

Informe L2-Grupo07

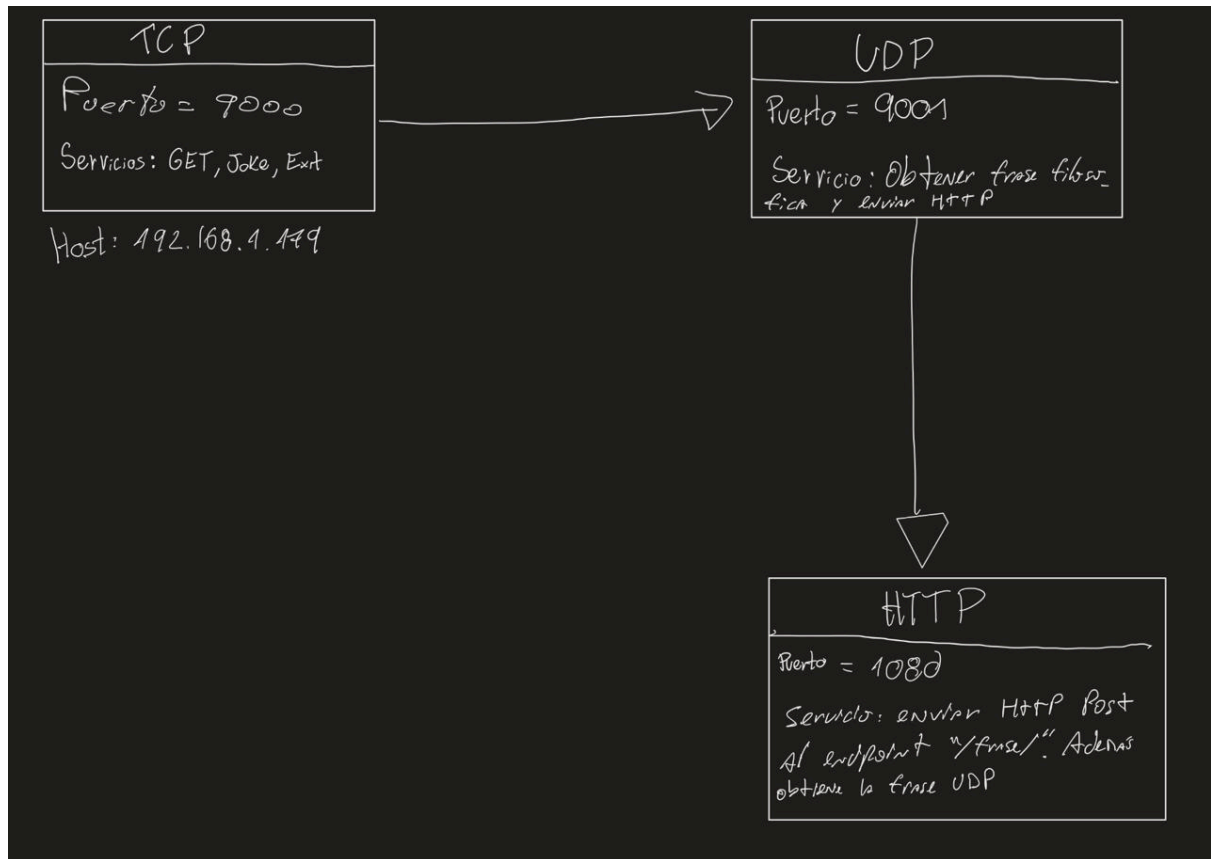
Pregunta 1

El protocolo HTTP (HyperText Transfer Protocol) corresponde a un estándar para la transferencia de información en la web. Permite a los navegadores web solicitar y recibir recursos. Éste protocolo funciona bajo un modelo cliente-servidor, donde el cliente envía una solicitud y el servidor responde con el recurso o información solicitada.

Con respecto a porqué es posible realizar solicitudes HTTP a través de un socket TCP, esto viene dado porque HTTP se apoya en TCP ya que este último nos proporciona confiabilidad y orden a la hora de transmitir paquetes. Al hacer una solicitud HTTP el cliente abre un socket TCP hacia el servidor, y dentro de ese canal envía un texto estructurado según las reglas de HTTP. HTTP no define cómo se envían los datos a nivel de red, sino que define la estructura que deben manejar estos mensajes (como headers, body, etc).

Pregunta 2

El diagrama realizado corresponde al siguiente:



Pregunta 3

La información obtenida de los servidores se puede apreciar en las siguientes fotos:

The screenshot shows the VS Code editor interface. The Explorer sidebar on the left displays the project structure for 'LABORATORIOS-INF256', including folders 'L1-Grupo07' and 'L2-Grupo07', and files like 'config.py', 'http_cliente.py', 'run_lab2.py', 'tcp_cliente.py', 'udp_cliente.py', 'README.md', and 'L1-Grupo07.zip'. The main editor window shows the 'config.py' file with the following code:

```
1 L2-Grupo07 > config.py > ...
2 import os
3
4 # Host/IP del servidor (hay que ajustarla el viernes en el lab según la que nos den)
5 HOST = "192.168.1.179"
6
7 # Puertos por defecto
8 TCP_PORT = 9000
9 UDP_PORT = None
10 HTTP_PORT = None
11
12 # ID del grupo para HTTP
13 GRUPO_ID = "GRUPO-07-MARTIN-IBANEZ-BENJAMIN-URRUTIA-DAVID-KRIPPER"
14
15 # Tiempo en segundos para operaciones de socket
16 SOCKET_TIMEOUT = 5
17
```

Below the editor, the TERMINAL panel shows the output of running 'python3 run_lab2.py':

```
davidkripper@Davids-MacBook-Air L2-Grupo07 % python3 run_lab2.py
== Fase TCP ==
Escribe comandos (GET, JOKE, EXIT). Escribe EXIT para terminar.

> Ingresa un comando: GET
< Respuesta:
PORT UDP-SERVER: 9001, PORT HTTP-SERVER: 9002.
> Ingresa un comando: |
```

The status bar at the bottom indicates the file is 'main*' at line 16, column 19, with 4 spaces, UTF-8 encoding, LF line endings, and Python 3.12.7 64-bit.

The screenshot shows the VS Code editor interface with the 'http_cliente.py' file open in the main editor. The Explorer sidebar is the same as in the first screenshot. The 'http_cliente.py' file contains the following code:

```
1 import socket
2 import os
3 import json
4 from config import HOST, SOCKET_TIMEOUT
5
6 def construir_post(frase, grupo_id, puerto_http):
7     body = json.dumps({"frase": frase, "grupo": grupo_id}, ensure_ascii=False).encode("utf-8")
8     req = (
9         "POST /frase/ HTTP/1.1\r\n"
10         "Host: " + HOST + ":" + str(puerto_http) + "\r\n"
11         "Content-Type: application/json; charset=utf-8\r\n"
12         "Content-Length: " + str(len(body)) + "\r\n"
13         "Connection: close\r\n\r\n"
14     ).encode("utf-8") + body
15     print(f"REQUEST: {req}")
16     return req
17
```

The TERMINAL panel shows the output of running 'python3 http_cliente.py post_frase':

```
Escribe comandos (GET, JOKE, EXIT). Escribe EXIT para terminar.

> Ingresa un comando: JOKE
< Respuesta:
¿Por qué el ordenador fue al doctor? ¡Porque tenía un virus!
LEGAT: ¿Por qué el ordenador fue al doctor? ¡Porque tenía un virus!
Respuesta Guardada
None
CAMBIANDO2: 9001
CAMBIANDO1: 9002
CAMBIANDO: ¿Por qué el ordenador fue al doctor? ¡Porque tenía un virus!

== Resumen TCP ==
UDP_PORT: 9001
HTTP_PORT: 9002
JOKE: ¿Por qué el ordenador fue al doctor? ¡Porque tenía un virus!

== Fase UDP ==
Frase UDP: La sabiduría empieza por reconocer la vulnerabilidad.

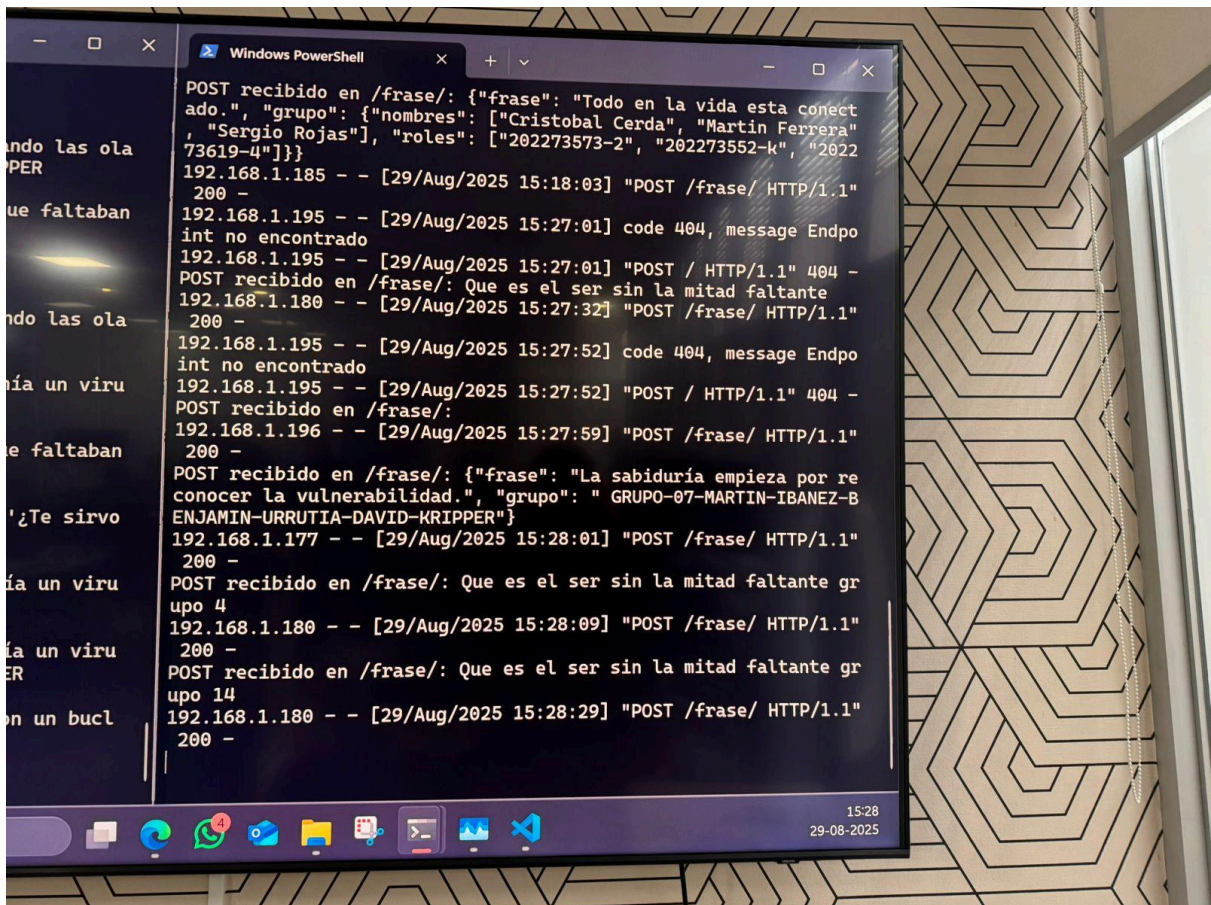
== Fase HTTP ==
REQUEST: b'POST /frase/ HTTP/1.1\r\nHost: 192.168.1.179:9002\r\nContent-Type: application/json; charset=utf-8\r\nContent-Length: 134\r\nConnection: close\r\n\r\n{"frase": "La sabidur\xac3\xada empieza por reconocer la vulnerabilidad.", "grupo": " GRUPO-07-MARTIN-IBANEZ-BENJAMIN-URRUTIA-DAVID-KRIPPER"}'
Traceback (most recent call last):
  File "/Users/davidkripper/Library/CloudStorage/OneDrive-Personal/Universidad/FourthY/2/Redes/Laboratorios-INF256/L2-Grupo07/run_lab2.py", line 40, in <module>
    main()
  File "/Users/davidkripper/Library/CloudStorage/OneDrive-Personal/Universidad/FourthY/2/Redes/Laboratorios-INF256/L2-Grupo07/run_lab2.py", line 34, in main
    status, body = post_frase(frase, GRUPO_ID, http_port)
  File "/Users/davidkripper/Library/CloudStorage/OneDrive-Personal/Universidad/FourthY/2/Redes/Laboratorios-INF256/L2-Grupo07/http_cliente.py", line 22, in post_frase
    s.connect((HOST, puerto_http))
```

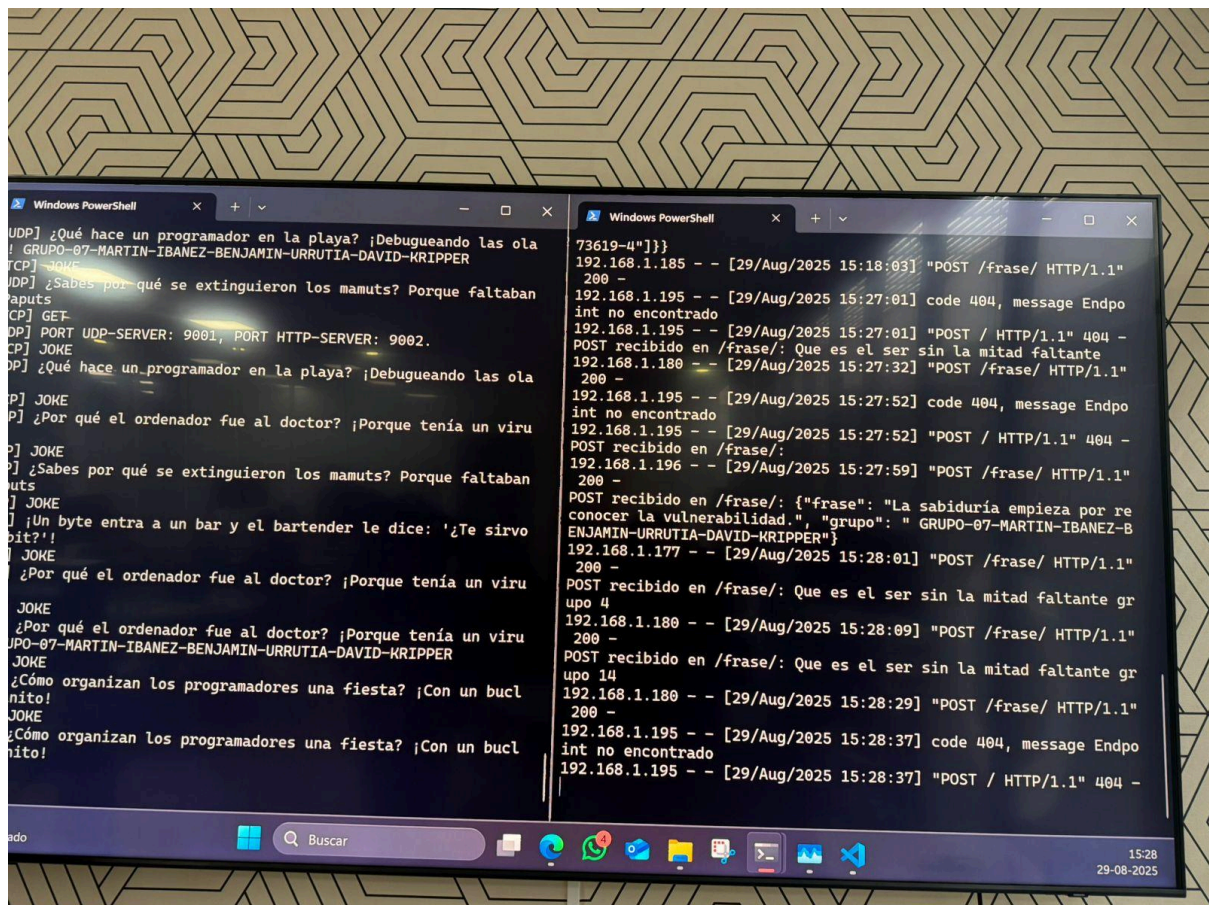
The status bar at the bottom indicates the file is 'main*' at line 20, column 58, with 4 spaces, UTF-8 encoding, LF line endings, and Python 3.12.7 64-bit.

The image shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'LABORATORIOS-INF256' with subfolders 'L1-Grupo07' and 'L2-Grupo07'. The 'L2-Grupo07' folder contains files like 'config.py', 'http_cliente.py', 'Lab_2_redes_2025_2...', 'README.md', 'run_lab2.py', 'tcp_cliente.py', 'udp_cliente.py', and '.gitignore'. The 'config.py' file is open in the editor, showing the following code:

```
1 L2-Grupo07 > config.py > ...
2 import os
3
4 HOST = "192.168.1.179"
5
6 TCP_PORT = 9000
7 UDP_PORT = 9001
8 HTTP_PORT = 1080
9
10 GRUPO_ID = " GRUPO-07-MARTIN-IBANEZ-BENJAMIN-URRUTIA-DAVID-KRIPPER"
11
12 SOCKET_TIMEOUT = 5
13
```

The terminal shows the output of running 'python3 run_lab2.py'. The output includes a welcome message, a list of commands (GET, JOKE, EXIT), and a series of responses from the server. The responses include a joke, a status change, and a list of names. The terminal also shows the output of a 'curl' command, which returns a JSON response with a message and a group ID.





Pregunta 4

Definimos nuestro protocolo HTTP a continuación:

- HTTP
 - Puertos utilizados: 80 / 443
 - Socket utilizado: TCP
 - Función principal: Transferencia de información en la web.

Otros protocolos de la capa de aplicación encontrados corresponden a:

- DNS
 - Puertos utilizados: 53
 - Socket utilizado: UDP/TCP
 - Función principal: Conversión de nombres de dominio en direcciones IP numéricas.
- SSH

- Puertos utilizados: 22
 - Socket utilizado: TCP
 - Función principal: Proporciona un acceso remoto seguro.
- IMAP
 - Puertos utilizados: 143 / 993
 - Socket utilizado: TCP
 - Función principal: Permite acceder y gestionar correos electrónicos desde varios dispositivos.
- FTP
 - Puertos utilizados: 20 / 21
 - Socket utilizado: TCP
 - Función principal: Permite la transferencia de archivos desde un servidor hacia un cliente.
- SMTP
 - Puertos utilizados: 25 / 465 / 587
 - Socket utilizado: TCP
 - Función principal: Permite mandar y recibir mails a través de la red.

Fuentes externas de información

1. <https://www.rfc-editor.org/rfc/rfc9110>
2. <https://developer.mozilla.org/es/docs/Web/HTTP>
3. <https://aws.amazon.com/es/what-is/smtp/>
4. <https://www.cloudflare.com/es-es/learning/dns/what-is-dns/>
5. <https://www.cbtnuggets.com/common-ports/what-is-port-22>
6. <https://support.microsoft.com/es-es/office/-qu%C3%A9-son-imap-y-pop-ca2c5799-49f9-4079-aeef-ddca85d5b1c9>
7. <https://learn.microsoft.com/en-us/connectors/ftp/>