Universidad de Guadalajara

Centro Universitario de Ciencias Exactas E Ingenierías (CUCEI)

Sistema de educación superior

Actividad 06 - Optimización de Transferencia de Archivos en una VPN con Algoritmos Voraces

Nombre del profesor:

JORGE ERNESTO LOPEZ ARCE DELGADO

Nombre de los alumnos y roles:

Jimenez Magaña Sergio	- Project Manager
Lopez Torres Raul Eduardo	- Kruskal
Renteria Xochipa Moises Alejandro	- Configuración de VPN y mediciones
Vázquez Mendoza David Fernando	- Dijkstra

Fecha
11 de Mayo del 2025

Parte 1. Configuración de la VPN

El asesor recomendó trabajar con dos VPNs diferentes: WireGuard y OpenVPN. La primera opción, WireGuard, es una VPN muy robusta pero con poca documentación práctica y, por lo tanto, difícil de configurar. OpenVPN es una VPN de pago si es que necesitas conectar más de dos dispositivos a la red; para este trabajo, se precisaba una VPN que permitiera la conexión entre, por lo menos, 4 dispositivos. Tras buscar otras opciones, el equipo dio con la VPN de Hamachi.
Se decidió trabajar con Hamachi por comodidad y facilidad en el setup de la VPN. El dispositivo de Rentería Moisés sirve como servidor y los demás dispositivos fungen el papel de clientes dentro de la red.
Hamachi establece IPs por defecto a cada cliente de la VPN. Las direcciones IPv4 de cada dispositivo se muestran en la **Tabla 1.1**.

Nombre dispositivo	**IPv4**
Rentería Moisés	25.54.198.46
Vazquez David	25.58.36.130
Lopez Raúl	25.4.219.19
Jimenez Sergio	25.59.171.8

Tabla 1.1 Direcciones IPv4 de cada dispositivo.

En la **Imagen 1.1** se muestra una captura de pantalla de la IGU de Hamachi, donde se muestran los 4 dispositivos conectados a la VPN.

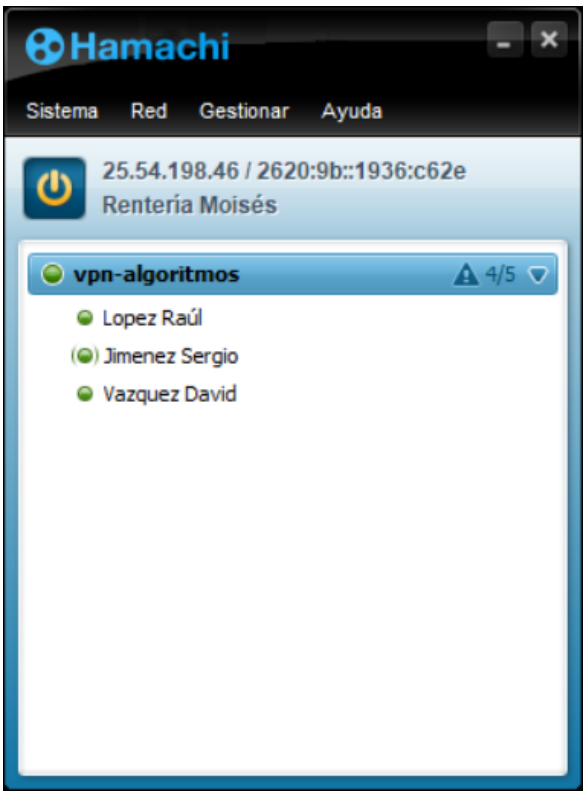


Imagen 1.1 IGU de Hamachi con 4/5 dispositivos conectados.

Parte 2. Medición de Métricas de Red

Se realizó un script en Python utilizando ping para medir la latencia entre los clientes de la VPN. Cada cliente ejecutó el script desde su dispositivo, haciendo ping a cada uno de los demás clientes de la VPN. Los resultados de la prueba de medición de latencia se muestran en la **Tabla 2.1**. Se adjuntan, además, tres capturas de pantalla de la consola de Jimenez Sergio haciendo ping tanto a Vazquez David (**Imagen 2.1**) como a Rentería Moisés (**Imagen 2.2**), así como una captura de Jimenez Sergio midiendo el ancho de banda con *iperf3* desde su dispositivo hasta el de Rentería Moisés (**Imagen 2.3**).

Recibe / Envía	**Rentería Moisés** (25.54.198.46)	**Vazquez David** (25.58.36.130)	**Lopez Raul** (25.4.219.19)	**Jimenez Sergio** (25.59.171.8)
Rentería Moisés (25.54.198.46)	-	30.2 ms	29.2 ms	30 ms
Vazquez David (25.58.36.130)	36.6 ms	-	14.1 ms	102 ms
Lopez Raul (25.4.219.19)	33 ms	10 ms	-	255 ms
Jimenez Sergio (25.59.171.8)	31 ms	116 ms	270 ms	-

Tabla 2.1 Resultados de la prueba de medición de latencia.

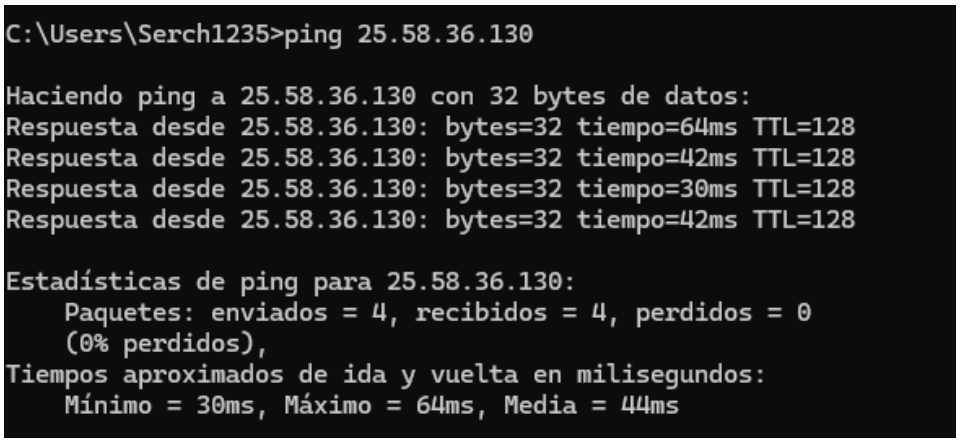


Imagen 2.1 Ping desde Sergio hacía David en el CMD.

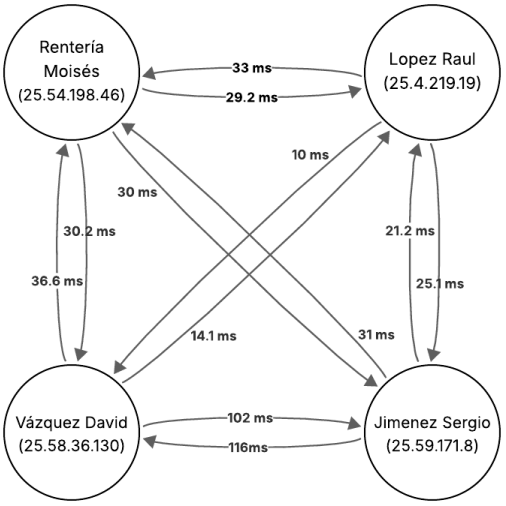
```
C:\Users\Serch1235>ping 25.54.198.46

Haciendo ping a 25.54.198.46 con 32 bytes de datos:
Respuesta desde 25.54.198.46: bytes=32 tiempo=122ms TTL=128
Respuesta desde 25.54.198.46: bytes=32 tiempo=86ms TTL=128
Respuesta desde 25.54.198.46: bytes=32 tiempo=86ms TTL=128
Respuesta desde 25.54.198.46: bytes=32 tiempo=66ms TTL=128

Estadísticas de ping para 25.54.198.46:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 66ms, Máximo = 122ms, Media = 90ms
```

Imagen 2.2 Ping desde Sergio hacía Moisés en el CMD.

```
-----------------------------------------------------------
Accepted connection from 25.54.198.46, port 30090
[  5] local 25.59.171.8 port 5201 connected to 25.54.198.46 port 30091
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.02   sec   238 KBytes  1.92 Mbits/sec
[  5]   1.02-2.00   sec   178 KBytes  1.48 Mbits/sec
[  5]   2.00-3.00   sec   306 KBytes  2.51 Mbits/sec
[  5]   3.00-4.00   sec   331 KBytes  2.71 Mbits/sec
[  5]   4.00-5.01   sec   169 KBytes  1.38 Mbits/sec
[  5]   5.01-6.00   sec   297 KBytes  2.44 Mbits/sec
[  5]   6.00-7.00   sec   190 KBytes  1.56 Mbits/sec
[  5]   7.00-8.00   sec   308 KBytes  2.53 Mbits/sec
[  5]   8.00-9.00   sec   309 KBytes  2.53 Mbits/sec
[  5]   9.00-10.01  sec   193 KBytes  1.58 Mbits/sec
[  5]  10.01-10.05  sec  24.0 KBytes  4.83 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.05  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.05  sec  2.49 MBytes  2.08 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Imagen 2.3 *iperf3* desde Sergio hasta Moisés.

A partir de los resultados anteriores, se dibujó un grafo representativo de la VPN, donde el peso de cada arista es la latencia obtenida de la ejecución del ping desde un nodo a otro nodo (cliente a cliente). El **Grafo 2.1** presenta dichos resultados.



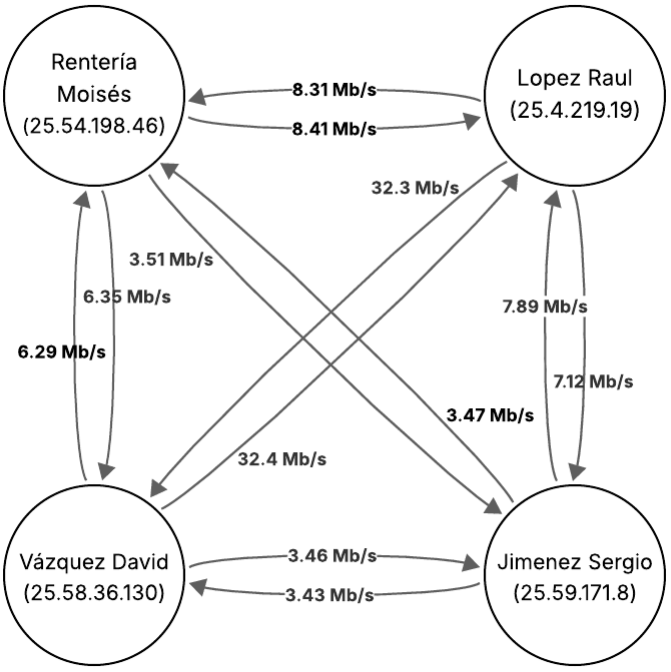
Grafo 2.1 Gráfico representativo de la prueba de latencia en la VPN.

También se ejecutó una prueba para medir el ancho de banda dentro de la VPN utilizando iperf3 desde la consola de Windows. Los resultados de la prueba se exponen en la **Tabla 2.2**.

Recibe / Envía	**Rentería Moisés** (25.54.198.46)	**Vazquez David** (25.58.36.130)	**Lopez Raul** (25.4.219.19)	**Jimenez Sergio** (25.59.171.8)
Rentería Moisés (25.54.198.46)	-	6.35 Mbits/sec	8.41 Mbits/sec	3.51 Mbits/sec
Vazquez David (25.58.36.130)	6.29 Mbits/sec	-	32.4 Mbits/sec	3.46 Mbits/sec
Lopez Raul (25.4.219.19)	8.31 Mbits/sec	32.3 Mbits/sec	-	7.12 Mbits/sec
Jimenez Sergio (25.59.171.8)	3.47 Mbits/sec	3.43 Mbits/sec	7.89 Mbits/sec	-

Tabla 2.2 Resultados de la prueba de medición de ancho de banda.

Al igual que con la latencia, se dibujó un grafo de la VPN con los pesos de las aristas siendo el ancho de banda de las conexiones entre los nodos. El dibujo se puede observar en el **Grafo 2.2**.



Grafo 2.2 Gráfico representativo de la prueba de ancho de banda en la VPN.

Por último, con las mediciones, se generó un gráfico que nos permite observar la relación entre la latencia y el ancho de banda de cada una de las conexiones. El **Diagrama 2.1** es el que lo ilustra.

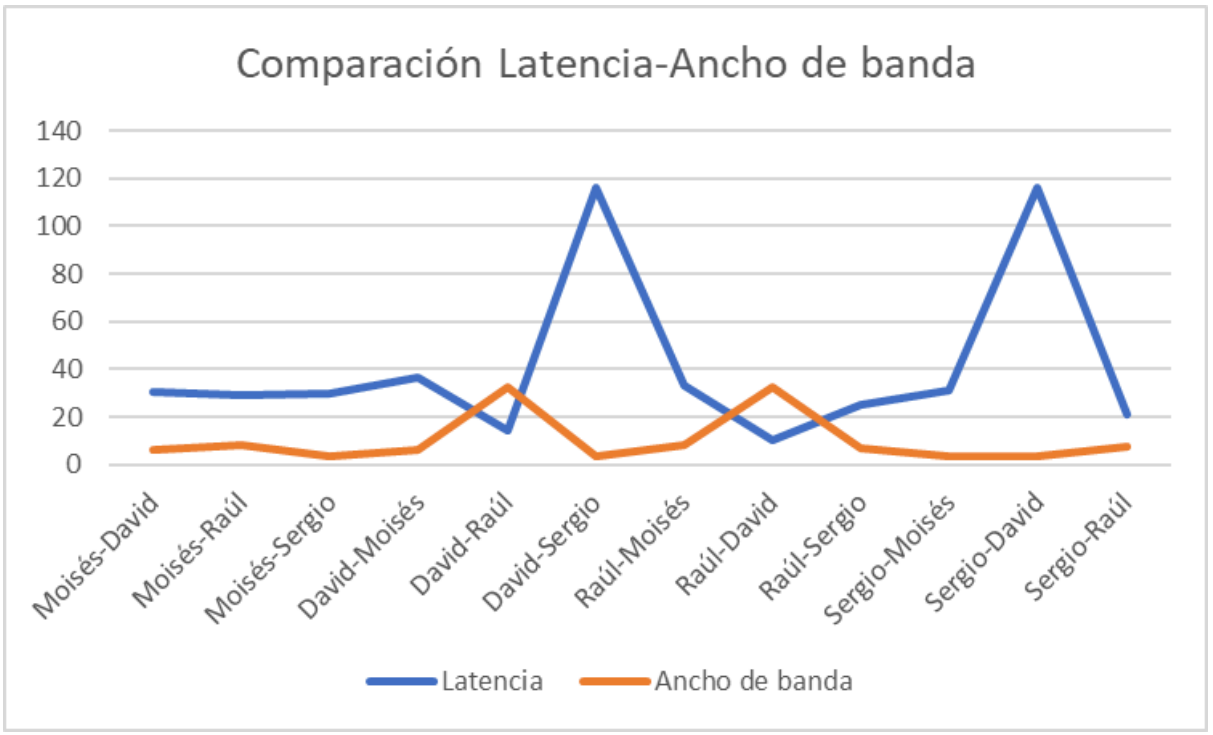


Diagrama 1.1 Comparación Latencia-Ancho de Banda

Algo que salta rápidamente a la vista son los puntos críticos superiores de la latencia en las conexiones David-Sergio y Sergio-David. Se puede observar que en dichos puntos, el ancho de banda encuentra sus puntos críticos inferiores.
Por otro lado, los puntos críticos inferiores de la latencia se encuentran en las conexiones David-Raúl y Raúl-David. En dichos puntos, esta vez, se observan también los puntos críticos superiores del ancho de banda.
El análisis del gráfico explica una relación donde a mayor ancho de banda, menor latencia y, a menor ancho de banda, mayor latencia.

Parte 3. Implementación de Dijkstra

Con apoyo del grafo resultante de la medición de latencia se obtuvo la idea general para implementar un script en Python que pudiera aplicar el algoritmo de Dijkstra en el grafo ponderado con la latencia de conexiones de la VPN.
En realidad, se generaron dos scripts:

1. Enviar: el script para enviar genera una IGU utilizando *tkinter*. Dicha interfaz permite escoger un nodo emisor y un nodo receptor, así como incluye un selector de archivos para poder seleccionar un archivo a enviar. La IGU se puede observar en la **Imagen 3.1**. El uso de este script es muy sencillo; simplemente se escoge como emisor el nodo desde el cual se accedió a la IGU y como emisor cualquier nodo que esté ejecutando el segundo script. Se selecciona cualquier archivo dentro del dispositivo y se presiona enviar. Si el archivo fue recibido con éxito, la IGU mostrará un mensaje como en la **Imagen 3.2**.

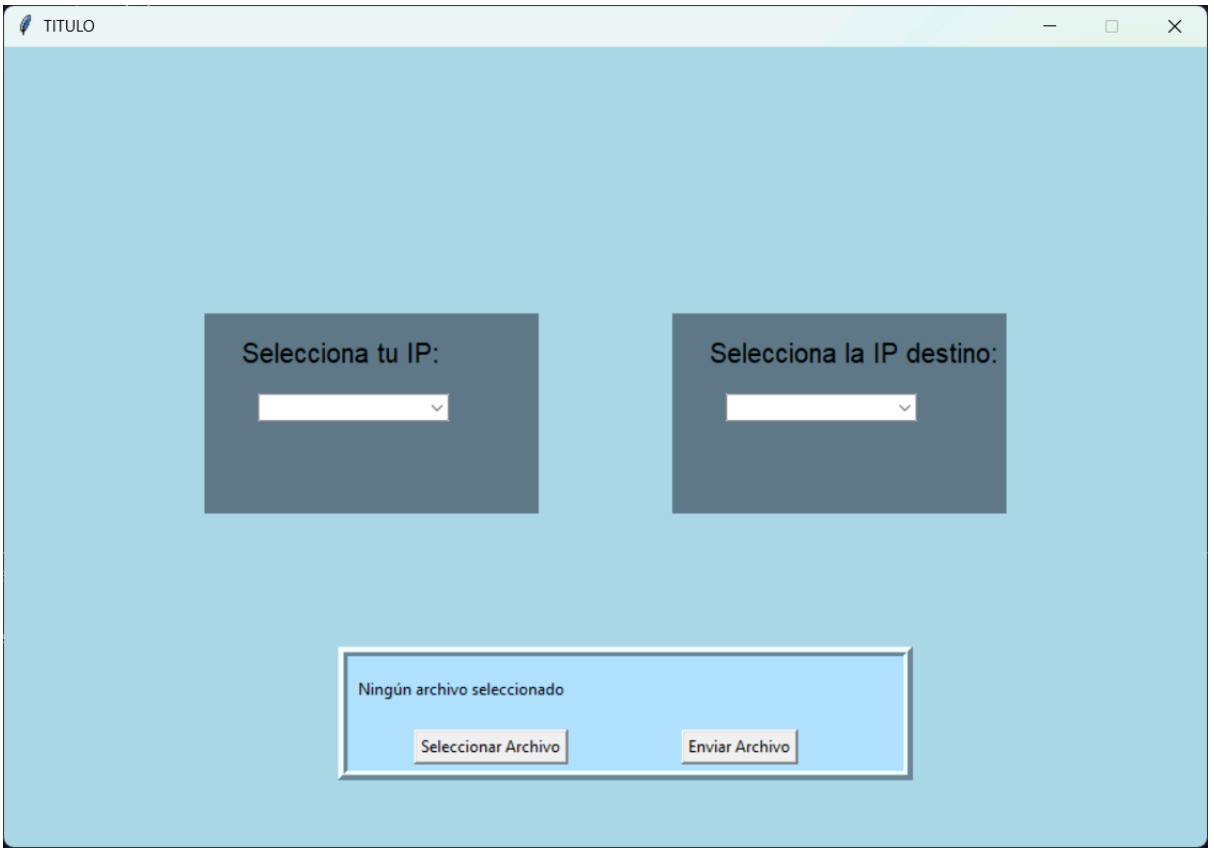


Imagen 3.1 IGU del script para enviar archivos.

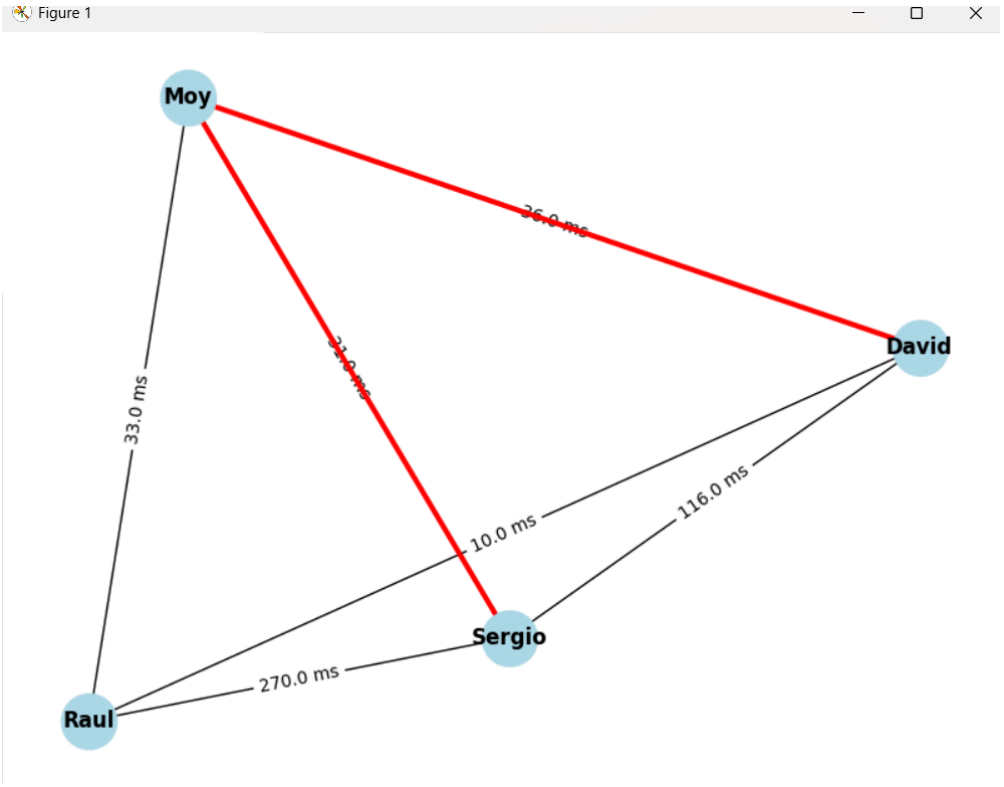


Imagen 3.2 Ejemplo de implementación de Dijkstra.

Imagen 3.3 Mensaje de transferencia de archivo exitosa.

2. Recibir: este script se encarga de establecer al nodo que lo ejecute en un estado de escucha, esperando a que otro nodo le envíe archivos desde el script de enviar y de igual forma seguir con la ruta implementada por el algoritmo. Si la conexión es exitosa y el archivo fue recibido, se observará un mensaje en la consola como el de la **Figura 3.3**.

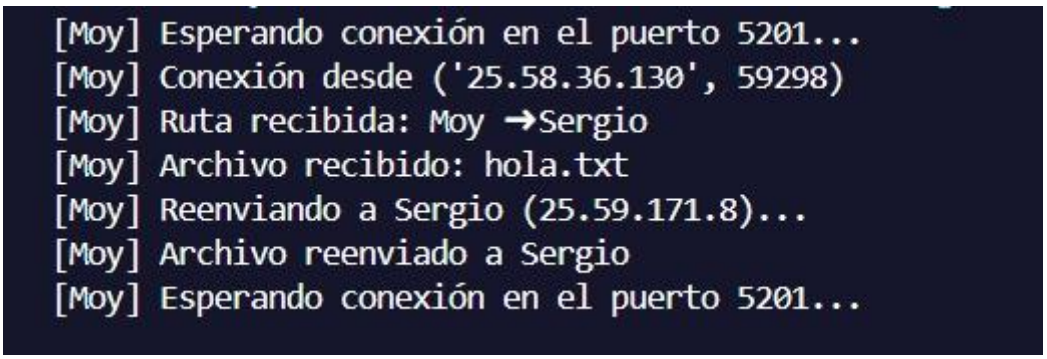


Imagen 3.4 Consola de Windows con una transferencia de archivo exitosa

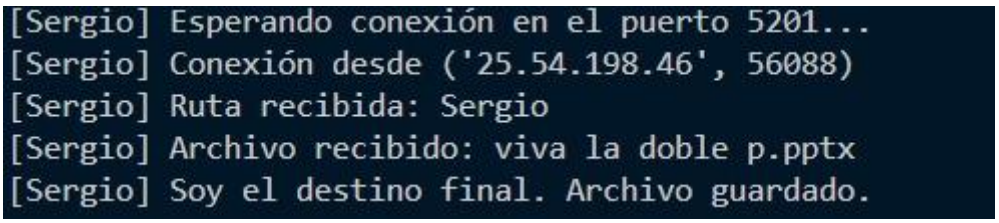


Imagen 3.5 Consola de Windows con una transferencia de archivo exitosa

Existe un pequeño bug: si el algoritmo establece una ruta del tipo X→Y→Z dónde X es el remitente, Z el receptor y Y son todos los nodos intermediarios entre X y Z, el archivo será compartido con todos los nodos Y y todos tendrán acceso a dicho archivo. Es un problema importante que, en este caso, no afecta en absoluto al propósito del proyecto, pero que, sin embargo, debe ser arreglado en el futuro.

Parte 4. Implementación de Kruskal

Utilizando el grafo ponderado obtenido de la medición del ancho de banda en la VPN (**Grafo 2.2**), se implementó un script de Python para aplicar el algoritmo de Kruskal al grafo y así obtener un árbol de expansión mínima.
El script construye el grafo con los dispositivos de la VPN como nodos y conexiones entre los mismos como aristas. La ponderación de las aristas es la medida del ancho de banda entre cada dispositivo.
Se aplica Kruskal sobre el grafo y se obtiene el MST para, posteriormente, exponer el resultado en un gráfico utilizando *networkx* para el mapeo del grafo y *matplotlib* para la visualización del grafo.
En la **Imagen 4.1** se observa el resultado obtenido al correr el script.

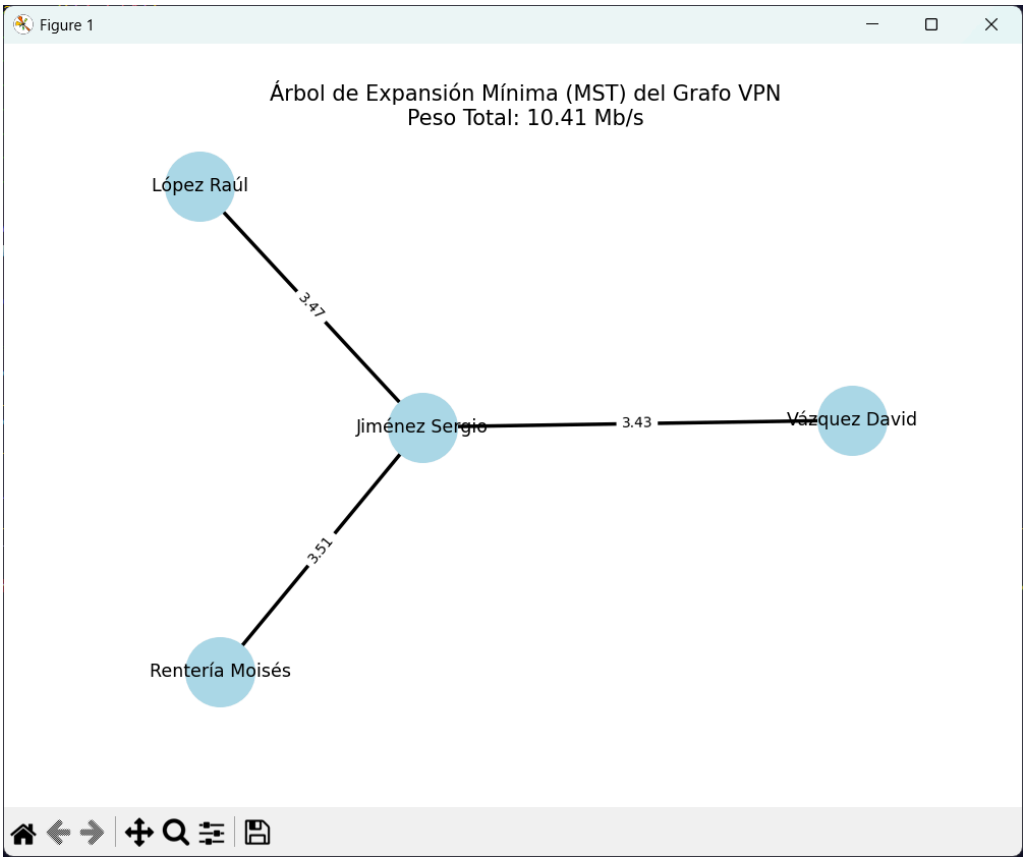


Imagen 4.1 Gráfico del MST de la VPN.

Conclusiones

La realización de este proyecto representó una oportunidad significativa para que el equipo se adentrará en un entorno de redes y conectividad mediante el uso de una VPN, un ámbito en el que los conocimientos previos eran mínimos o inexistentes. Este proceso implicó un aprendizaje colectivo que abarcó desde los fundamentos técnicos hasta la aplicación práctica de conceptos complejos.
El primer desafío fue la configuración de la VPN, una etapa que resultó particularmente exigente debido a la falta de experiencia del equipo en este campo. La elección de la VPN adecuada se convirtió en un obstáculo inicial, ya que requería evaluar opciones que se ajustaran a los objetivos del proyecto. Finalmente, Hamachi se destacó como la solución más accesible y funcional, permitiendo al equipo establecer una red virtual que cumplía con los requerimientos específicos del proyecto de manera eficiente.
Una vez superada la configuración de la VPN y con la red operativa, el equipo se enfocó en comprender conceptos clave como la latencia y el ancho de banda. Estos términos, aunque fundamentales en el ámbito de las redes, representaron un reto inicial debido a su naturaleza técnica. Sin embargo, tras un esfuerzo colectivo, el equipo logró no solo entenderlos, sino también medirlos de manera práctica, sentando las bases para la siguiente fase del proyecto: la creación de grafos.
La elaboración de los grafos resultó ser un proceso más familiar para el equipo, dado que los integrantes ya contaban con conocimientos previos en esta área. No obstante, la asignación de ponderaciones a las aristas introdujo cierta complejidad. La necesidad de considerar dos pesos distintos para cada arista —uno correspondiente a la latencia y otro al ancho de banda— generó confusión inicial, ya que implicaba manejar múltiples variables simultáneamente en la representación gráfica de la red.
Finalmente, la aplicación de algoritmos sobre los grafos marcó otra etapa de aprendizaje intensivo. El equipo tuvo que dedicar tiempo y esfuerzo a profundizar en la comprensión de los conceptos subyacentes a estos algoritmos, asegurándose de aplicarlos correctamente para obtener resultados válidos. Este proceso reflejó la capacidad del equipo para superar las barreras iniciales de conocimiento y adaptarse a los desafíos técnicos del proyecto, consolidando un aprendizaje integral en el manejo de redes, grafos y algoritmos.