

# INF-256 Redes de Computadores

## Laboratorio 2

Agosto 25, 2025

Francisco Domínguez - Beatriz Vasquez - Catalina Broughton  
202104520-1 / 201904659-4 / 202011512-5  
Grupo 8

## 1 Introducción

El presente laboratorio tiene por objetivo principal aplicar de forma práctica conceptos fundamentales de la capa de aplicación y del uso de sockets en arquitecturas cliente-servidor: construir clientes que interactúen mediante TCP y UDP, obtener información desde servidores y finalmente registrar una frase filosófica mediante una solicitud HTTP POST. El enunciado del laboratorio describe tres etapas: (1) cliente TCP (comandos GET, JOKE, EXIT) que entrega puertos/servicios, (2) cliente UDP que consulta una frase, y (3) cliente TCP que realiza una petición HTTP POST al endpoint "Que es el ser sin la mitad faltante" para almacenar la frase junto a un identificador de grupo.

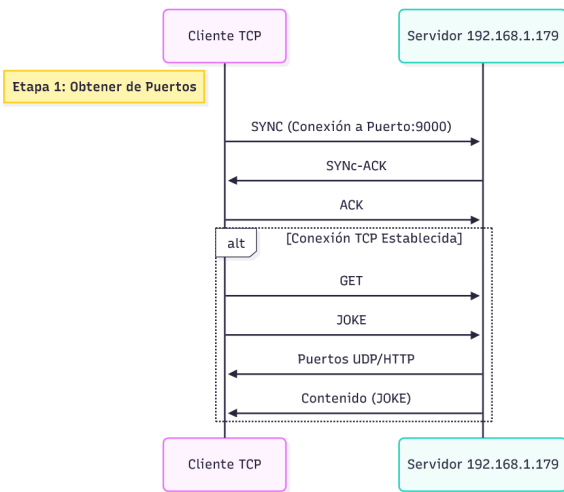
## 2 Respuestas a preguntas del enunciado

### 2.1 ¿Qué es el protocolo HTTP y para qué se usa? ¿Por qué es posible realizar solicitudes HTTP a través de un socket TCP?

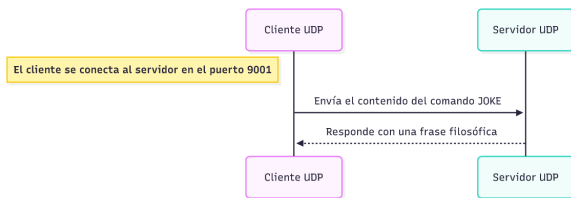
HTTP es un protocolo de aplicación para el intercambio de mensajes entre cliente y servidor web. Se utiliza principalmente para la navegación y el acceso a servicios en línea. Su ejecución sobre TCP es natural ya que HTTP requiere fiabilidad en la transmisión de datos, algo que TCP garantiza. Por ello, al abrir un socket TCP y enviar bytes en formato HTTP, se establece la comunicación esperada entre cliente y servidor.

### 2.2 Diagramas de interacción del laboratorio

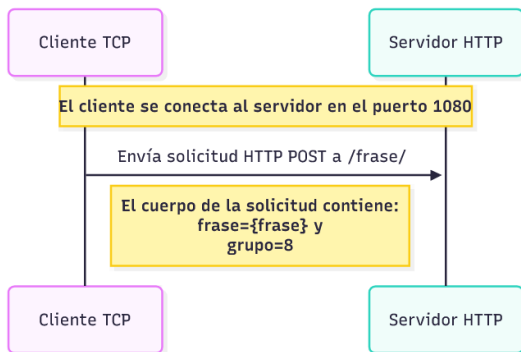
El laboratorio se estructura en tres etapas: cliente TCP, cliente UDP y cliente TCP-HTTP. Estos se realizaron conectados a la red de internet llamada **Lab\_cliente-servidor**. A continuación se presentan los diagramas correspondientes a las conexiones involucradas en el laboratorio:



(a) Conexión TCP



(b) Conexión UDP



(c) Conexión TCP-HTTP

## 2.3 Información obtenida de los servidores

Durante la experiencia se obtuvieron las siguientes respuestas:

**Servidor TCP** El servidor TCP entregó los puertos de conexión para servicios UDP y HTTP (UDP: 9001, HTTP: 1080), además de respuestas a los comandos GET, JOKE y EXIT.

```

PS C:\Users\Panch\OneDrive - Universidad Técnica Federico Santa María\2025-2\Redes\llb2_presencial> python TCP.py
Ingrese un comando (GET / JOKE / EXIT): GET
Servidor: PORT UDP-SERVER: 9001, PORT HTTP-SERVER: 9002.
Ingrese un comando (GET / JOKE / EXIT): JOKE
Servidor: ¿Sabes por qué se extinguieron los mamuts? Porque faltaban Paputs
Ingrese un comando (GET / JOKE / EXIT): EXIT
Servidor: Conexión cerrada. Adios
Cerrando cliente...

```

**Servidor UDP** El servidor UDP respondió con la frase filosófica enviada al cliente.

```

PS C:\Users\Panch\OneDrive - Universidad Técnica Federico Santa María\2025-2\Redes\llb2_presencial> python UDP.py
=== FRASE FILOSÓFICA (UDP) ===
Que es el ser sin la mitad faltante

```

**Servidor HTTP** El servidor HTTP confirmó el almacenamiento de la frase mediante una respuesta con código de estado HTTP.

```

PS C:\Users\Panch\OneDrive - Universidad Técnica Federico Santa María\2025-2\Redes\llb2_presencial> python TCP-HTTP.py
POST /frase/ HTTP/1.1
Host: 192.168.1.179:1000
Content-Type: text/plain
Content-Length: 51
Connection: close

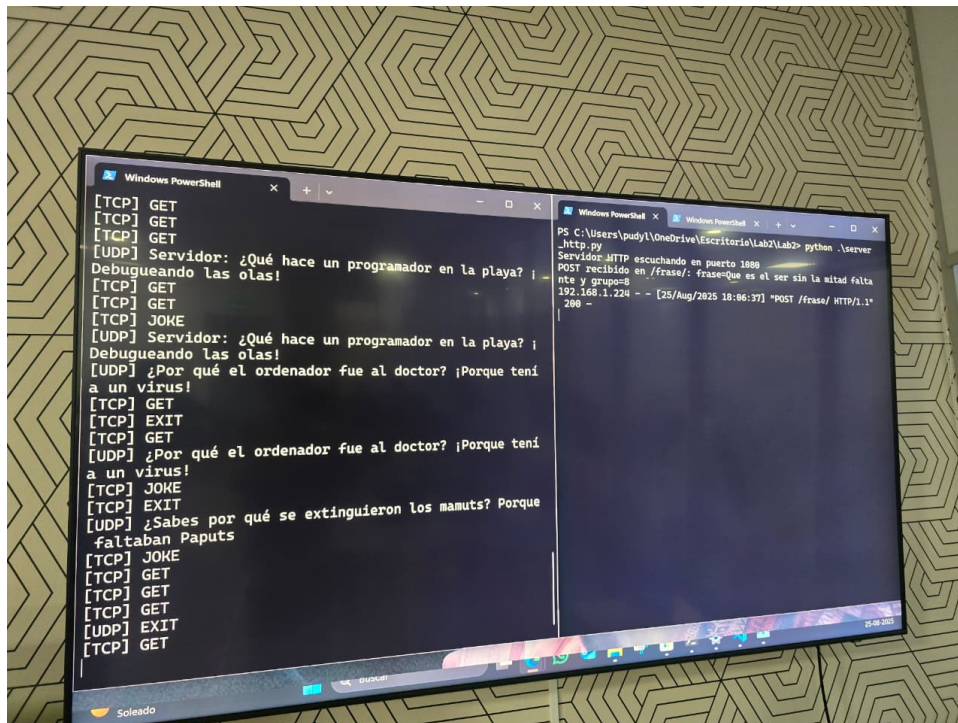
frase=Que es el ser sin la mitad faltante y grupo=8

[Éxito] Respuesta del servidor HTTP recibida:
HTTP/1.0 200 OK
Server: BaseHTTP/0.6 Python/3.12.3
Date: Mon, 25 Aug 2025 22:06:37 GMT
Content-Type: application/json

PS C:\Users\Panch\OneDrive - Universidad Técnica Federico Santa María\2025-2\Redes\llb2_presencial>

```

**Resultado final** El flujo completo de la práctica se cerró con la confirmación final de interacción cliente-servidor.



## 2.4 Otros protocolos de la capa de aplicación

A continuación, se describen otros cinco protocolos:

- **FTP** (File Transfer Protocol): puerto 21 (control) y 20 (datos), usa TCP, sirve para transferencia de archivos.
- **SMTP** (Simple Mail Transfer Protocol): puerto 25, usa TCP, permite el envío de correos electrónicos.
- **DNS** (Domain Name System): puerto 53, usa UDP para consultas y TCP para transferencias de zona, resuelve nombres de dominio en direcciones IP.
- **SSH** (Secure Shell): puerto 22, usa TCP, proporciona acceso remoto seguro a sistemas.
- **DHCP** (Dynamic Host Configuration Protocol): puertos 67 (servidor) y 68 (cliente), usa UDP, permite asignar parámetros de red a hosts de manera dinámica.

Comparados con HTTP, todos son protocolos de aplicación, pero difieren en su propósito y en el transporte utilizado. HTTP usa TCP por la necesidad de transmisión fiable de datos web, mientras que otros como DNS y DHCP usan UDP para mayor rapidez en transacciones cortas.

## 3 Conclusiones

- El laboratorio permitió comprender la interacción entre protocolos de transporte (TCP, UDP) y de aplicación (HTTP).
- Se evidenció que la elección de TCP o UDP depende de las necesidades de la aplicación: fiabilidad versus velocidad.
- La experiencia práctica refuerza los conceptos de arquitecturas cliente-servidor y el rol de los sockets en la comunicación de red.

## 4 Bibliografía

### References

- [1] Kurose, J. F., & Ross, K. W. (2017). Computer Networking: A Top-Down Approach (7th ed.). Pearson.
- [2] Díaz, J. (2025). Capítulo 1 y 2 – INF-256 Redes de Computadores. Universidad Técnica Federico Santa María.