy cuenta de que estaba cometiendo un error, lo mas adecuado era adaptarse a la pagina en uso de nuestro CMD ya que, como veremos mas adelante en el reporte podrían surgir errores al momento de ejecutar el programa en nuestra máquina virtual.

Por lo que para poder usar la pagina por defecto de mi computadora la cual es la 850, no solo era necesario poner que tipo de codificación se usaria al momento de desplegar el mensaje enviado en el chat o al momento de enviar el mensaje de forma multicast, sino que también al momento de leer el mensaje desde la consola, en mi caso uso Scanner, por lo que también se le tenia que colocar el tipo de decodificación, como estamos usando la pagina 850 el nombre que reconoce Java es IBM850, y así como vemos en la figura 2 el programa funciona correctamente.



```
C:\Windows\system32\cmd.exe - java Chat Prueba
C:\Users\josue\Escritorio\Distribuidos\Tareas>chcp 850
Página de códigos activa: 850
C:\Users\josue\Escritorio\Distribuidos\Tareas>java Chat Prueba
Ingrese el mensaje a enviar:
¿
Ingrese el mensaje a enviar:
Prueba:¿
á
Ingrese el mensaje a enviar:
Prueba:á
```

Figura 2: Prueba usando la página 850 e IBM850.

Para llegar a lo anterior tuve que pasar por varias pruebas y básicamente por el método del ensayo y error, de hecho la solución a este problema cuando estaba probando el programa en la máquina virtual. Me tardé poco más de 4 horas en total para poder solucionar este problema, la ventaja fue que el costo de la máquina virtual era de menos de 1 peso por hora de uso por lo que si comparamos los casi 4 pesos que se usaron para esta práctica contra los 100 dólares que nos da el correo institucional es casi nada.

Ahora el siguiente punto interesante fue como recibir el mensaje que escribirá el usuario, a través de mi experiencia de trabajo con Java solo vi 2 alternativas, usar Scanner o usar BufferedReader, por lo que se y recuerdo Scanner hace lo mismo que BufferedReader al mismo nivel con la misma eficiencia, sin embargo, Scanner tiene muchas funciones más pero el cambio o la diferencia que considero más importante es el tamaño del buffer y es que BufferedReader tiene un tamaño de 8192 caracteres en comparación del Scanner tiene solo 1024 caracteres en el buffer, al ver eso considero mejor usar Scanner ya que al ser un chat y con base a las pruebas a realizar es más que suficiente para este caso. Ahora si vemos el código dado para recibir mensajes necesitamos una longitud del mensaje, lo usual sería usar el mismo tamaño del buffer de Scanner, sin embargo, para estas pruebas sería contraproducente ya que ningún mensaje a enviar tiene esa longitud y esto provocaría que se imprimieran caracteres vacíos, en mi opinión lo mejor es ajustar el tamaño con base a los requerimientos de la práctica, además de que al momento de las pruebas el chat se vea mejor, por lo que el valor a usar será 50 ya que es el tamaño suficiente para la prueba, en caso de querer enviar mensajes más largos se tendrá que cambiar la variable `BUFSIZ` a lo que se necesite pero como valor máximo 1024 ya que si ponemos un valor más grande leeremos los 1024 y los caracteres restantes no se imprimirán, en caso de que necesitemos enviar mensajes más grandes tendríamos cambiar la forma de lectura en consola.

2.1. Configuración de maquina virtual

Ahora vayamos a la creación de la maquina virtual la cual tendrá que tener las siguientes características.

- SO: Windows Server 2012.
- Nombre de la maquina virtual: W2019630428

Microsoft Azure

Inicio > Máquinas virtuales > Máquinas virtuales

Instituto Politécnico Nacional (correo.ipn.mx)

+ Crear

Filtrar por cualquier ca...

Nombre ↑ Suscripción ↑

No hay máquinas virtuales para mostrar

Crear una máquina virtual que ejecute Linux o Windows. Seleccione una imagen de Marketplace o use una imagen personalizada propia.

Más información acerca de Windows Virtual Machines o Más información sobre Linux Virtual Machines o

Crear una máquina virtual

Detalles de instancia

Nombre de máquina virtual *

Región *

Opciones de disponibilidad

Imagen *

Ver todas las imágenes

Instancia de Azure de acceso puntual ☐

Tamaño *

Ver todos los tamaños

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.

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

Seleccionar puertos de entrada *

Esto permitirá que todas las direcciones IP accedan a la máquina virtual.

Revisar y crear

< Anterior

Siguiendo: Discos >

Figura 3: Información básica.

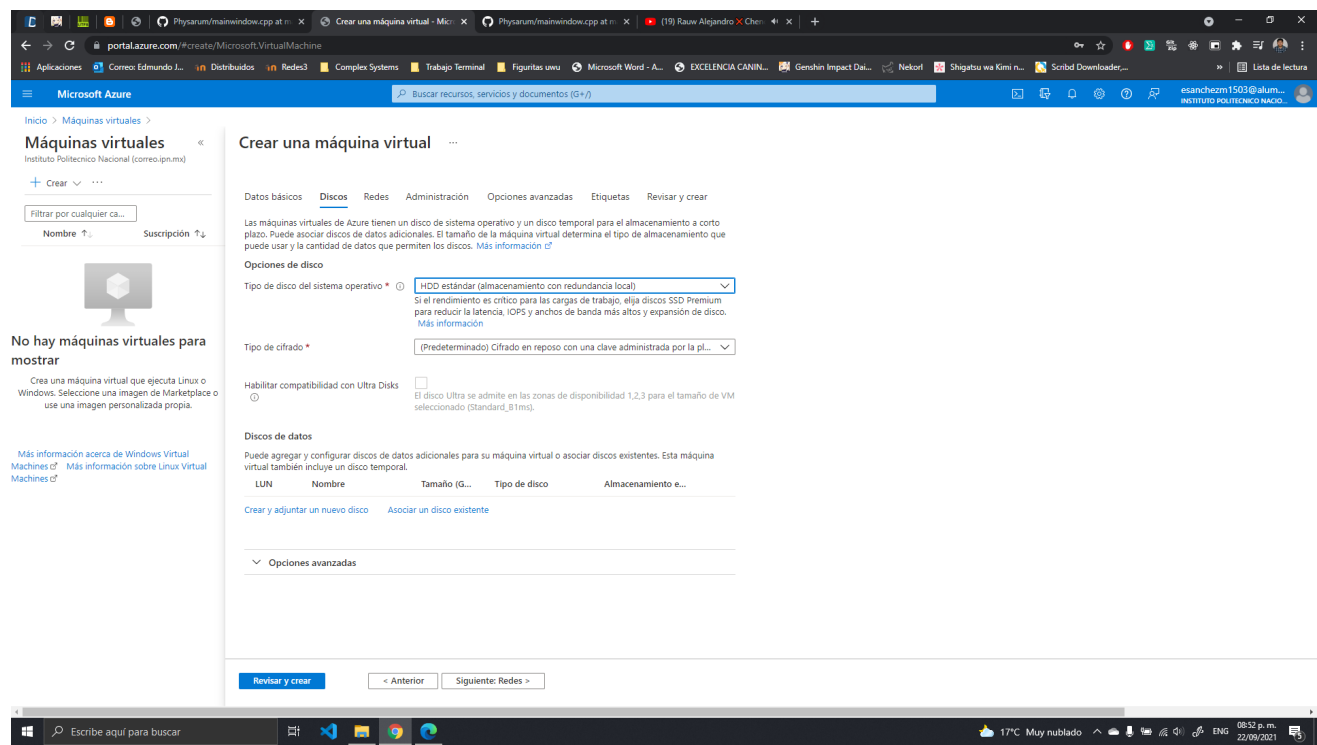


Figura 4: Configuración del tipo de disco.

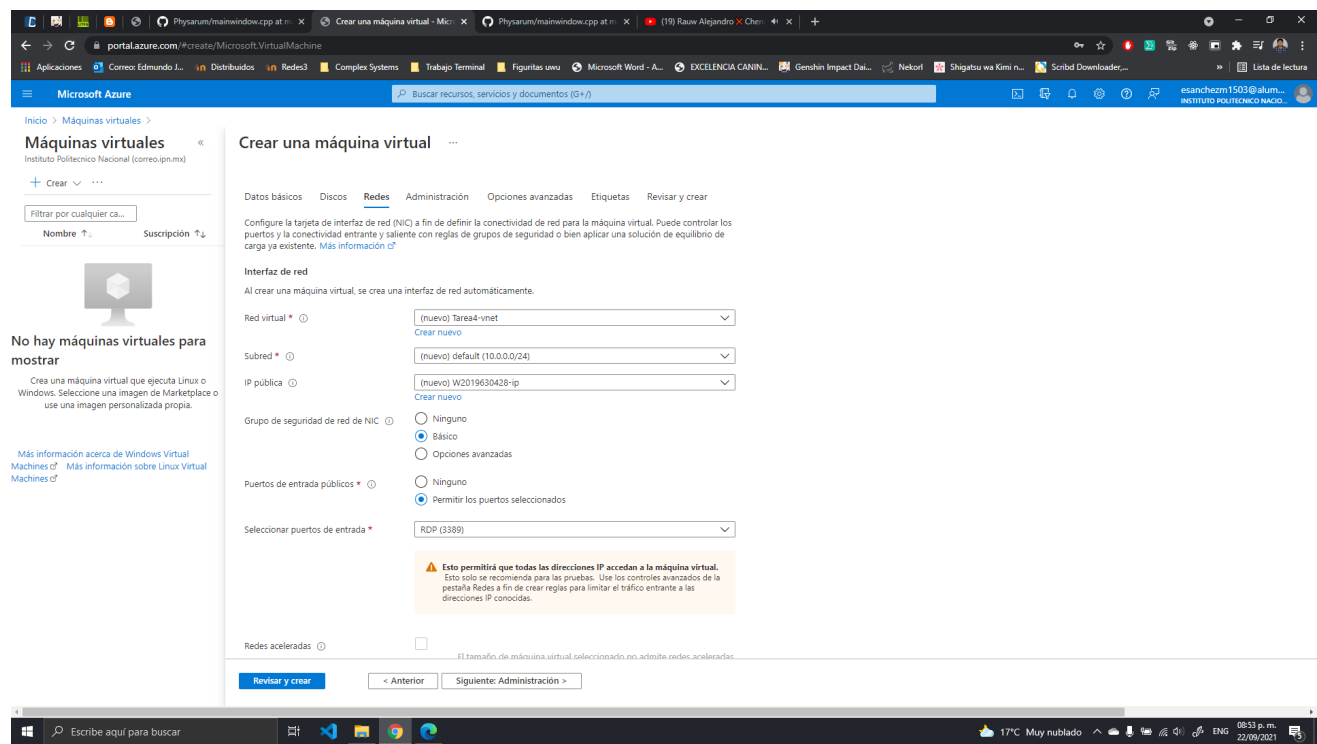


Figura 5: Información sobre la redes.

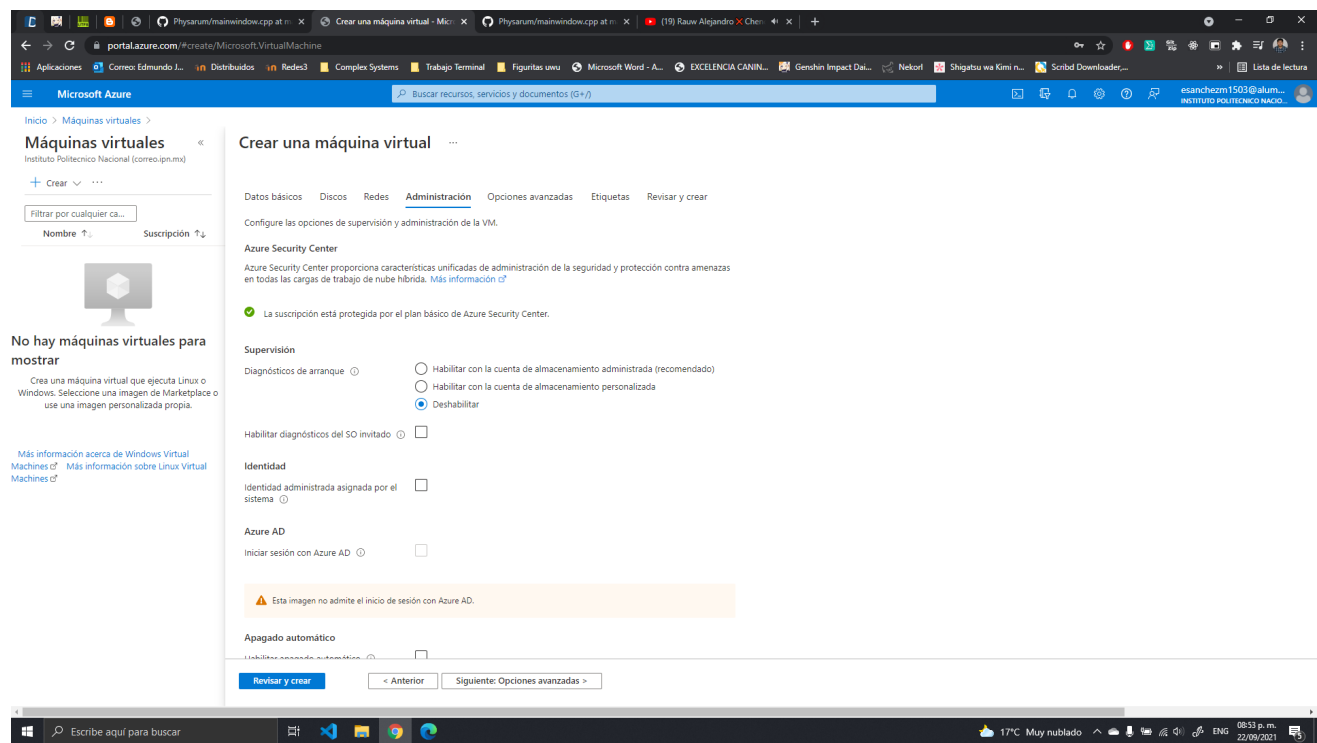


Figura 6: Configuración de la administración.

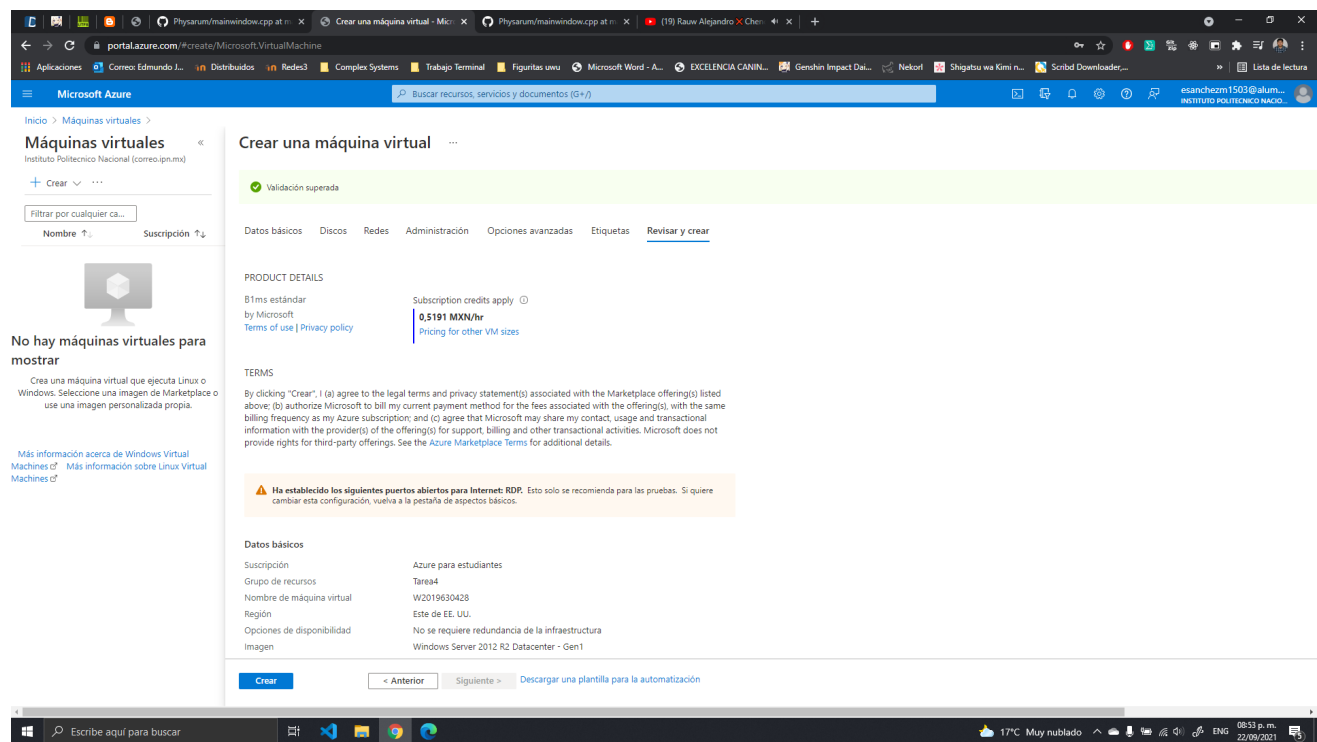


Figura 7: Creación de la maquina virtual.

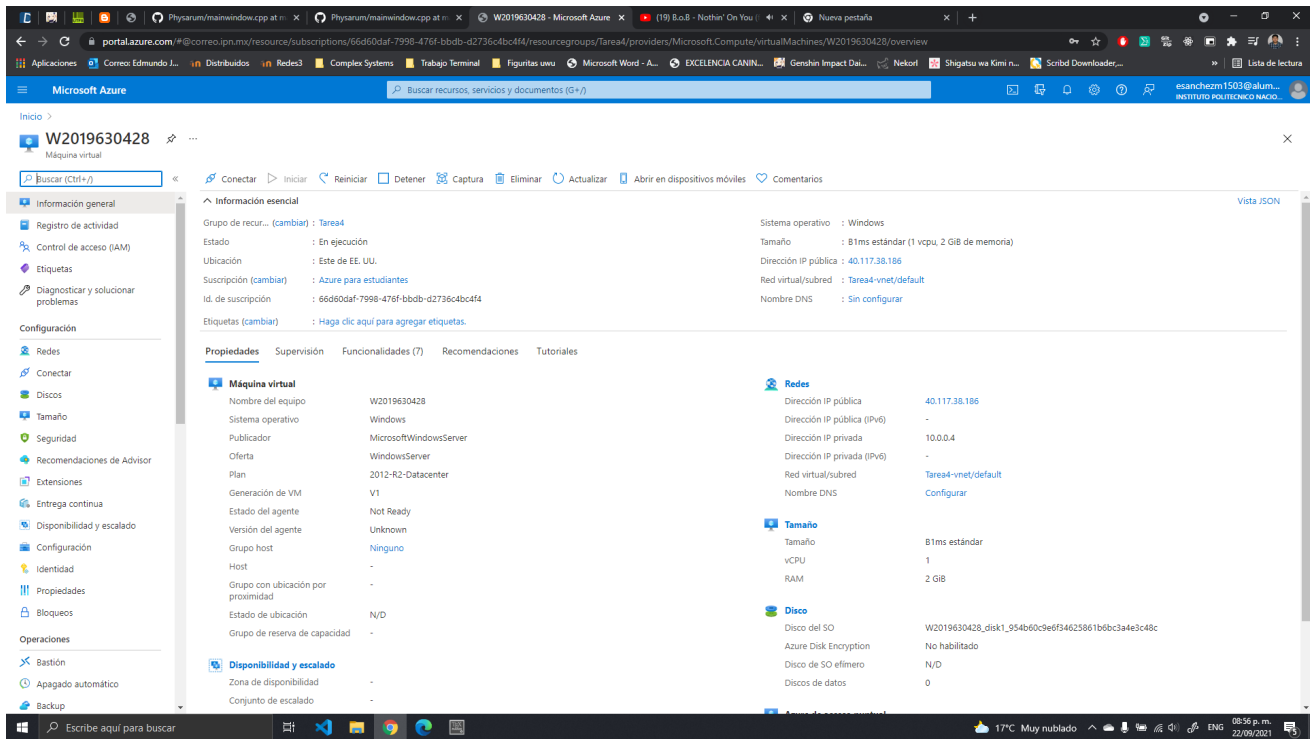


Figura 8: Panel de control de la maquina virtual.

Una vez creada la máquina virtual se procedió ir al apartado se accedido al apartado de conectar para poder descargar nuestro archivo RPD que es necesario para hacer la conexión con nuestra máquina virtual, esto se puede ver en la imagen 9.

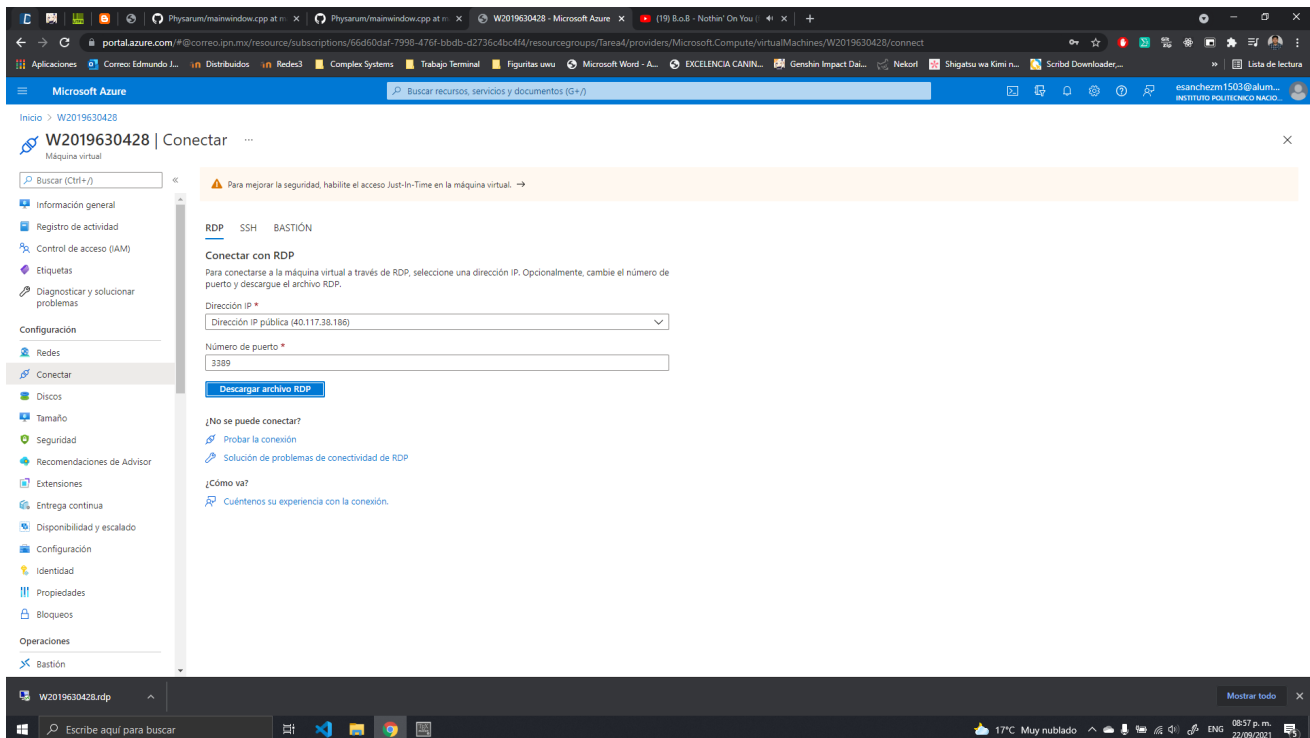


Figura 9: Descarga del archivo RPD.

Mientras se realizaba la descarga del archivo RPD, se procedió a crear un disco lógico local en la unidad F, como dato adicional, la figura 10 tiene diferente fecha ya que el desarrollo de la practica empezó en el inicio de la semana de entrega.

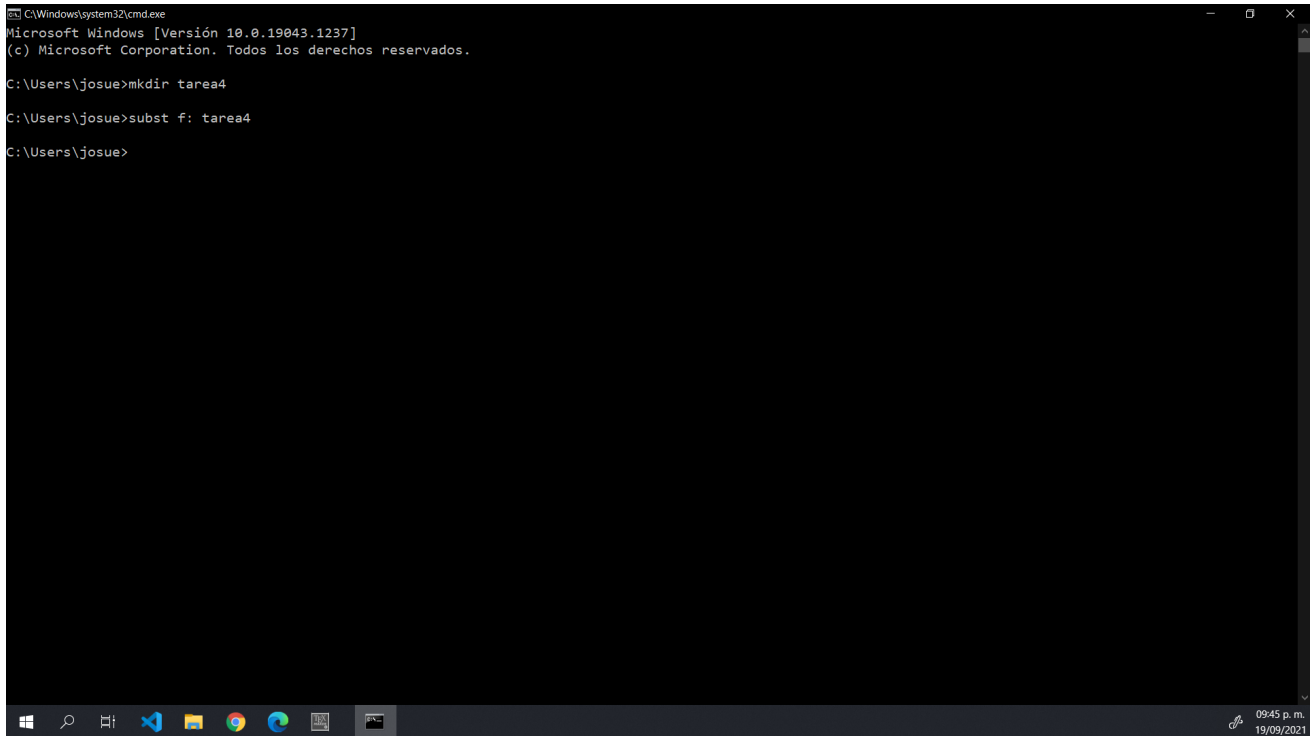


Figura 10: Creación de disco lógico.

Al tener todo listo se procede a editar nuestro archivo RPD para compartir recursos, se compartirá el disco lógico recientemente creado, una vez editado se procede a realizar la conexión con nuestra maquina virtual, todo lo anterior se puede ver de la figura 11 a la 13. Mientras se realizaba lo anterior decidí aprovechar el tiempo y descargar el JDK 8u202 para poder instalarlo en la maquina virtual como se ve en la figura 14, una vez terminada la descarga copie el JDK así como el archivo Chat.java en nuestro disco lógico (figura 15).

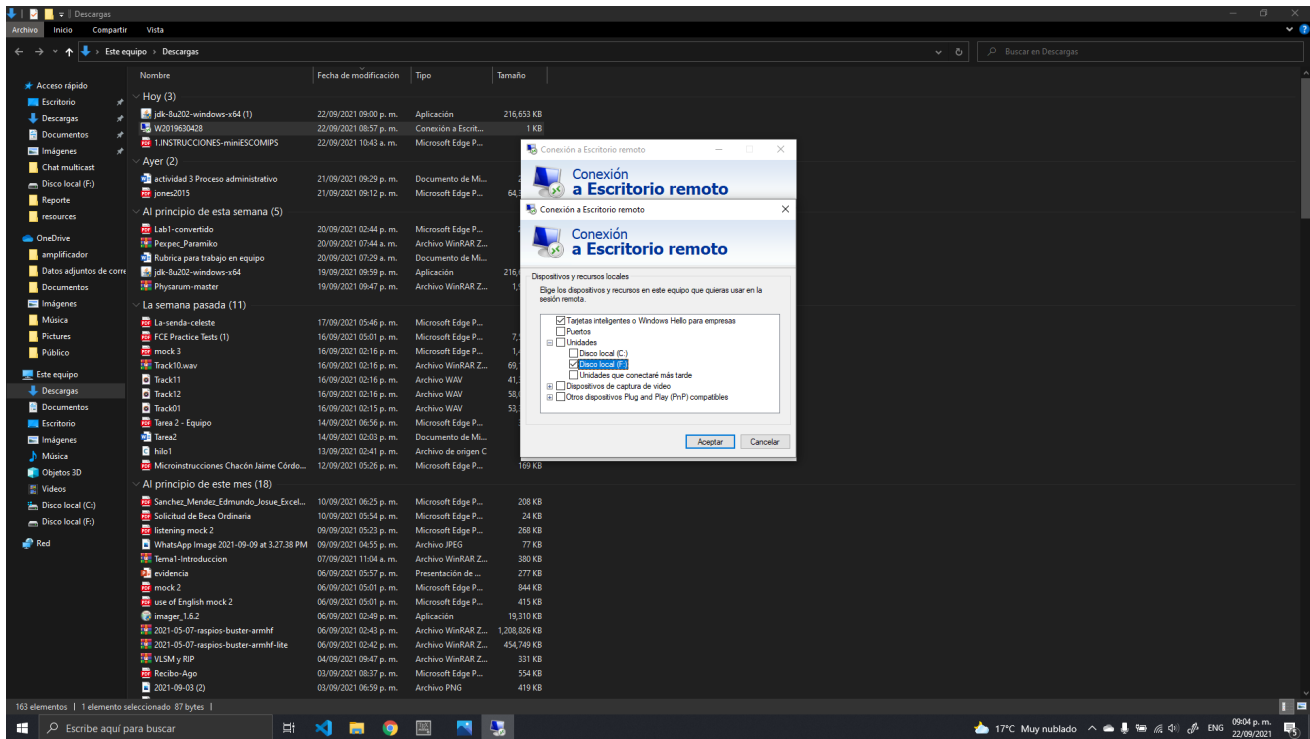


Figura 11: Editando archivo RPD para compartir nuestro disco lógico.

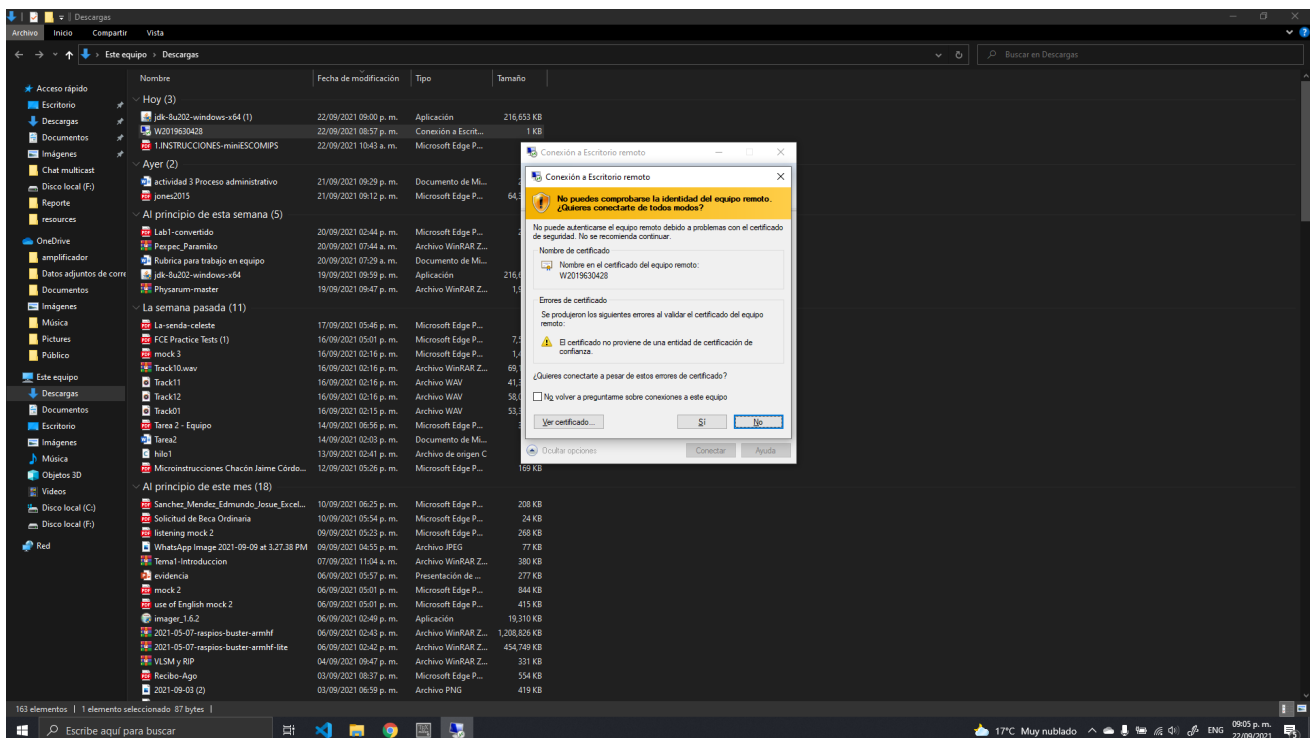


Figura 12: Inicio de conexión con la maquina virtual.

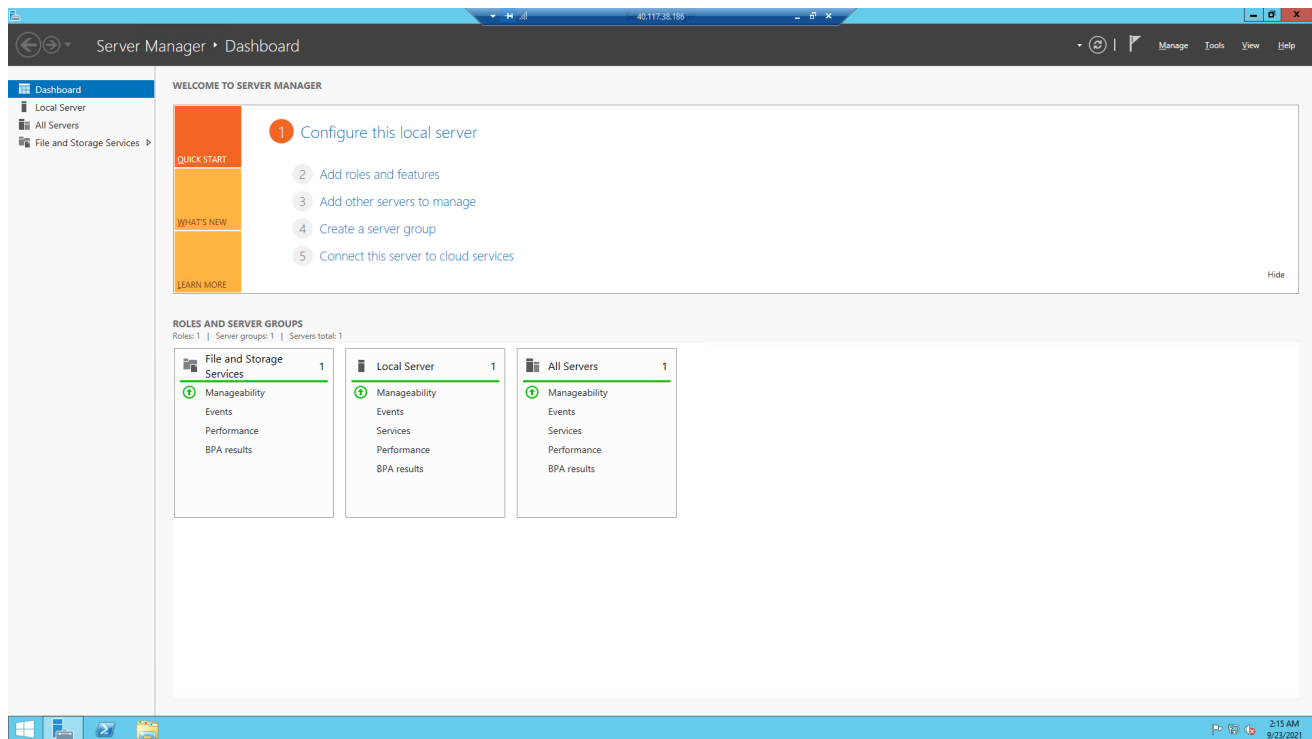


Figura 13: Conexión realizada de manera exitosa con la maquina virtual.

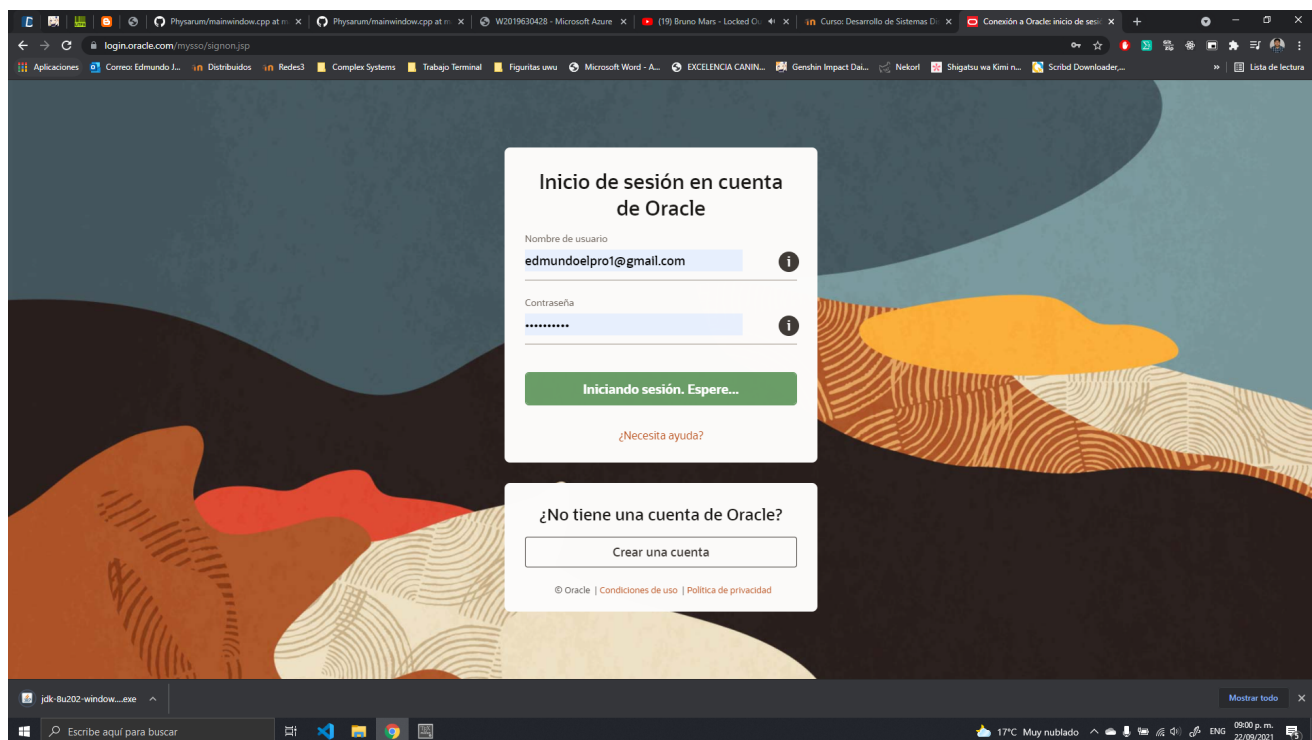


Figura 14: Descarga del JDK 8u202.

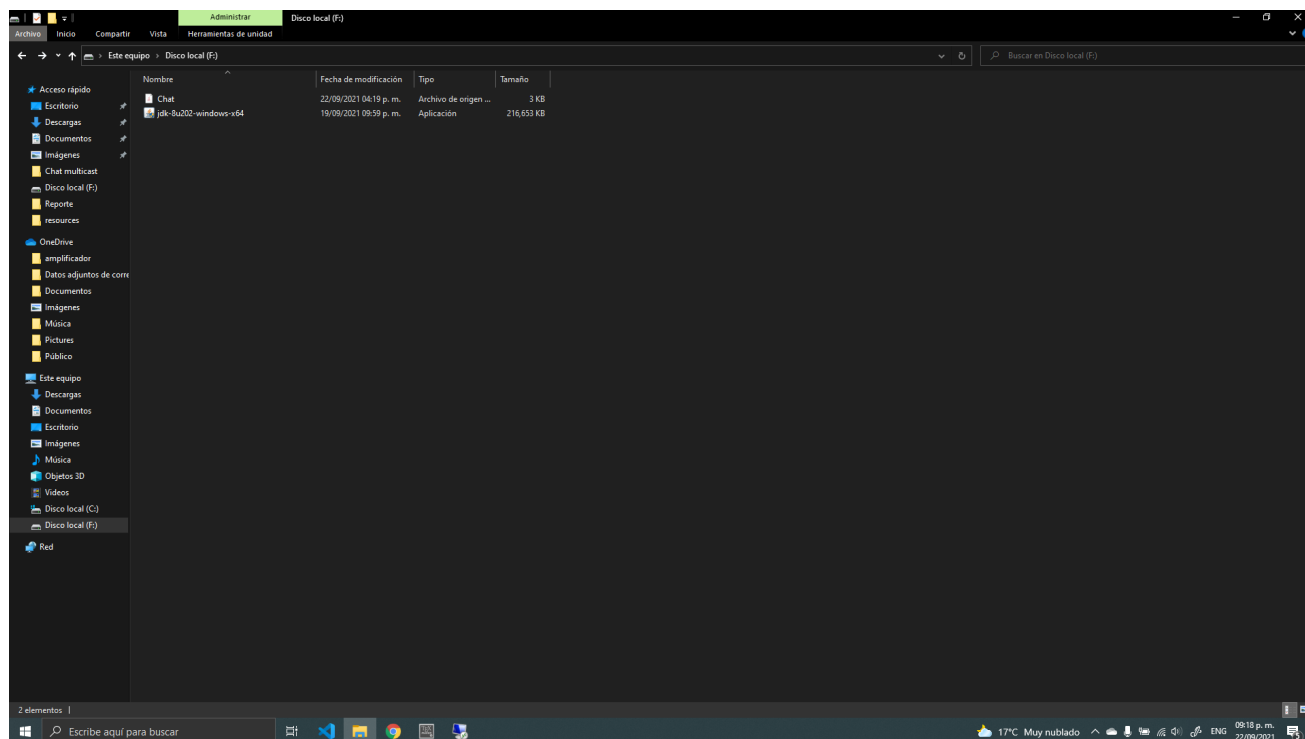


Figura 15: Contenido del disco lógico.

Me gustaría comentar que no me fue posible tomar la screenshot en donde saliera tanto la máquina virtual como mi barra de tareas, ya que no se me permitía cambiar la resolución de la máquina y cuando lo hacía de manera manual no se visualizaba por completo la máquina virtual. Una vez hecho todo lo anterior solo nos queda instalar Java, agregar `JAVA_HOME` como variable de sistema, agregarla a path, reiniciar, además de añadir el distribución del teclado latinoamericano para realizar las pruebas y eso sería todo. Esto se puede ver desde la figura 16 a la 20.

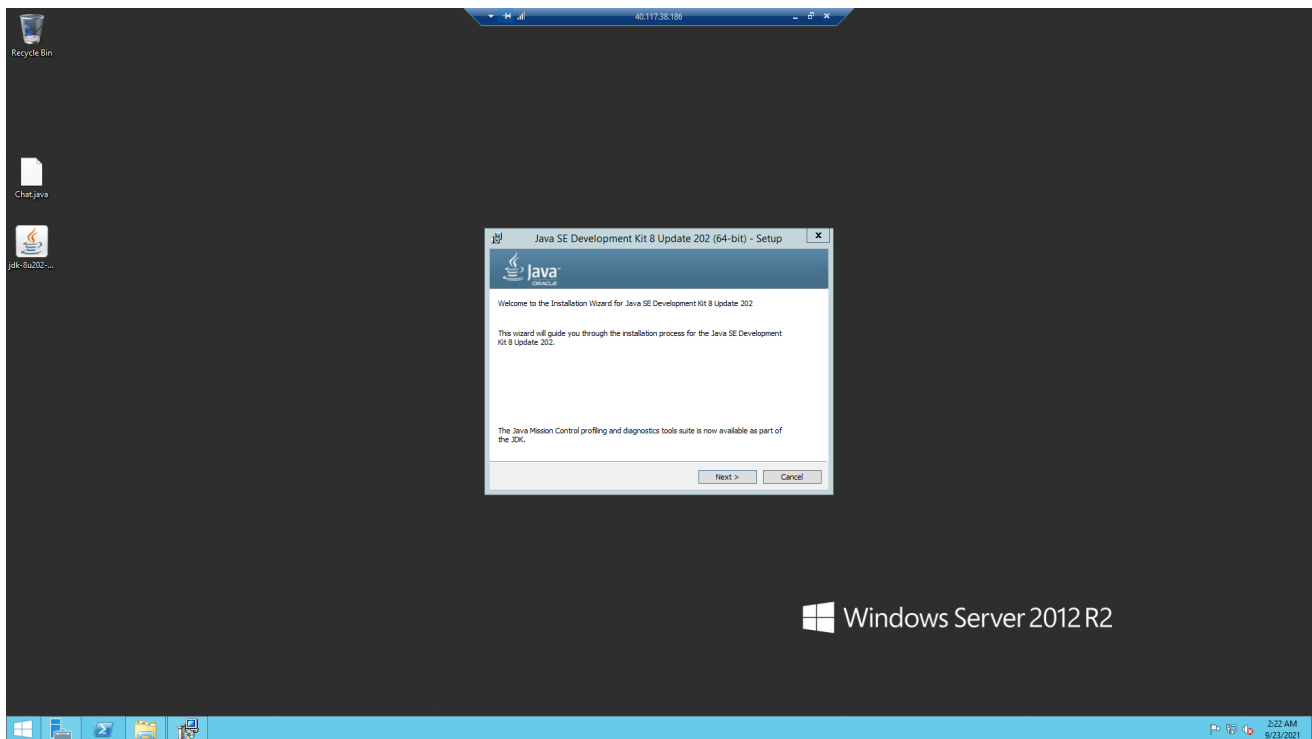


Figura 16: Iniciando instalación de Java.

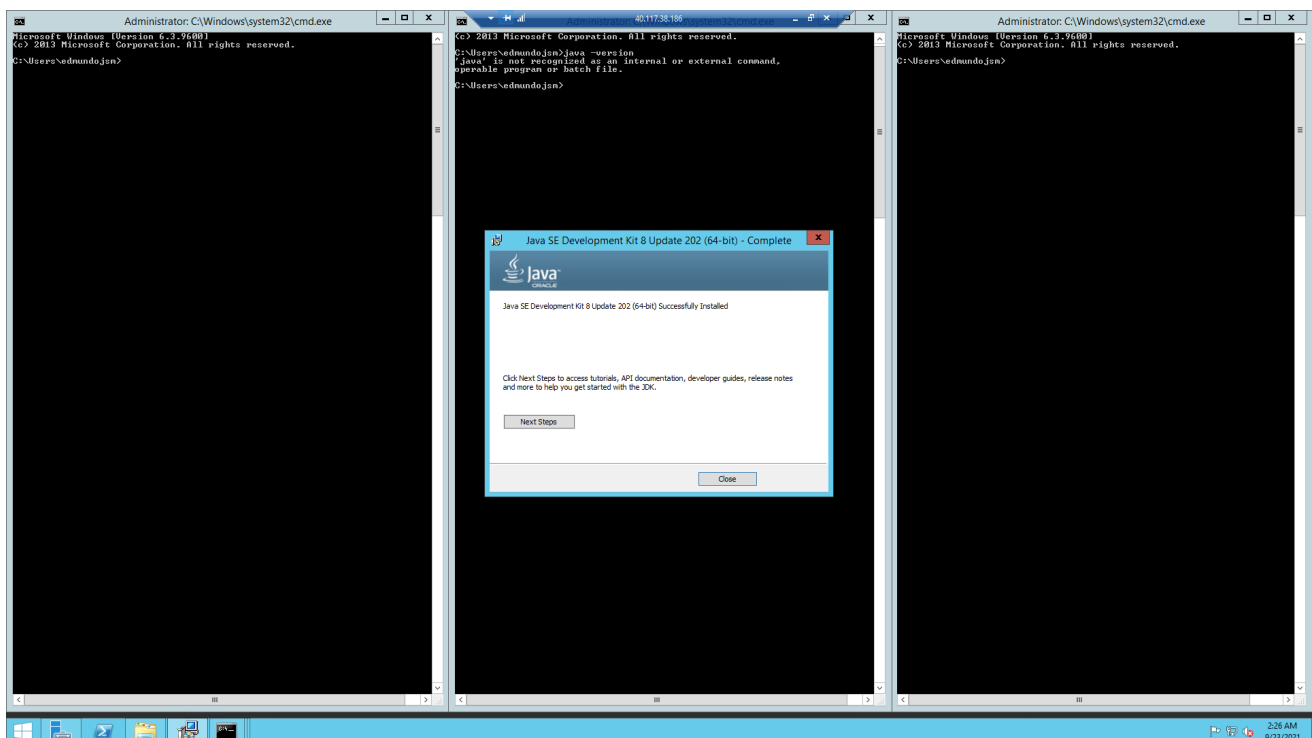


Figura 17: Fin de la instalación de Java.

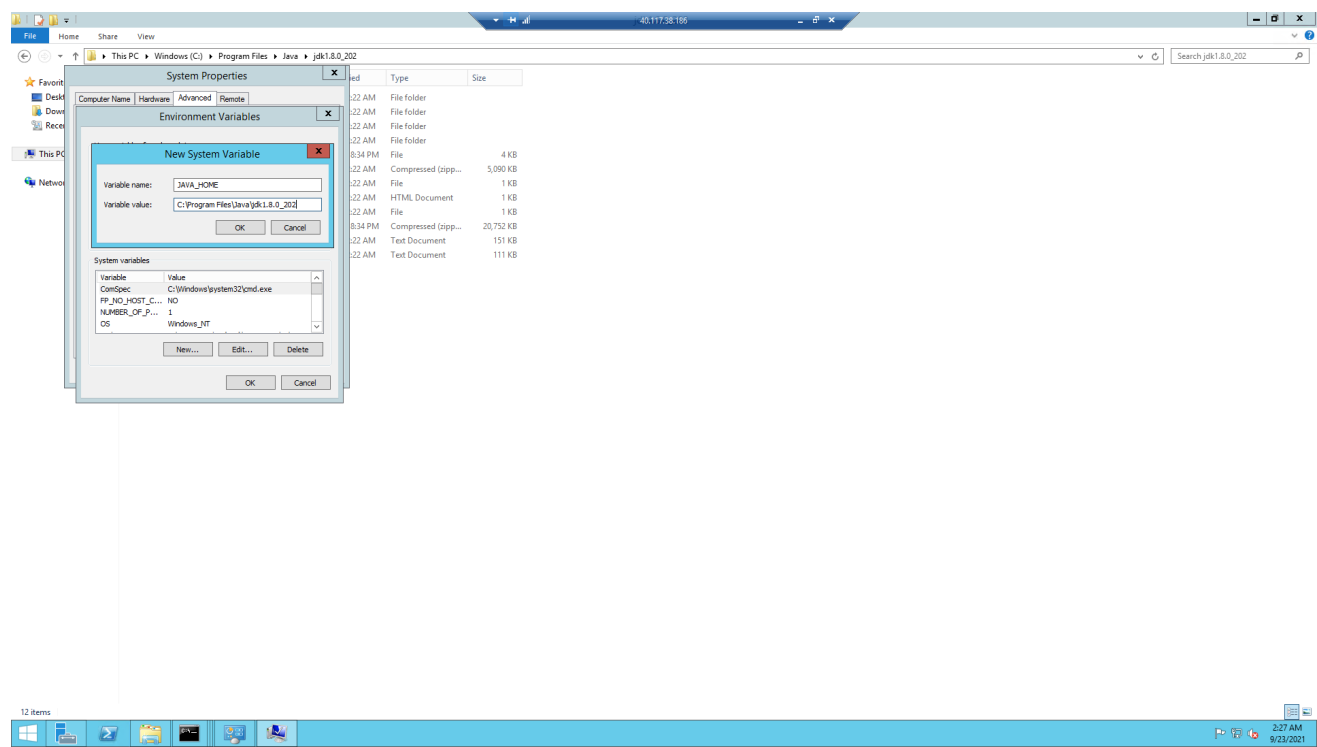


Figura 18: Añadiendo JAVA_HOME.

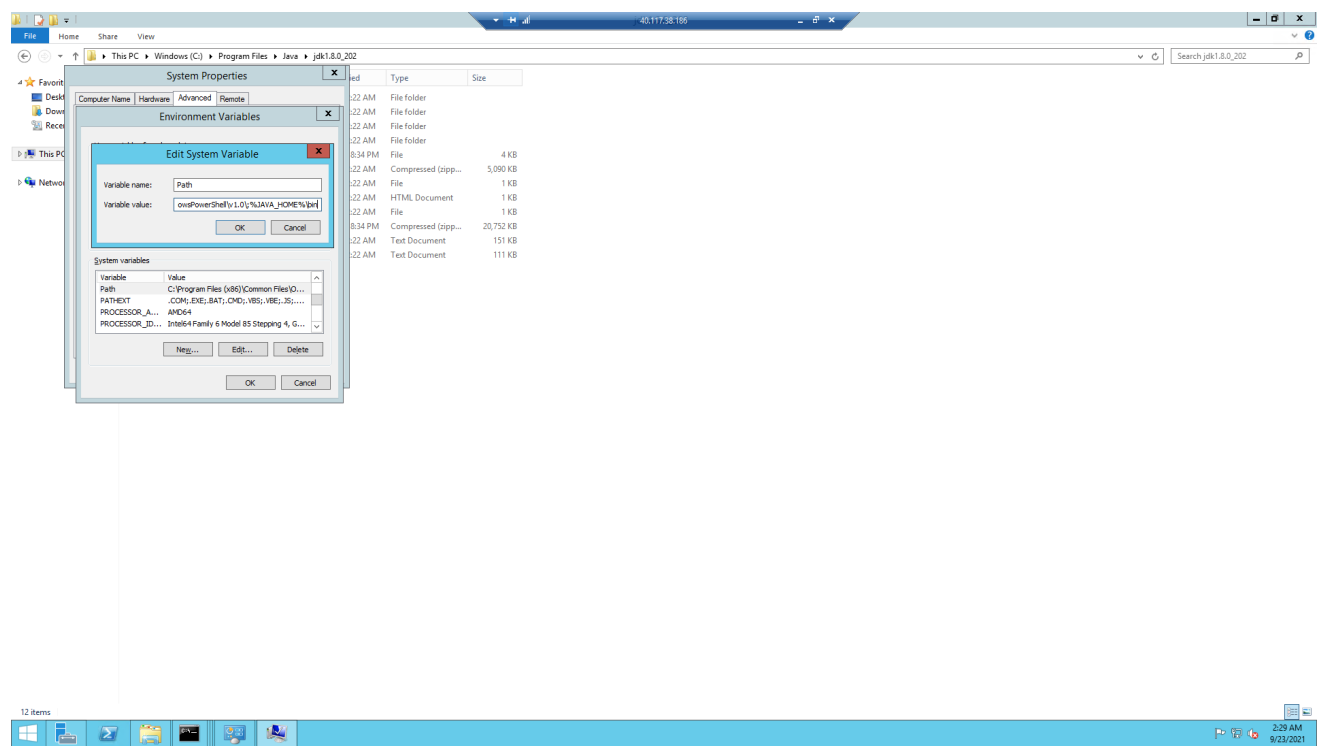


Figura 19: Actualizando variable path.

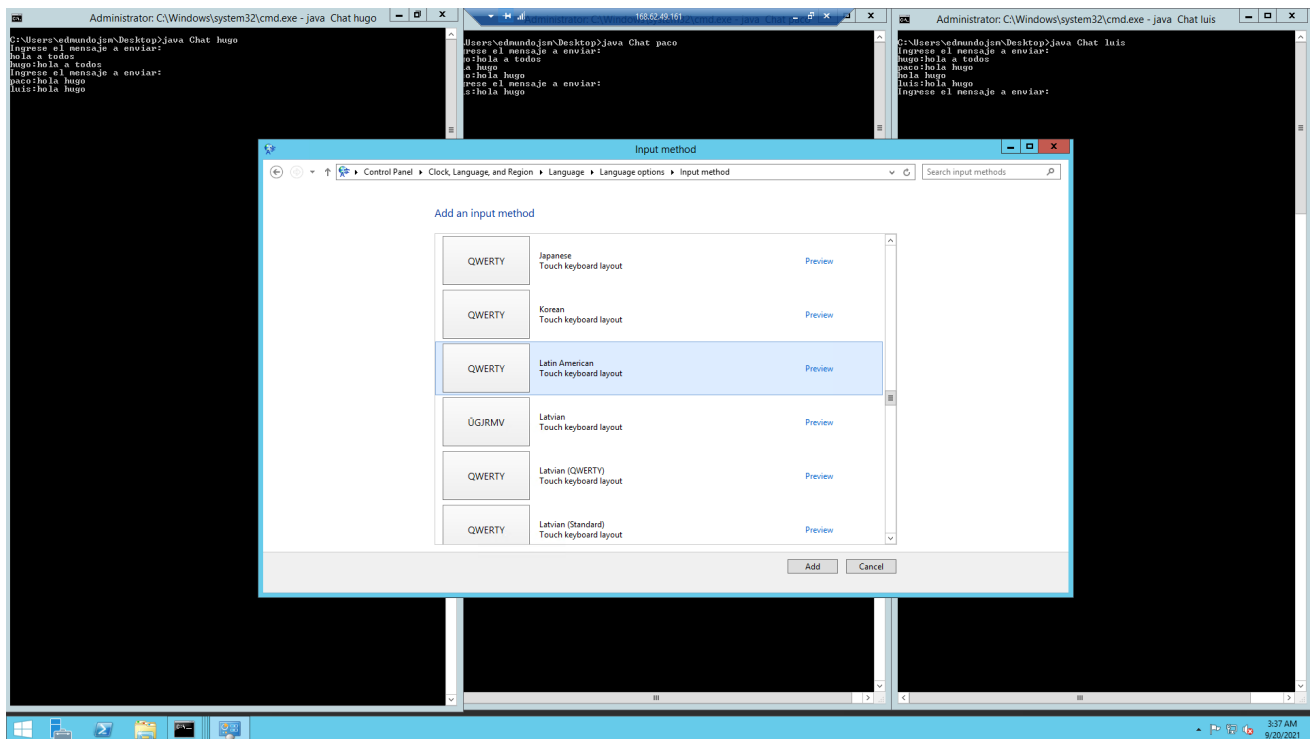


Figura 20: Agregando distribución de teclado.

2.2. Compilación del código

Al inicio del reporte comente que inicialmente use el código de pagina 1252 y UTF-8 para la practica pero esto no funciona, de hecho como podemos ver en la figura 21 vemos unos errores en el CMD de codificación incorrecta, por otro lado tenemos el PowerShell que si bien la codificación de entrada era correcta el despliegado no, al investigar resulta ser que esos cuadros salen ya que la codificación del carácter no se encuentra definida en la pagina de código, esto se puede ver si usamos el archivo Chatv1.java claramente este nombre tendra que cambiar a Chat ya que el nombre de la clase debe de ser el mismo que el nombre del archivo.

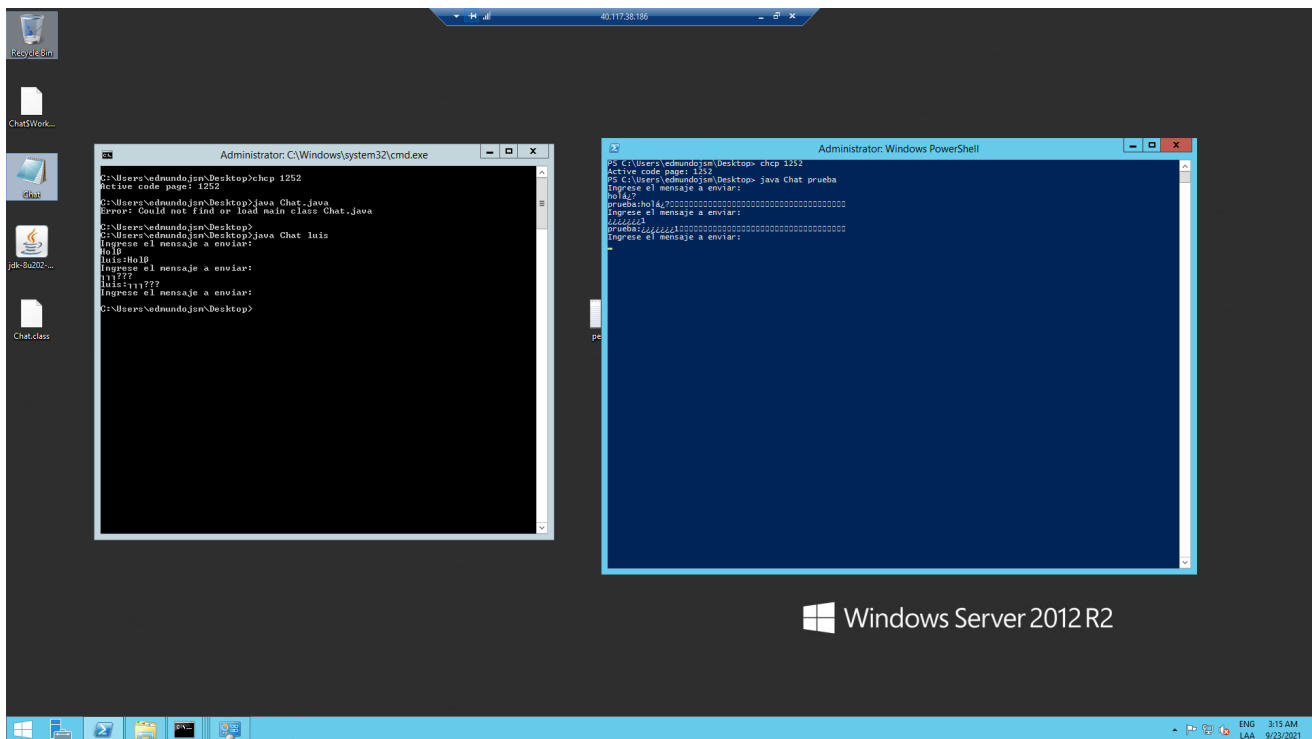


Figura 21: Errores usando pagina de código 1252 y UTF8.

Ahora como vemos en la figura 22 vemos que el código de pagina por defecto de Windows Server 2012 es la 437, al buscar la tabla de caracteres definidos vemos que contiene los caracteres especiales de la practica por lo que en lugar de modificar el código de pagina lo mejor es adaptar el programa.

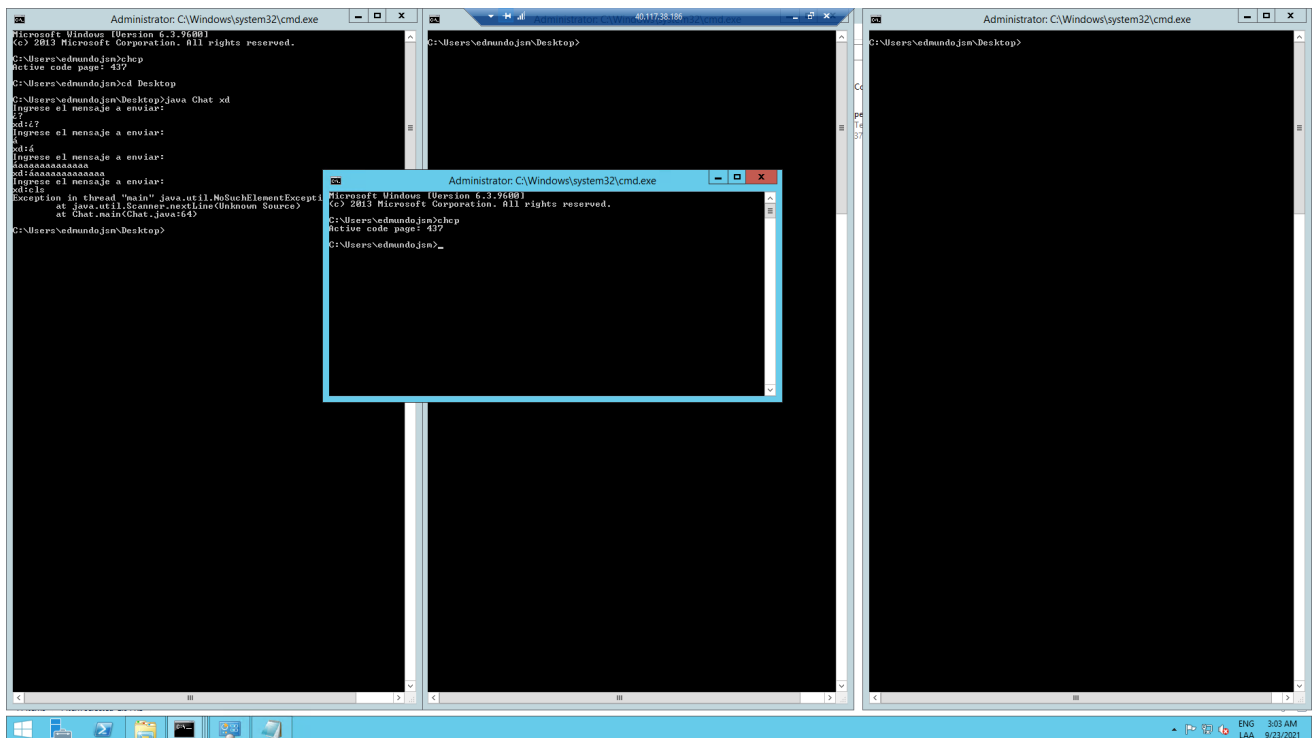


Figura 22: Código de pagina por defecto.

Ahora si en la imagen 23 podemos ver la compilación del archivo Chat.java y que además Java esta correctamente instalado en nuestra maquina.

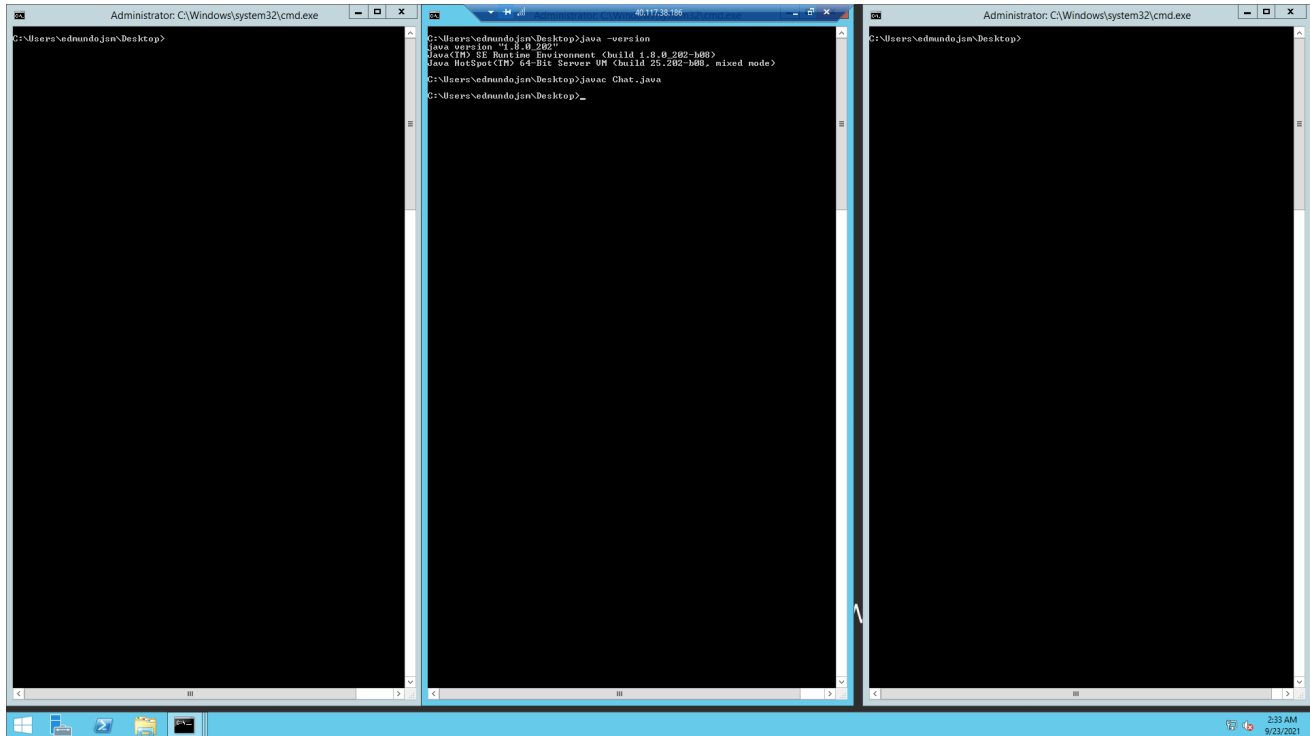


Figura 23: Compilación e instalación de Java correctas.

Una vez compilado nuestro programa se procedió a realizar la conversación solicita, adicional a esto en la misma figura, la cual es la figura 24 podemos ver como también se esta utilizando el código de pagina por defecto.



En esta práctica desarrollamos un cliente multicast, pero como se nos menciona en clase el desarrollo de este tipo de servidores es muy sencilla, ya que todos los mensajes que se envían serán recibidos por todos los dispositivos conectados a la misma red y al mismo puerto, realmente no hay mayor problema en el desarrollo del chat. Realmente lo tardó fue lo que vimos a través de este reporte que es el código de página y que tipo de codificación le corresponde, también hacer la elección de cómo leer la entrada del teclado. Finalmente me gustaría recordar que en el archivo comprimido se encuentra el archivo que lleva como nombre Chat.java es el que se usó para realizar la ejecución del programa, sin embargo, también se encuentra un archivo Chatv1.java que es el archivo que se desarrolló primeramente con el uso del código de página 1252 y utf8 mencionar que si se quiere usar este archivo se tendrá que modificar el nombre a Chat ya que si no se hace eso el nombre de la clase no coincidirá y Java no nos dejara compilar el archivo.

Oracle. n.d. Supported Encodings. [online] Available at: <https://docs.oracle.com/javase/8/docs/technotes/guides/intl/>
[Accessed 23 September 2021].

ESCOM-IPN

<https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html> [Accessed 23 September 2021].

Es.wikipedia.org. 2020. Windows-1252 - Wikipedia, la enciclopedia libre. [online] Available at: <https://es.wikipedia.org/wiki/Windows-1252> [Accessed 23 September 2021].

Es.wikipedia.org. 2021. Página de códigos 437 - Wikipedia, la enciclopedia libre. [online] Available at: https://es.wikipedia.org/wiki/P%C3%A1gina_de_c%C3%B3digos_437 [Accessed 23 September 2021].

Es.wikipedia.org. 2020. Página de códigos 850 - Wikipedia, la enciclopedia libre. [online] Available at: https://es.wikipedia.org/wiki/P%C3%A1gina_de_c%C3%B3digos_850 [Accessed 23 September 2021].