



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Distribuidos

Profesor:

Carlos Pineda Guerrero

Asignatura:

Tarea 1. Implementación de un proxy inverso
HTTPS y servidores HTTP en la nube.

Grupo:

7CV1

Alumna:

Mateo García Alejandra Michelle

Introducción

El presente documento aborda la implementación práctica de una arquitectura de sistemas distribuidos centrada en un proxy inverso HTTPS y múltiples servidores HTTP de backend, desplegada en un entorno de computación en la nube (Microsoft Azure).

Un proxy inverso es un componente fundamental que actúa como intermediario a nivel de aplicación (Capa 7 de OSI), situándose estratégicamente frente a los servidores de origen. Su función principal es aceptar las conexiones entrantes de los clientes y redirigirlas a uno o varios servidores backend. Esta arquitectura centralizada ofrece beneficios críticos en entornos de producción:

- **Seguridad:** Oculta la topología interna de la red y la identidad de los servidores de origen, actuando como una primera capa de defensa contra ataques cibernéticos. Además, facilita la terminación SSL/TLS (SSL Offloading), centralizando la gestión de certificados y liberando recursos de cómputo en los servidores de contenido.
- **Balanceo de Carga:** Permite la distribución eficiente del tráfico entre múltiples servidores, asegurando la alta disponibilidad y la escalabilidad horizontal del servicio.
- **Optimización del Rendimiento:** Soporta el almacenamiento en caché de contenido estático y la compresión de respuestas para acelerar la entrega de datos al cliente.

La tarea requirió el desarrollo y despliegue de: 1) un servidor HTTP/1.1 con soporte para el mecanismo de caché condicional (utilizando los encabezados Last-Modified e If-Modified-Since); 2) un proxy inverso en Java (AdministradorTráfico.java y AdministradorTráficoSSL.java) para gestionar peticiones HTTP y HTTPS y demostrar la capacidad de redirección de tráfico selectiva a dos backends ; y 3) la configuración de un entorno de máquinas virtuales con Ubuntu Server y Windows Server en la plataforma Azure, incluyendo la configuración de reglas de puerto (iptables) y certificados autofirmados.

Desarrollo

Para el desarrollo de esta práctica es necesario hacer la Implementación del uso de la caché. El servidor deberá enviar al navegador en cada respuesta el encabezado Last-Modified y deberá procesar el encabezado If-Modified-Since que envíe al navegador e implementar la funcionalidad de servidor web. Cuando el servidor reciba una petición "GET /archivo.html HTTP/1.1" se deberá leer el archivo.html del disco local y se deberá regresar al navegador el contenido del archivo con el tipo mime "text/html". En este caso nos apoyamos de la herramienta Chatgpt que nos modifico el archivo de servidor HTTP: <https://chatgpt.com/share/68cf9460-a9d0-8002-91fd-03ddd0c4f708>.

Para el siguiente programa AdministradorTrafico.java es un proxy inverso. Lo que hace es ponerse en medio entre el navegador y dos servidores. Cuando el navegador pide algo (con un GET), el proxy recibe esa petición y la manda a los dos servidores que tiene configurados. Luego, el proxy recibe las respuestas de ambos, pero solo le devuelve al navegador la del Servidor-1; la del Servidor-2 no la manda, aunque sí la recibe. La idea es que aprendas cómo un proxy puede manejar el tráfico, redirigir peticiones y decidir qué respuesta darle al cliente. De igual manera se realizó con chatgpt: <https://chatgpt.com/share/68cf9460-a9d0-8002-91fd-03ddd0c4f708>.

Como se indica en la tarea, se creo las 3 máquinas virtuales, todas con Ubuntu server, la maquina 1 fue nuestro cliente, este se conectaba a las máquinas 2 y 3 que fungían como los servidores HTTPS.

Microsoft Azure portal (PWA) - Crear una máquina virtual - Microsoft Azure

Microsoft Azure

Buscar recursos, servicios y documentos (G+ /)

Copilot

amateog1700@alumno... INSTITUTO POLITECNICO NACIO...

Inicio > T1 > Marketplace > Máquina virtual >

Crear una máquina virtual

Ayuda para crear una máquina virtual de bajo coste Ayuda para crear una VM optimizada para alta disponibilidad +1

Ayuda para crear una máquina virtual de bajo coste Ayuda para crear una VM optimizada para alta disponibilidad Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo

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 (748b53d1-496b-42b5-b738-76b692ae0a51)

Grupo de recursos * T1

Crear nuevo

Detalles de instancia

Nombre de máquina virtual * T1-UBUNTU-2021640448-1

Región * (Canada) Canada Central

Implementación en una zona extendida de Azure

Opciones de disponibilidad Zona de disponibilidad

Opciones de zona

☒ Zona autoseleccionada

Elija hasta 3 zonas de disponibilidad, una máquina virtual por zona

☐ Zona seleccionada por Azure (versión preliminar)

< Anterior Siguiente: Discos > Revisar y crear

Enviar comentarios

12:59 a. m.

Se muestra que se abrieron los puertos 443 y 20.

The screenshot shows the 'Crear una máquina virtual' (Create a virtual machine) page in the Microsoft Azure portal. The user is logged in as 'amateog1700@alumno... INSTITUTO POLITECNICO NACIO...'. The page is in the 'Reglas de puerto de entrada' (Inbound ports) step. The 'Puertos de entrada públicos' (Public inbound ports) section shows 'Ninguno' (None) selected, with a radio button for 'Permitir los puertos seleccionados' (Allow selected ports) also present. The 'Seleccionar puertos de entrada' (Select inbound ports) dropdown is set to 'HTTP (80), HTTPS (443), SSH (22)'. A warning message states: '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.' (This will allow all IP addresses to access the virtual machine. This is only recommended for testing. Use the advanced controls in the Networks tab to create rules to limit incoming traffic to known IP addresses.) The bottom navigation bar shows '< Anterior' (Previous), 'Siguiendo: Discos >' (Next: Disks), and 'Revisar y crear' (Review and create).

Se seleccionaron discos HDD de 30 GiB

The screenshot shows the 'Crear una máquina virtual' (Create a virtual machine) page in the Microsoft Azure portal, specifically the 'Disco del SO' (OS Disk) configuration step. The user is logged in as 'amateog1700@alumno... INSTITUTO POLITECNICO NACIO...'. The 'Disco del SO' section shows 'Valor predeterminado de la imagen (30 GiB)' (Default image value (30 GiB)) for the size. The 'Tipo de disco del sistema operativo' (Operating system disk type) is set to 'HDD estándar (almacenamiento con redundancia local)' (Standard HDD (local redundancy storage)). A note states: 'El tamaño de la máquina virtual seleccionada es compatible con los discos premium. Se recomienda SSD Premium para elevadas cargas de trabajo de E/S por segundo. Las máquinas virtuales con discos SSD Premium optan al acuerdo de nivel de servicio de conectividad del 99,9%.' (The size of the selected virtual machine is compatible with premium disks. Premium SSD is recommended for high I/O workloads. Virtual machines with Premium SSD disks opt into the 99.9% connectivity service level agreement.) The 'Eliminar con VM' (Delete with VM) checkbox is checked. The 'Administración de claves' (Key management) dropdown is set to 'Clave administrada por la plataforma' (Platform-managed key). The 'Habilitar compatibilidad con Ultra Disks' (Enable Ultra Disks compatibility) checkbox is unchecked. The bottom navigation bar shows '< Anterior' (Previous), 'Siguiendo: Redes >' (Next: Networks), and 'Revisar y crear' (Review and create).

Resumen de las configuraciones de la máquina Virtual 1.

Microsoft Azure portal (PWA) - Crear una máquina virtual - Microsoft Azure

Microsoft Azure | Buscar recursos, servicios y documentos (G+/I) | Copilot | amateog1700@alumno... INSTITUTO POLITÉCNICO NACIO...

Inicio > Grupos de recursos > T1 > Marketplace > Máquina virtual >

Crear una máquina virtual

Ayuda para crear una máquina virtual de bajo coste | Ayuda para crear una VM optimizada para alta disponibilidad | +1

Validación superada

Ayuda para crear una máquina virtual de bajo coste | Ayuda para crear una VM optimizada para alta disponibilidad | Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo

Datos básicos

Suscripción	Azure for Students
Grupo de recursos	T1
Nombre de máquina virtual	T1-UBUNTU-2021630448-1
Región	West US 2
Opciones de disponibilidad	Zona de disponibilidad
Opciones de zona	Zona autoseleccionada
Zona de disponibilidad	3
Tipo de seguridad	Máquinas virtuales de inicio seguro
Habilitar arranque seguro	Sí
Habilitar vTPM	Sí
Supervisión de integridad	No
Imagen	Ubuntu Server 24.04 LTS - Gen2
Arquitectura de VM	x64

< Anterior | Siguiente > | Crear

Descargar una plantilla para la automatización | Enviar comentarios

Creación de la máquina virtual 2 con Ubuntu Server, este es el servidor 1.

Microsoft Azure portal (PWA) - Crear una máquina virtual - Microsoft Azure

Microsoft Azure | Buscar recursos, servicios y documentos (G+/I) | Copilot | amateog1700@alumno... INSTITUTO POLITÉCNICO NACIO...

Inicio > T1 > Marketplace > Máquina virtual >

Crear una máquina virtual

Ayuda para crear una máquina virtual de bajo coste | Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo | +1

Ayuda para crear una máquina virtual de bajo coste | Ayuda para crear una VM optimizada para alta disponibilidad | Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo

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 (748b53d1-496b-42b5-b738-76b692ae0a51) |

Grupo de recursos * | T1 |

Crear nuevo

Detalles de instancia

Nombre de máquina virtual * | T1-UBUNTU-2021640448-2 |

Región * | |

Implementación en una zona extendida de Azure

Opciones de disponibilidad | No se requiere redundancia de la infraestructura |

Tipo de seguridad | Cargando... |

Configurar características de seguridad

< Anterior | Siguiente: Discos > | Revisar y crear

Enviar comentarios

Resumen general de la máquina virtual 2 que este sería el servidor 1.

Microsoft Azure portal showing the 'Crear una máquina virtual' (Create a virtual machine) page. The page displays basic information for the VM, including subscription, resource group, name, region, and availability options. The 'Crear' (Create) button is visible at the bottom.

Datos básicos	
Suscripción	Azure for Students
Grupo de recursos	T1
Nombre de máquina virtual	T1-UBUNTU-2021630448-3
Región	West US 2
Opciones de disponibilidad	Zona de disponibilidad
Opciones de zona	Zona autoseleccionada
Zona de disponibilidad	3
Tipo de seguridad	Máquinas virtuales de inicio seguro
Habilitar arranque seguro	Sí
Habilitar vTPM	Sí
Supervisión de integridad	No

Para esto se tuvo que abrir el puerto 8080 en las configuraciones de red y permitir la entrada del puerto 8080.

Microsoft Azure portal showing the 'Agregar regla de seguridad de entrada' (Add inbound security rule) dialog box. The rule is configured to allow traffic on port 8080. The background shows the 'Configuración de red' (Network configuration) page for the VM.

Configuración de red

Reglas

Grupo de seguridad de red T1-UBUNTU-2021630448-2241_z3

Afecta a 0 subredes, 1 interfaces de red

Buscar reglas

Origen == todo

Destino

Prioridad

Nombre

Reglas de puerto de entrada (4)

Prioridad	Nombre
300	SSH
65000	AllowVnetInBound
65001	AllowAzureLoadBalancerInBound
65500	DenyAllInBound

Reglas de puerto de salida (3)

Agregar regla de seguridad de entrada

T1-UBUNTU-2021630448-2-nsg

Origen

Any

Intervalos de puertos de origen *

*

Destino

Any

Servicio

Custom

Intervalos de puertos de destino *

8080

Protocolo

☒ Any

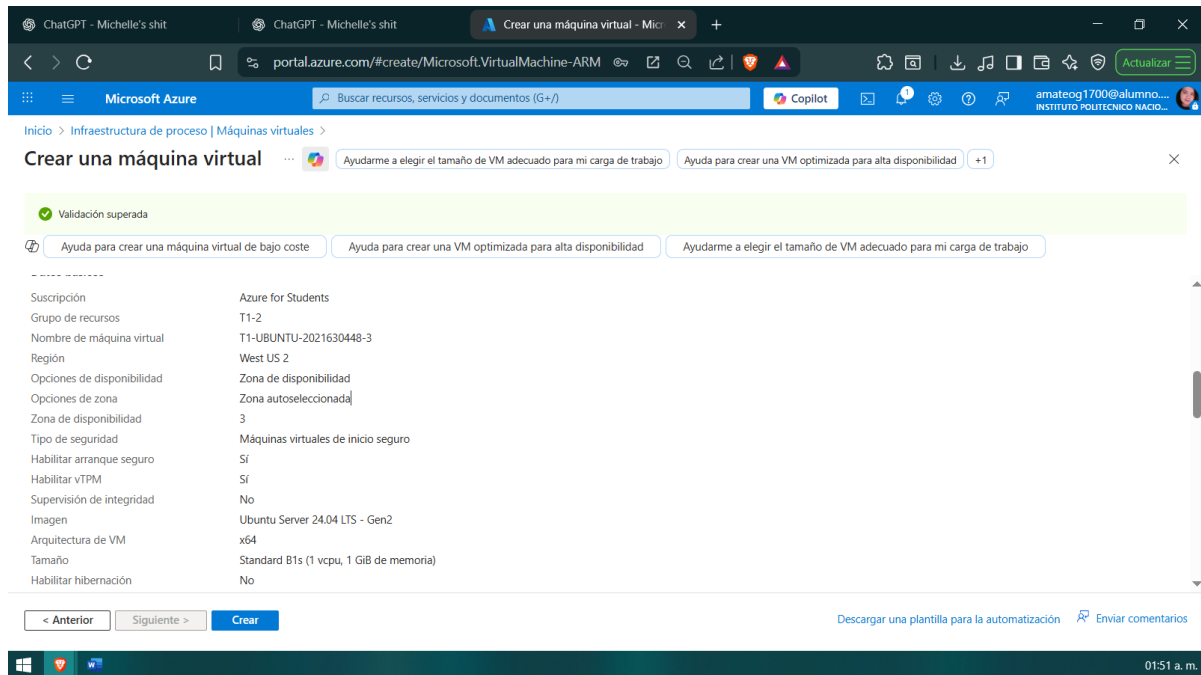
☐ TCP

☐ UDP

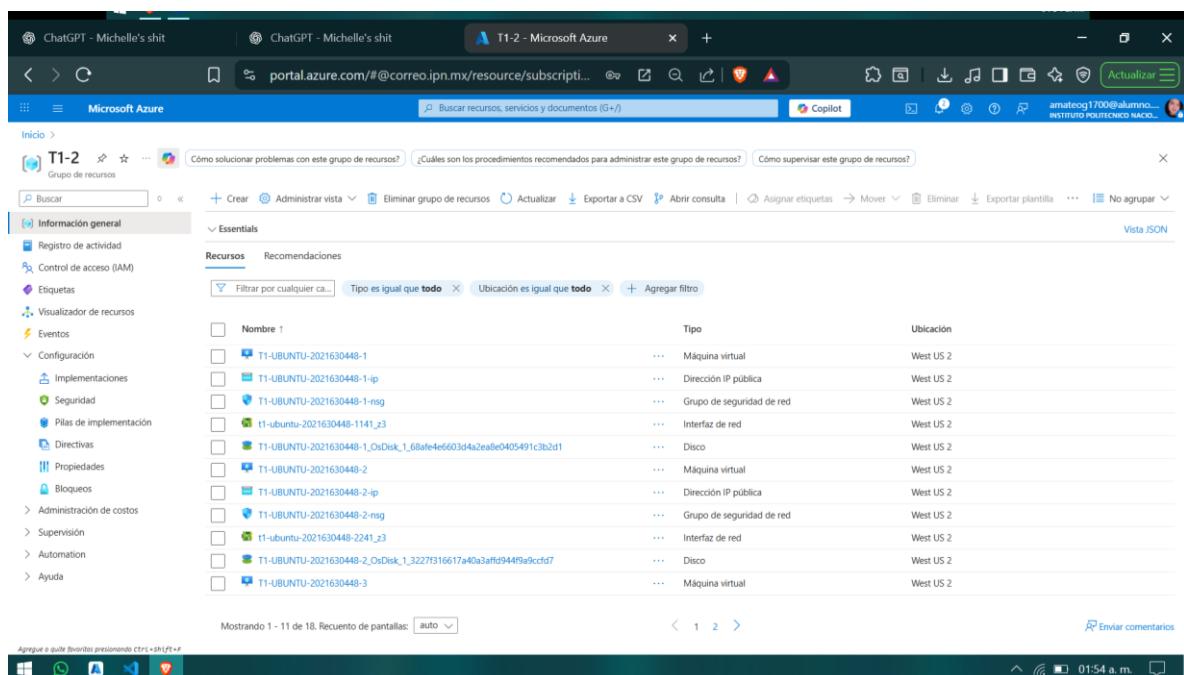
Agregar

Cancelar

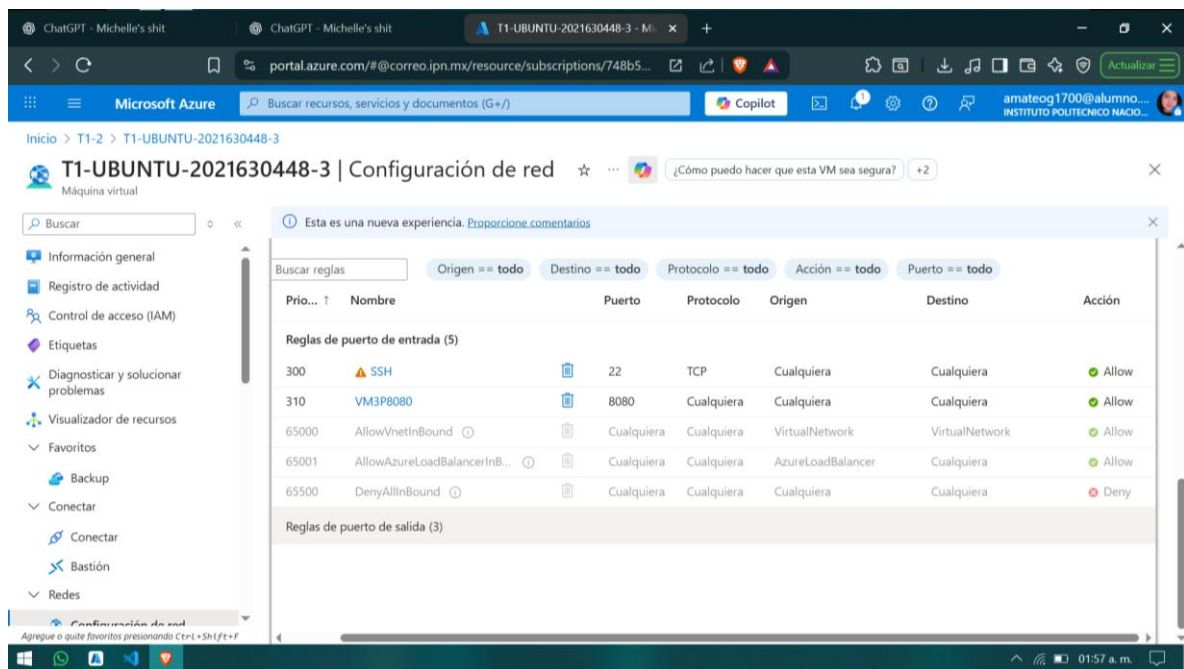
Por último, tenemos la Máquina virtual 3, igual siendo este el segundo servidor de la tarea, en este caso se hicieron 3 máquinas virtuales con las mismas características como se muestra en las imágenes.



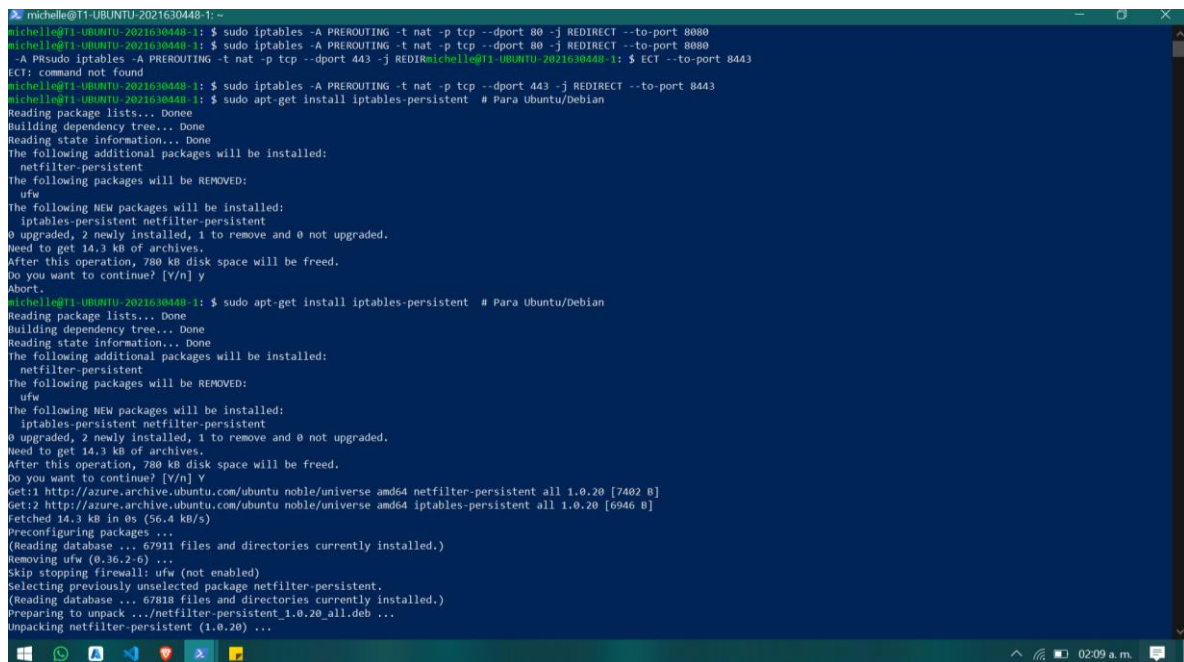
Por último, muestro las 3 VM creadas en el mismo grupo de recursos con sus respectivas interfaces de red, ips, NSG, entre otros más.



De igual forma se abrió el puerto 8080 en las configuración de red.



En la primera máquina virtual se mapearon los puertos 80 y 443 a los puertos 8080 y 8443 respectivamente como se muestra en la imagen. Esto redirige las peticiones que llegan a los primeros puertos que los segundos puertos sean los que tengan el tráfico.



En la máquina virtual 1 se ejecutó el administrador de tráfico ejecutando el puerto 8080 adicionando las maquinas virtuales 2 y 3 con sus Ips públicas con el puerto 8080, esto permitiendo las conexiones con los sockets, además de recibiendo las peticiones get.

```
michelle@T1-UBUNTU-2021630448-1: ~/SistemasDistribuidosPracticas/Practica 1
$ cd ~/SistemasDistribuidosPracticas/Practica 1
$ java AdministradorTráfico 8080 20.3.133.86 8080 20.64.249.99 8080
Proxy inverso escuchando en puerto 8080
Petición recibida:
GET /index.html HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: es-MX,es-419;q=0.9,es;q=0.8,en;q=0.7
If-Modified-Since: Sun, 21 Sep 2025 08:36:49 GMT

Petición recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
Accept: /*
Referer: http://20.57.134.21/index.html
Accept-Encoding: gzip, deflate
Accept-Language: es-MX,es-419;q=0.9,es;q=0.8,en;q=0.7

Petición recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
Accept: /*
Referer: http://20.57.134.21/index.html
Accept-Encoding: gzip, deflate
Accept-Language: es-MX,es-419;q=0.9,es;q=0.8,en;q=0.7

Petición recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
Accept: /*
Referer: http://20.57.134.21/index.html
```

```
michelle@T1-UBUNTU-2021630448-1: ~/SistemasDistribuidosPracticas/Practica 1
Accept-Language: es-US,es;q=0.7
Accept-Encoding: gzip, deflate

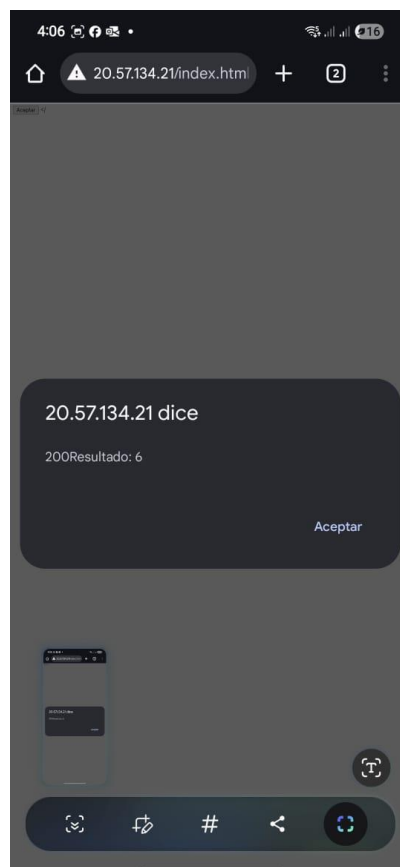
Petición recibida:
GET /favicon.ico HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-GPC: 1
Accept-Language: es-US,es;q=0.7
Referer: http://20.57.134.21/index.html
Accept-Encoding: gzip, deflate

Petición recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: /*
Sec-GPC: 1
Accept-Language: es-US,es;q=0.7
Referer: http://20.57.134.21/index.html
Accept-Encoding: gzip, deflate
```

Tenemos ahora los 2 servidores ejecutándose como se muestra en la maquina virtual 2 y 3, este servidor fue modificado con ChatGPT, esta muestra las peticiones get.

```
michelle@T1-UBUNTU-2021630448-2: ~/SistemasDistribuidosPracticas/Practica 1
michelle@T1-UBUNTU-2021630448-2: $ javac ServidorHTTP.java
michelle@T1-UBUNTU-2021630448-2: $ sudo java ServidorHTTP
Servidor HTTP escuchando en puerto 8080...
Petición: GET /index.html HTTP/1.1
Petición: GET /suma?a=1&b=2&c=3 HTTP/1.1
Petición: GET /index.html HTTP/1.1
Petición: GET /favicon.ico HTTP/1.1
Petición: GET /suma?a=1&b=2&c=3 HTTP/1.1
```

Como se muestra a continuación, en el teléfono inteligente en este caso mi celular personal al darle en el botón aceptar y agregar con la IP publica en este caso es <http://20.57.134.21/index.html>

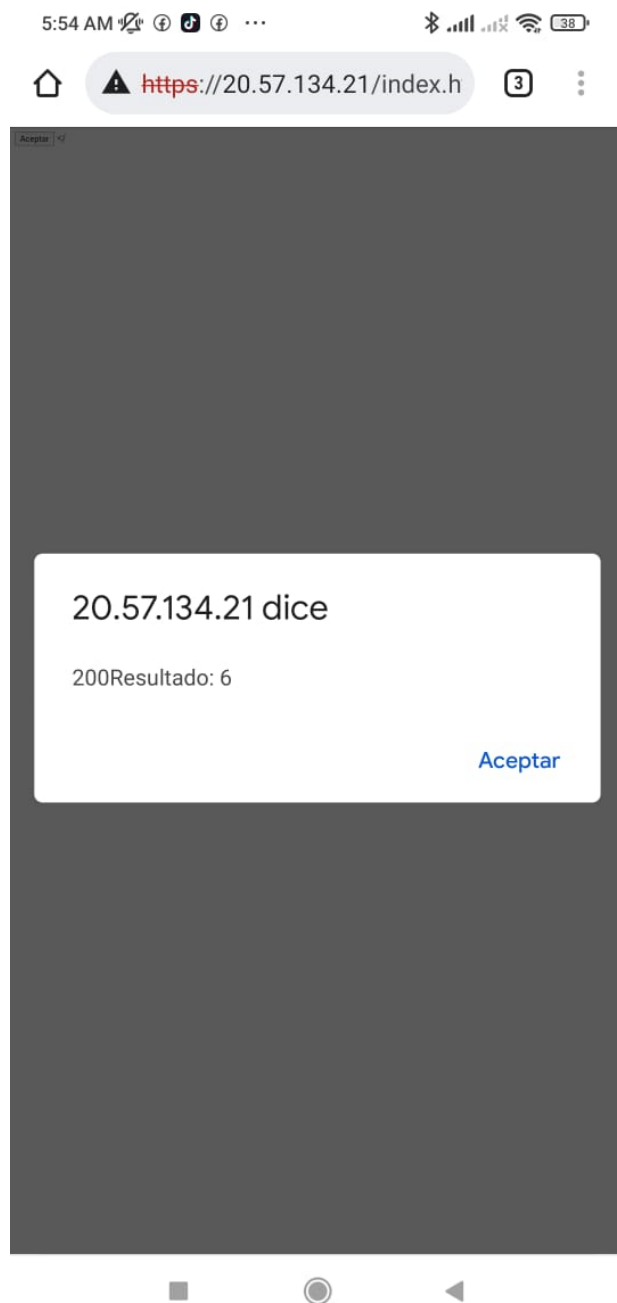


En el programa AdministradorTraficoSSL que se modificó con ChatGPT, primero se generaron los certificados auto firmados para poder realizar las peticiones HTTP, después de generarla, después se ejecuta como se muestra en las siguientes imágenes.

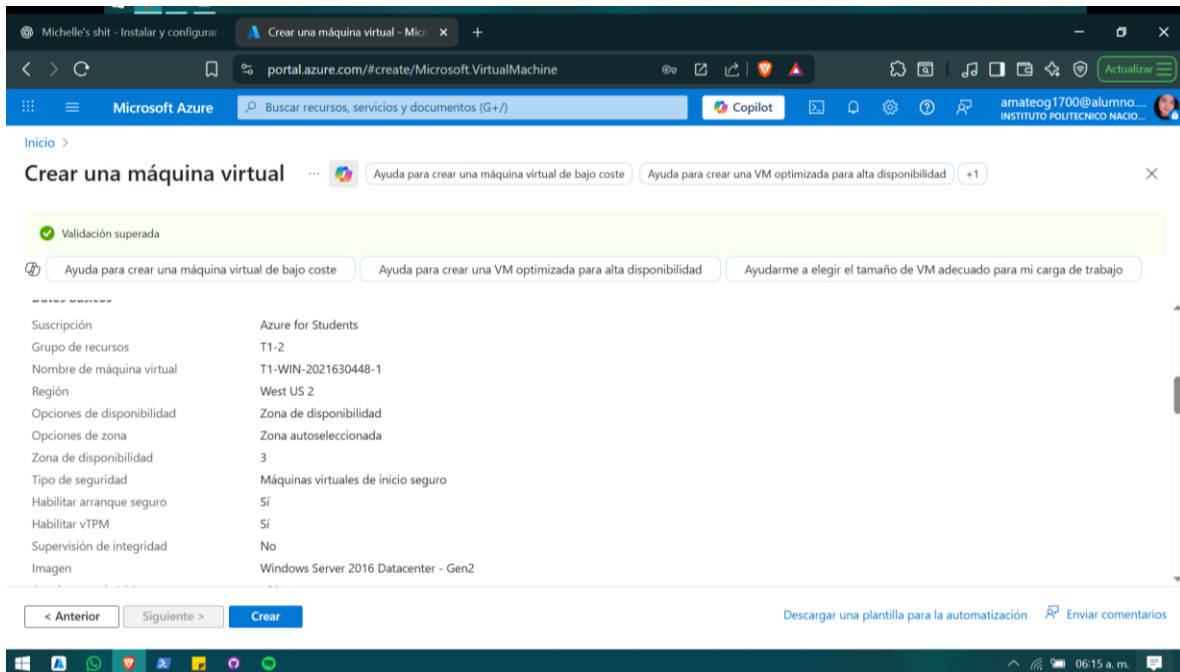
```
michelle@T1-UBUNTU-2021630448-1: ~/SistemasDistribuidosPractica1
michelle@T1-UBUNTU-2021630448-1:~$ keytool -genkeypair -keyalg RSA -alias certificado_servidor -keystore keystore_servidor.jks -storepass 1234567 -v
Validity 365 -keysize 2048 -dname "CN=localhost, OU=MiOrganizacion, O=MiEmpresa, L=CDMX, ST=CDMX, C=MX"
keytool error: java.lang.Exception: Key pair not generated, alias <certificado_servidor> already exists
michelle@T1-UBUNTU-2021630448-1:~$ ls
AdministradorTrafico.class  AdministradorTrafico.java  AdministradorTraficoSSL.java  ServidorHTTP.java  index.html  keystore_servidor.jks
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ cd ..
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ git pull origin main
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 2), reused 4 (delta 2), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 491 bytes | 245.00 KiB/s, done.
From https://github.com/michelloamg/SistemasDistribuidosPracticas
* branch            main                -> FETCH_HEAD
Updating 3f4c4ab..c161084
Fast-forward
 Practica1/AdministradorTraficoSSL.java | 2
 1 file changed, 1 insertion(+), 1 deletion(-)
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ cd Pract*
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ ls
AdministradorTrafico.class  AdministradorTrafico.java  AdministradorTraficoSSL.java  ServidorHTTP.java  index.html  keystore_servidor.jks
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ java AdministradorTraficoSSL.java
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1$ java AdministradorTraficoSSL 8080 20.3.133.86 8080 20.64.249.99 8080
Proxy Inverso SSL escuchando en puerto 8080
javax.net.ssl.SSLException: Unsupported or unrecognized SSL message
    at java.base/sun.security.ssl.SSLSocketImpl.handleUnknownRecord(SSLSocketImpl.java:457)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:175)
    at java.base/sun.security.ssl.SSLTransport.decode(SSLTransport.java:113)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1510)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1425)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:455)
    at java.base/sun.security.ssl.SSLSocketImpl.ensureNegotiated(SSLSocketImpl.java:925)
    at java.base/sun.security.ssl.SSLSocketImpl$AppInputStream.read(SSLSocketImpl.java:1016)
    at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:287)
    at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:330)
    at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:190)
    at java.base/java.io.InputStreamReader.read(InputStreamReader.java:177)
    at java.base/java.io.BufferedReader.fill(BufferedReader.java:162)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:329)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:396)
    at AdministradorTraficoSSL.manejadorConexion(AdministradorTraficoSSL.java:46)
    at AdministradorTraficoSSL.lambda$main$0(AdministradorTraficoSSL.java:30)
    at java.base/java.lang.Thread.run(Thread.java:840)
javax.net.ssl.SSLException: Unsupported or unrecognized SSL message
```

```
michelle@T1-UBUNTU-2021630448-1:~/SistemasDistribuidosPractica1
javax.net.ssl.SSLHandshakeException: Received fatal alert: certificate_unknown
    at java.base/sun.security.ssl.Alert.createSSLException(Alert.java:131)
    at java.base/sun.security.ssl.Alert.createSSLException(Alert.java:117)
    at java.base/sun.security.ssl.TransportContext.fatal(TransportContext.java:370)
    at java.base/sun.security.ssl.Alert$AlertConsumer.consume(Alert.java:293)
    at java.base/sun.security.ssl.TransportContext.dispatch(TransportContext.java:209)
    at java.base/sun.security.ssl.SSLTransport.decode(SSLTransport.java:172)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1510)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1425)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:455)
    at java.base/sun.security.ssl.SSLSocketImpl.ensureNegotiated(SSLSocketImpl.java:925)
    at java.base/sun.security.ssl.SSLSocketImpl$AppInputStream.read(SSLSocketImpl.java:1016)
    at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:287)
    at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:330)
    at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:190)
    at java.base/java.io.InputStreamReader.read(InputStreamReader.java:177)
    at java.base/java.io.BufferedReader.fill(BufferedReader.java:162)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:329)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:396)
    at AdministradorTraficoSSL.manejadorConexion(AdministradorTraficoSSL.java:46)
    at AdministradorTraficoSSL.lambda$main$0(AdministradorTraficoSSL.java:30)
    at java.base/java.lang.Thread.run(Thread.java:840)
Petición SSL recibida:
GET /suma?n=1&b=2&c=3 HTTP/1.1
Host: 20.57.134.21
Connection: keep-alive
sec-ch-ua: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?1
User-Agent: Mozilla/5.0 (Linux; Android 11; M2004J19C) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.85 Mobile Safari/537.36
sec-ch-ua-platform: "Android"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://20.57.134.21/index.html
Accept-encoding: gzip, deflate, br
Accept-Language: es-US;q=0.9,es-419;q=0.8,en;q=0.7,ny;q=0.6
```

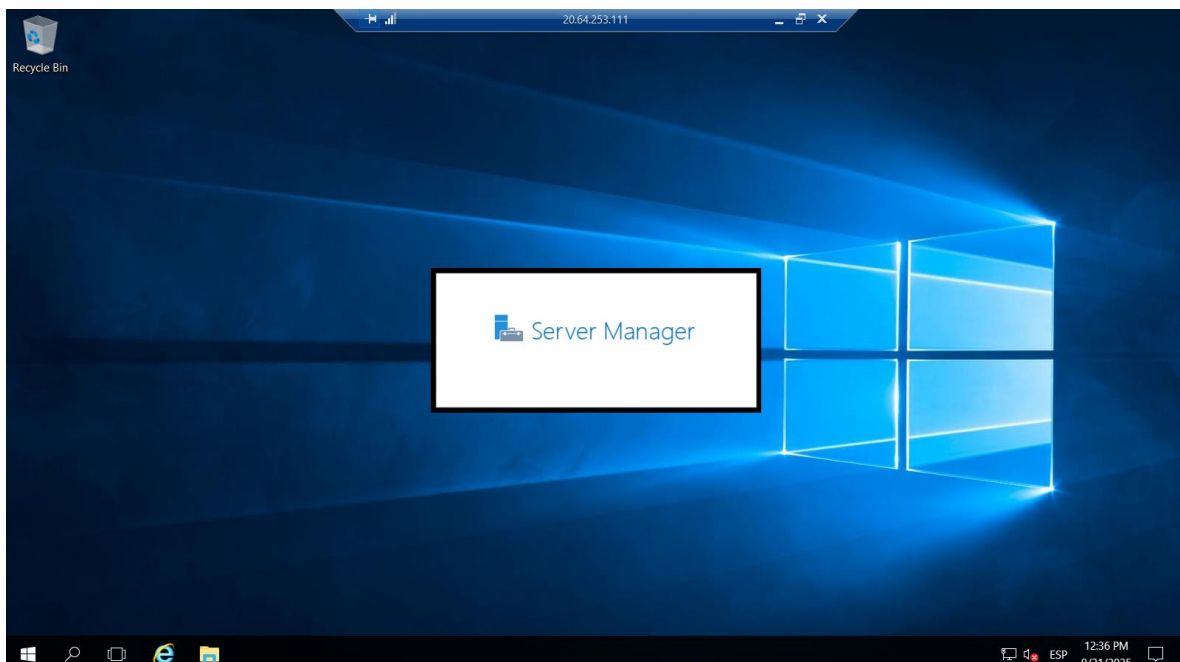
En este caso se hizo la prueba en otro celular, con el protocolo HTTPS sale con advertencia, sin embargo, al entrar y dar en aceptar, nos da el resultado 6.



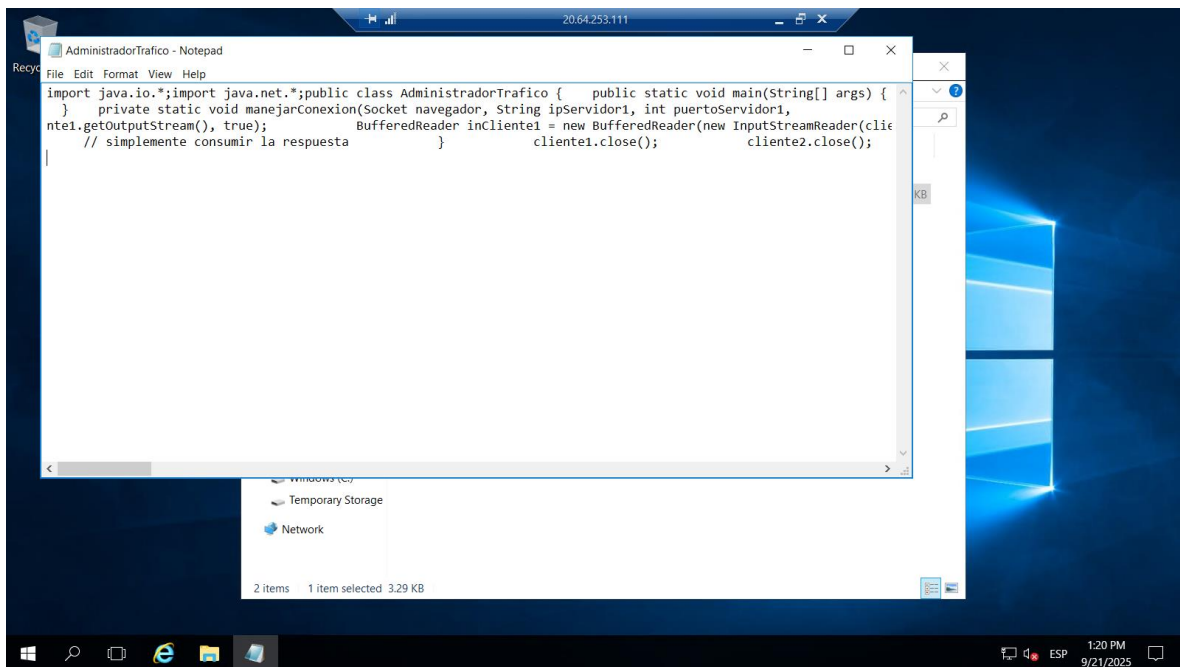
En este caso se eliminó la maquina virtual 1 de Ubuntu server y ahora se hace una máquina virtual con Windows server, con data center de la generación 2, con un disco normal HDD.



Por medio del protocolo RDP y montando el disco local F que generamos desde local para que pudiéramos agregar los archivos necesarios, en este caso ingresamos con el archivo que bajamos de la nube, después ingresamos por medio de contraseña e ingresamos.



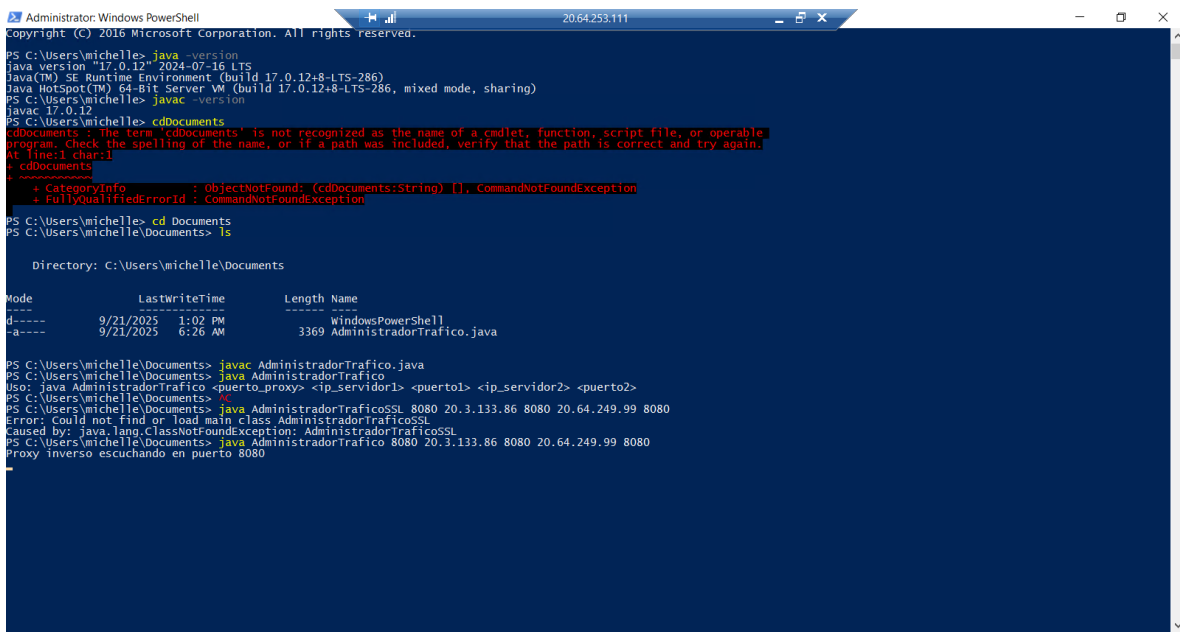
Para las siguientes imágenes se muestra la ejecución del administrador de trafico, este fue creado por la IA. Aquí se muestra que se hizo la ejecución.



The screenshot shows a Windows desktop with a blue background. A Notepad window titled "AdministradorTrafico - Notepad" is open, displaying the following Java code:

```
import java.io.*;import java.net.*;public class AdministradorTrafico {    public static void main(String[] args) {        }        private static void manejarConexion(Socket navegador, String ipServidor1, int puertoServidor1,        nte1.getOutputStream(), true);        BufferedReader inCliente1 = new BufferedReader(new InputStreamReader(clie        // simplemente consumir la respuesta        }        cliente1.close();        cliente2.close();    }    }
```

Below the Notepad window, a File Explorer window is open, showing the "Temporary Storage" and "Network" locations. The status bar at the bottom indicates "2 items 1 item selected 3.29 KB".



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell" with the following commands and output:

```
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\michelle> java -version
java version "17.0.12" 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 17.0.12+8-LTS-286)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.12+8-LTS-286, mixed mode, sharing)
PS C:\Users\michelle> javac -version
javac 17.0.12
PS C:\Users\michelle> cd Documents
cdDocuments : the term 'cdDocuments' is not recognized as the name of a cmdlet, function, script file, or operable
program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
~ cdDocuments ~
~ ~
+ CategoryInfo          : ObjectNotFound: (cdDocuments:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\michelle> cd Documents
PS C:\Users\michelle\Documents> ls

Directory: C:\Users\michelle\Documents

Mode                LastWriteTime         Length Name
----                -
d-----          9/21/2025   1:02 PM                WindowsPowerShell
-a-----          9/21/2025   6:26 AM             3369 AdministradorTrafico.java

PS C:\Users\michelle\Documents> javac AdministradorTrafico.java
PS C:\Users\michelle\Documents> java AdministradorTrafico
Usage: java AdministradorTrafico <puerto_proxy> <ip_servidor1> <puerto1> <ip_servidor2> <puerto2>
PS C:\Users\michelle\Documents> java AdministradorTrafico 8080 20.3.133.86 8080 20.64.249.99 8080
Error: Could not find or load main class AdministradorTraficoSSL
Caused by: java.lang.ClassNotFoundException: AdministradorTraficoSSL
PS C:\Users\michelle\Documents> java AdministradorTrafico 8080 20.3.133.86 8080 20.64.249.99 8080
proxy inverso escuchando en puerto 8080
```

En la siguiente imagen se muestra que se da el resultado al apretar el botón aceptar, esto, solamente la diferencia que tuve que poner el puerto ya que no dejaba ingresar ya que bloqueaba la conexión.



En la siguiente imagen se ejecuta el administrador de tráfico SSL, primero se generó el certificado auto firmado y después se ejecutó.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\michelle> cd documents
PS C:\Users\michelle\documents> java AdministradorTráficoSSL 8443 20.3.133.86 8080 20.64.249.99 8080
Proxy inverso SSL escuchando en puerto 8443
javax.net.ssl.SSLHandshakeException: Remote host terminated the handshake
    at java.base/sun.security.ssl.SSLSocketImpl.handleEOF(SSLSocketImpl.java:1715)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1514)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1421)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:455)
    at java.base/sun.security.ssl.SSLSocketImpl.ensureNegotiated(SSLSocketImpl.java:921)
    at java.base/sun.security.ssl.SSLSocketImpl$AppInputStream.read(SSLSocketImpl.java:1012)
    at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:270)
    at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:313)
    at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:188)
    at java.base/java.io.InputStreamReader.read(InputStreamReader.java:177)
    at java.base/java.io.BufferedReader.fill(BufferedReader.java:162)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:329)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:396)
    at AdministradorTráficoSSL.manejarConexion(AdministradorTráficoSSL.java:46)
    at AdministradorTráficoSSL.lambda$main$0(AdministradorTráficoSSL.java:30)
    at java.base/java.lang.Thread.run(Thread.java:842)
Caused by: java.io.EOFException: SSL peer shut down incorrectly
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.read(SSLSocketImplInputRecord.java:489)
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.readHeader(SSLSocketImplInputRecord.java:478)
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.decode(SSLSocketImplInputRecord.java:160)
    at java.base/sun.security.ssl.SSLTransport.decode(SSLTransport.java:111)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1506)
    at java.base/sun.security.ssl.SSLSocketImpl.handleEOF(SSLSocketImpl.java:1715)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1514)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1421)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:455)
    at java.base/sun.security.ssl.SSLSocketImpl.ensureNegotiated(SSLSocketImpl.java:921)
    at java.base/sun.security.ssl.SSLSocketImpl$AppInputStream.read(SSLSocketImpl.java:1012)
    at java.base/sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:270)
    at java.base/sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:313)
    at java.base/sun.nio.cs.StreamDecoder.read(StreamDecoder.java:188)
    at java.base/java.io.InputStreamReader.read(InputStreamReader.java:177)
    at java.base/java.io.BufferedReader.fill(BufferedReader.java:162)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:329)
    at java.base/java.io.BufferedReader.readLine(BufferedReader.java:396)
    at AdministradorTráficoSSL.manejarConexion(AdministradorTráficoSSL.java:46)
    at AdministradorTráficoSSL.lambda$main$0(AdministradorTráficoSSL.java:30)
    at java.base/java.lang.Thread.run(Thread.java:842)
Caused by: java.io.EOFException: SSL peer shut down incorrectly
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.read(SSLSocketImplInputRecord.java:489)
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.readHeader(SSLSocketImplInputRecord.java:478)
    at java.base/sun.security.ssl.SSLSocketImplInputRecord.decode(SSLSocketImplInputRecord.java:160)
    at java.base/sun.security.ssl.SSLTransport.decode(SSLTransport.java:111)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1506)
```



```

at AdminStrador.HaricOSS.LambdaMain10.AdminStrador.HaricOSS.java:30)
at java.base/java.lang.Thread.run(Thread.java:842)

Petici??n SSL recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.64.253.111:8443
Connection: keep-alive
sec-ch-ua-platform: "Android"
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
sec-ch-ua: "Chromium";v="140", "Not=A?Brand";v="24", "Brave";v="140"
sec-ch-ua-mobile: ?1
Accept: */*
Sec-GPC: 1
Accept-Language: es-US,es;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://20.64.253.111:8443/index.html
Accept-Encoding: gzip, deflate, br, zstd

Petici??n SSL recibida:
GET /suma?a=1&b=2&c=3 HTTP/1.1
Host: 20.64.253.111:8443
Connection: keep-alive
sec-ch-ua-platform: "Android"
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0 Mobile Safari/537.36
sec-ch-ua: "Chromium";v="140", "Not=A?Brand";v="24", "Brave";v="140"
sec-ch-ua-mobile: ?1
Accept: */*
Sec-GPC: 1
Accept-Language: es-US,es;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://20.64.253.111:8443/index.html
Accept-Encoding: gzip, deflate, br, zstd

```

Para la última captura, se ingresó de manera normal con el HTTPS, de igual manera nos decía que la conexión era insegura, sin embargo, pude ingresar, de igual manera tuve que poner el puerto por las mismas razones de que la conexión me lo rechazaba.



Al final se eliminaron el grupo de recursos.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a search bar, and a user profile. The main content area is divided into two sections: 'Servicios de Azure' (Azure Services) and 'Recursos' (Resources).

Servicios de Azure

- Crear un recurso
- Máquinas virtuales
- Grupos de recursos
- Suscripciones
- Cuentas de almacenamie...
- Aplicación de funciones
- Cost Manage...
- Almacenes de Backup
- Supervisar
- Más servicios

Recursos

Reciente Favorito

Nombre	Tipo	Última consulta
T1-WIN-2021630448-1	Máquina virtual	hace 58 minutos
T1-UBUNTU-2021630448-3	Máquina virtual	hace 1 hora
T1-UBUNTU-2021630448-2	Máquina virtual	hace 1 hora
TTII-MIDUELO	Grupo de recursos	hace 2 horas
NetworkWatcher_westus2	Network Watcher	hace 3 horas

Notificaciones

- Más eventos en el registro de actividad --Descartar todo
- Se eliminó el grupo de recursos T1-2. Se eliminó el grupo de recursos T1-2. hace 3 minutos
- Máquina virtual detenida correctamente. La máquina virtual 'T1-WIN-2021630448-1' se detuvo correctamente. hace 5 minutos
- Se actualizó la regla de seguridad. La regla de seguridad 'efarr' se guardó correctamente. hace 8 minutos

Conclusiones

El proyecto culminó exitosamente con la implementación y verificación de la arquitectura de proxy inverso en la nube. La práctica demostró la operatividad de los principios de los sistemas distribuidos a través de una aplicación funcional de balanceo de carga rudimentario y terminación SSL.

Técnicamente, se logró la configuración de la infraestructura como servicio (IaaS) en Azure, asegurando la conectividad mediante la apertura y redirección de puertos (80 y 443 a 8080 y 8443) a nivel del sistema operativo con iptables. El desarrollo de la lógica del proxy inverso en Java (`AdministradorTráfico.java`) ilustró cómo se manejan y reenvían sockets para lograr la mediación de las peticiones, destacando la gestión de la solicitud a dos servidores y el reenvío selectivo de la respuesta del Servidor-1 al cliente.

La implementación de la variante HTTPS del proxy probó la capacidad de manejar peticiones seguras mediante el uso de keys y certificados autofirmados, aunque en un entorno de prueba, lo que generó la advertencia de conexión insegura en el cliente. Finalmente, la integración del manejo de caché HTTP (Last-Modified/If-Modified-Since) en los servidores backend validó el control sobre la eficiencia en la transferencia de datos y la reducción de la carga del servidor.