

Tarea Individual - 2° Cuatrimestre 2025

## “Análisis estadístico de los tiempos de RTT (Round Trip Time)”

Docente: José Ignacio Alvarez Hamelin

Alumna	Padrón	Email
Valdivia Wong, Angie Isabella	103727	avaldivia@fi.uba.ar

**Fecha de entrega: 24/08/25**

Comentarios:

## I. Introducción:

El RTT (Round Trip Time) representa el tiempo que tarda un paquete en ir desde el emisor al receptor y volver. Para medirlo se utilizó el comando ping hacia la dirección IP 8.8.8.8 (servidor DNS de Google). El objetivo del presente trabajo es capturar los tiempos de respuesta durante dos horas, analizarlos estadísticamente y graficar su distribución en escala lineal y log-log.

## II. Desarrollo de la ejecución:

Como se observa en la imagen en la terminal hice el ping al ip de google y a la vez el resultado lo guardaba en ping\_output.txt para su posterior uso en el análisis del resultado; con el wireshark filtré usando el ICMP. Se observó que durante 2 hrs el ping anduvo bien ya que siempre había request y reply durante todo ese tiempo:

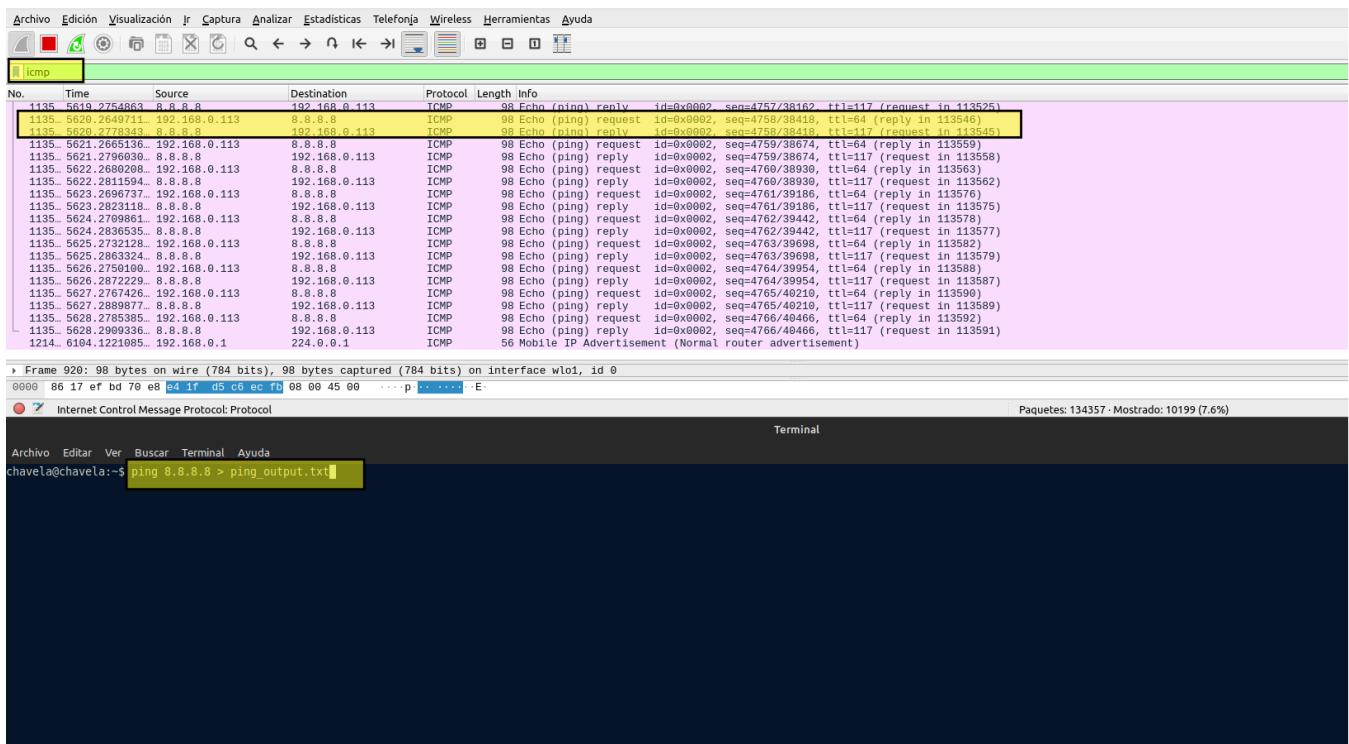


Figura 01: Captura de información

Una vez finalizado corroboro que se haya escrito bien usando el comando cat:

```
Archivo Editar Ver Buscar Terminal Ayuda
chavela@chavela:~$ ping 8.8.8.8 -c 40
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=16.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=15.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=117 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=117 time=12.1 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=117 time=12.2 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=117 time=12.2 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=117 time=13.5 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=117 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=117 time=13.5 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=117 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=117 time=13.8 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=117 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=117 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=27 ttl=117 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=28 ttl=117 time=13.7 ms
64 bytes from 8.8.8.8: icmp_seq=29 ttl=117 time=13.2 ms
64 bytes from 8.8.8.8: icmp_seq=30 ttl=117 time=13.9 ms
64 bytes from 8.8.8.8: icmp_seq=31 ttl=117 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=32 ttl=117 time=12.0 ms
64 bytes from 8.8.8.8: icmp_seq=33 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=34 ttl=117 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=35 ttl=117 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=36 ttl=117 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=37 ttl=117 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=38 ttl=117 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=39 ttl=117 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=40 ttl=117 time=11.8 ms
64 bytes from 8.8.8.8: icmp_seq=41 ttl=117 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=42 ttl=117 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=43 ttl=117 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=44 ttl=117 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=45 ttl=117 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=46 ttl=117 time=13.3 ms
64 bytes from 8.8.8.8: icmp_seq=47 ttl=117 time=13.3 ms
64 bytes from 8.8.8.8: icmp_seq=48 ttl=117 time=12.3 ms
```

Figura 02: verificación

Luego procedí a filtrar los datos que me servirán para graficar:

```
Archivo Editar Ver Buscar Terminal Ayuda
chavela@chavela:~$ cat ping_output.txt | grep "time=" | awk -F'time=' '{print $2}' | awk '{print $1}' > tiempos.txt
```

Figura 03: Selecciono la columna de time y lo guardo en tiempos.txt

Verifiqué que los contenidos estén bien:

```
Archivo Editar Ver Buscar Terminal Ayuda
chavela@chavela:~$ cat tiempos.txt
16.9
15.3
12.6
15.6
13.0
12.5
12.4
12.5
13.1
13.6
13.2
12.7
12.1
12.2
13.4
13.5
12.0
14.5
13.5
12.5
12.8
13.8
13.1
14.9
12.8
12.5
17.1
13.2
12.6
13.4
12.0
13.8
12.5
12.7
13.2
13.0
13.0
11.3
11.3
11.8
12.8
12.4
13.1
12.6
13.4
13.3
13.6
13.5
12.5
12.9
12.4
11.9
14.0
12.5
```

Figura 04: Extracción de los valores de tiempo en milisegundos

### III. Análisis estadístico y conclusión:

En este caso usé Python + matplotlib en GoogleColab para graficar usé el siguiente código:

```
[4] ✘ os  ⏎ import numpy as np
      import matplotlib.pyplot as plt

      # Cargar los tiempos en ms
      tiempos = np.loadtxt("tiempos.txt")

      # Estadísticas básicas
      n = len(tiempos)
      rtt_min = np.min(tiempos)
      rtt_max = np.max(tiempos)
      rtt_mean = np.mean(tiempos)
      rtt_median = np.median(tiempos)
      rtt_std = np.std(tiempos)

      # Mostrar en consola
      print("== Estadísticas RTT ==")
      print(f"Cantidad de muestras: {n}")
      print(f"Minímo: {rtt_min:.2f} ms")
      print(f" Máximo: {rtt_max:.2f} ms")
      print(f"Media: {rtt_mean:.2f} ms")
      print(f"Mediana: {rtt_median:.2f} ms")
      print(f"Desv. estándar: {rtt_std:.2f} ms")

      # Guardar estadísticas en un archivo
      with open("estadisticas.txt", "w") as f:
          f.write("== Estadísticas RTT ==\n")
          f.write(f"Cantidad de muestras: {n}\n")
          f.write(f"Minímo: {rtt_min:.2f} ms\n")
          f.write(f" Máximo: {rtt_max:.2f} ms\n")
          f.write(f"Media: {rtt_mean:.2f} ms\n")
          f.write(f"Mediana: {rtt_median:.2f} ms\n")
          f.write(f"Desv. estándar: {rtt_std:.2f} ms\n")

      # Histograma escala lineal
      plt.figure()
      plt.hist(tiempos, bins=50, edgecolor='black')
      plt.xlabel("Tiempo de respuesta (ms)")
      plt.ylabel("Frecuencia")
      plt.title("Distribución de tiempos de ping (escala lineal)")
      plt.savefig("lineal.png")

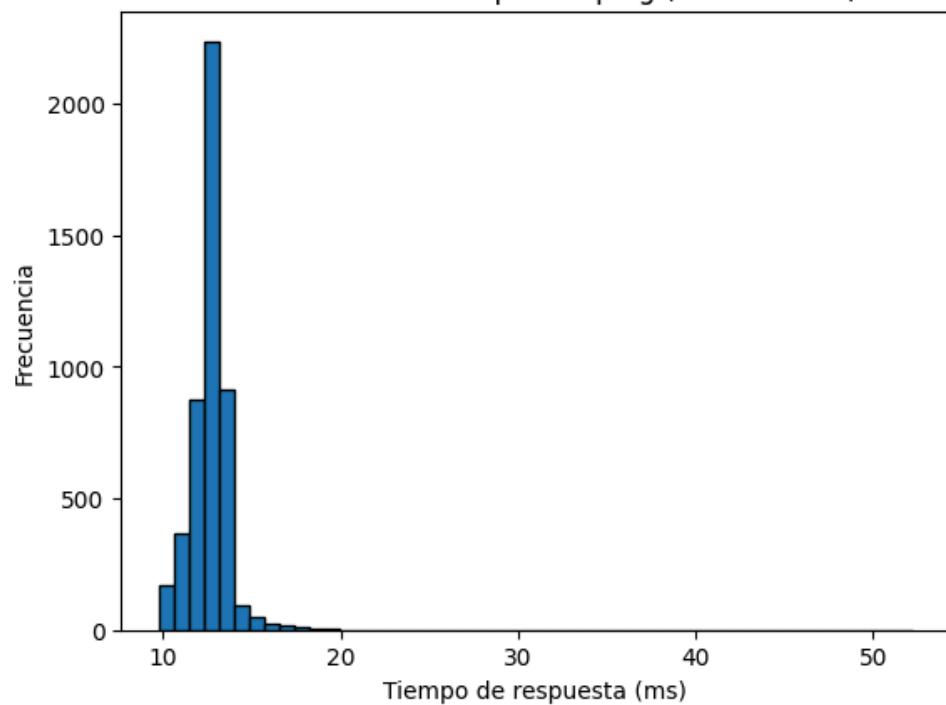
      # Histograma escala log-log
      plt.figure()
      plt.hist(tiempos, bins=50, edgecolor='black')
      plt.xscale('log')
      plt.yscale('log')
      plt.xlabel("Tiempo de respuesta (ms) [log]")
      plt.ylabel("Frecuencia [log]")
      plt.title("Distribución de tiempos de ping (escala log-log)")
      plt.savefig("loglog.png")

      print("\nGráficos guardados en lineal.png y loglog.png")
      print("Estadísticas guardadas en estadisticas.txt")
```

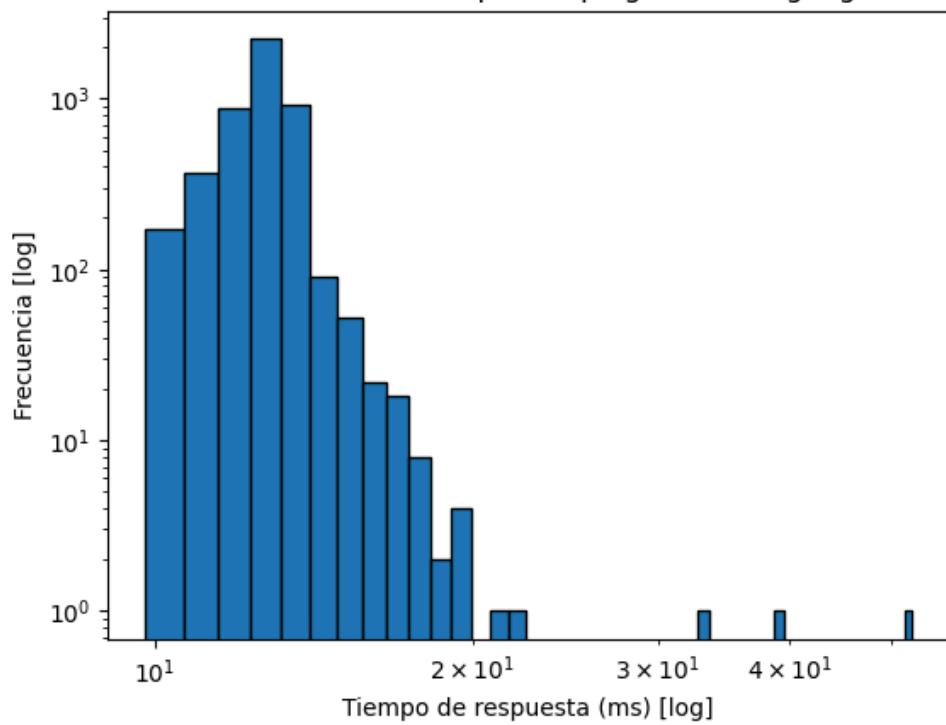
Figura 05: Código

Obteniendo como resultado los siguientes gráficos:

Distribución de tiempos de ping (escala lineal)



Distribución de tiempos de ping (escala log-log)



Y resultado final:

==== Estadísticas RTT ===

Cantidad de muestras: 4766

Mínimo: 9.79 ms

Máximo: 52.20 ms

Media: 12.67 ms

Mediana: 12.70 ms

Desv. estándar: 1.26 ms

La media y la mediana son prácticamente iguales (**12.67 ms vs. 12.70 ms**), lo que indica que la distribución de los RTT es **simétrica** y no está sesgada fuertemente hacia la izquierda o derecha.

La **desviación estándar baja (1.26 ms)** confirma que la mayoría de los tiempos están muy concentrados alrededor de la media, lo cual refleja una **conexión estable**.

El histograma en escala **lineal** muestra una clara concentración entre los **10 y 15 ms**, donde se encuentran casi todos los valores.

El histograma en escala **log-log** revela que, aunque la mayor parte de los datos están en un rango estrecho, existen **algunos valores atípicos** (picos de más de 30 ms e incluso hasta 52 ms). Estos son eventos esporádicos de mayor latencia, posiblemente asociados a congestión momentánea o fluctuaciones en la red.

El RTT mínimo de ~10 ms es consistente con una conexión local estable, y el máximo de ~52 ms sigue siendo bajo en comparación con retrasos que podrían indicar pérdidas o problemas de red.

En general, los resultados muestran que la conexión hacia 8.8.8.8 es **altamente estable**, con tiempos de respuesta **predecibles y bajos** (alrededor de 12–13 ms). Los pocos outliers detectados no afectan significativamente la media ni la mediana, aunque son visibles en la escala log-log