

Laboratorio 1

Redes

José Luis Gramajo Moraga, Carné 22907

17 de julio de 2025 Guatemala, Guatemala

Parte 1 (Grupal)

Primera parte: transmisión de códigos

- ¿Qué esquema (código) fue más fácil de transmitir y por qué? ¿Qué esquema (código) fue más difícil de transmitir y por qué?
 - Más fácil: Código Morse. Nos resultó más sencillo porque el Morse es muy popular y familiar: basta con distinguir “bips” cortos y largos, lo cual es intuitivo. Además, muchos de nosotros ya habíamos visto o usado este esquema, por lo que no hubo curva de aprendizaje en la representación de los pulsos.
 - Más difícil: Código Baudot. Fue más complejo porque nunca lo habíamos practicado: cada carácter requiere una secuencia fija de 5 bits, y mantener la sincronización (pausas exactas entre bits y entre caracteres) resultó confuso. La falta de experiencia generó mayor duda al contar los “unos” y “ceros” en tiempo real.
- ¿Qué esquema tuvo menos errores (incluir datos que lo evidencien)?

Para cada esquema transmitimos **6 mensajes** de aproximadamente **12 caracteres** cada uno (≈ 72 caracteres en total):

- **Código Morse**
 - Errores registrados: **3 caracteres malinterpretados**
 - Tasa de error: $3 / 72 \approx 4,2 \%$
- **Código Baudot**
 - Errores registrados: **15 caracteres malinterpretados**
 - Tasa de error: $15 / 72 \approx 20,8 \%$

Conclusión: El código Morse presentó una tasa de error significativamente menor (4,2 % vs. 20,8 %), lo que confirma que fue más preciso y confiable en nuestra práctica.

Segunda parte: transmisión “empaquetada”

- ¿Qué dificultades involucra el enviar un mensaje de forma “empaquetada”?

1. Orden de llegada

- Los audios pueden llegar desordenados debido a la red
- WhatsApp no garantiza que lleguen en secuencia

2. Identificación de paquetes

- Difícil saber qué audio corresponde a qué parte del mensaje
- Necesidad de identificar cada fragmento verbalmente

3. Calidad del audio

- Ruido, cortes o mala calidad pueden corromper partes del mensaje
- Dificultad para entender algunos fragmentos

4. Timing y pausas

- Coordinar cuándo enviar cada audio
- Evitar saturar al receptor con muchos mensajes simultáneos

5. Reensamblaje manual

- El receptor debe escuchar todos los audios y reconstruir mentalmente
- Mayor probabilidad de errores al juntar las partes

6. Pérdida de contexto

- Sin el mensaje completo, es difícil verificar si falta algún fragmento
- No hay forma automática de detectar errores

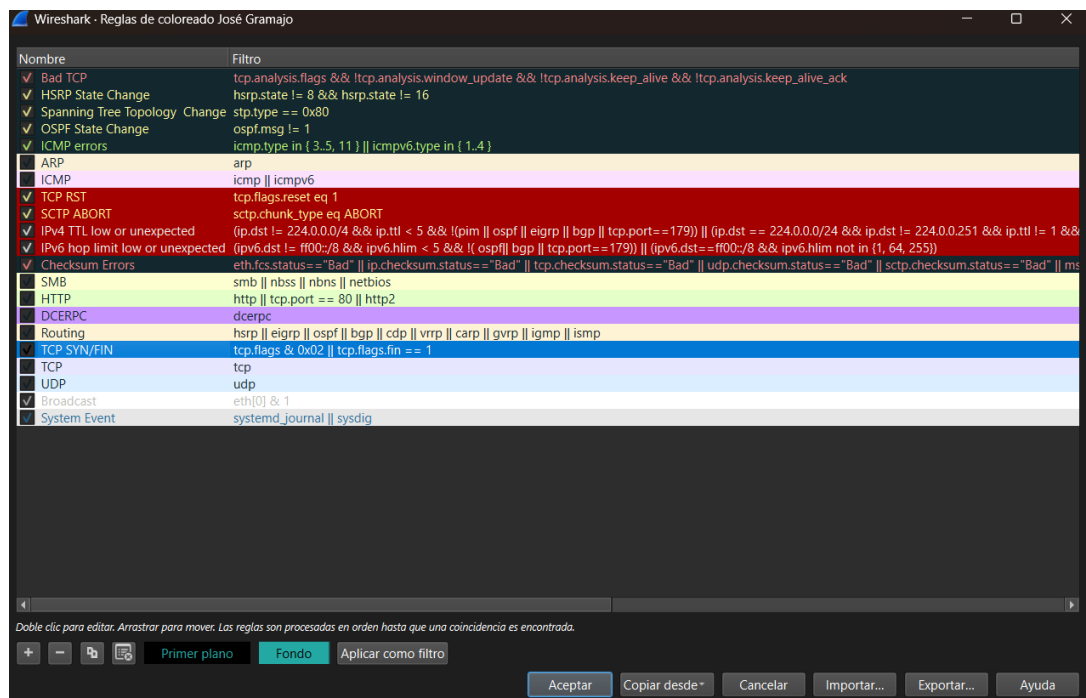
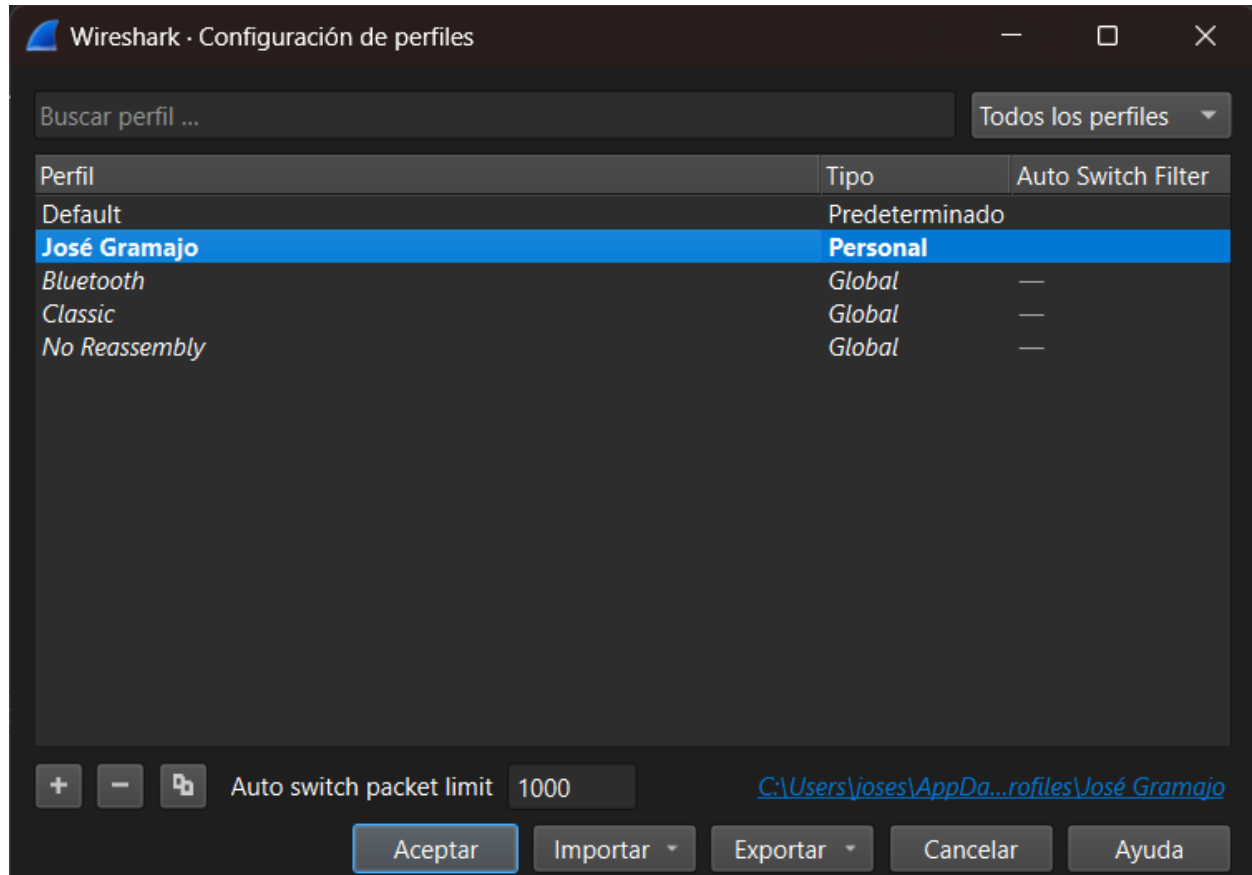
Tercera parte: conmutación de mensajes

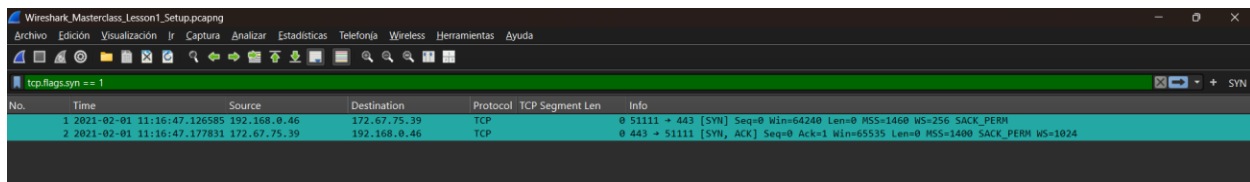
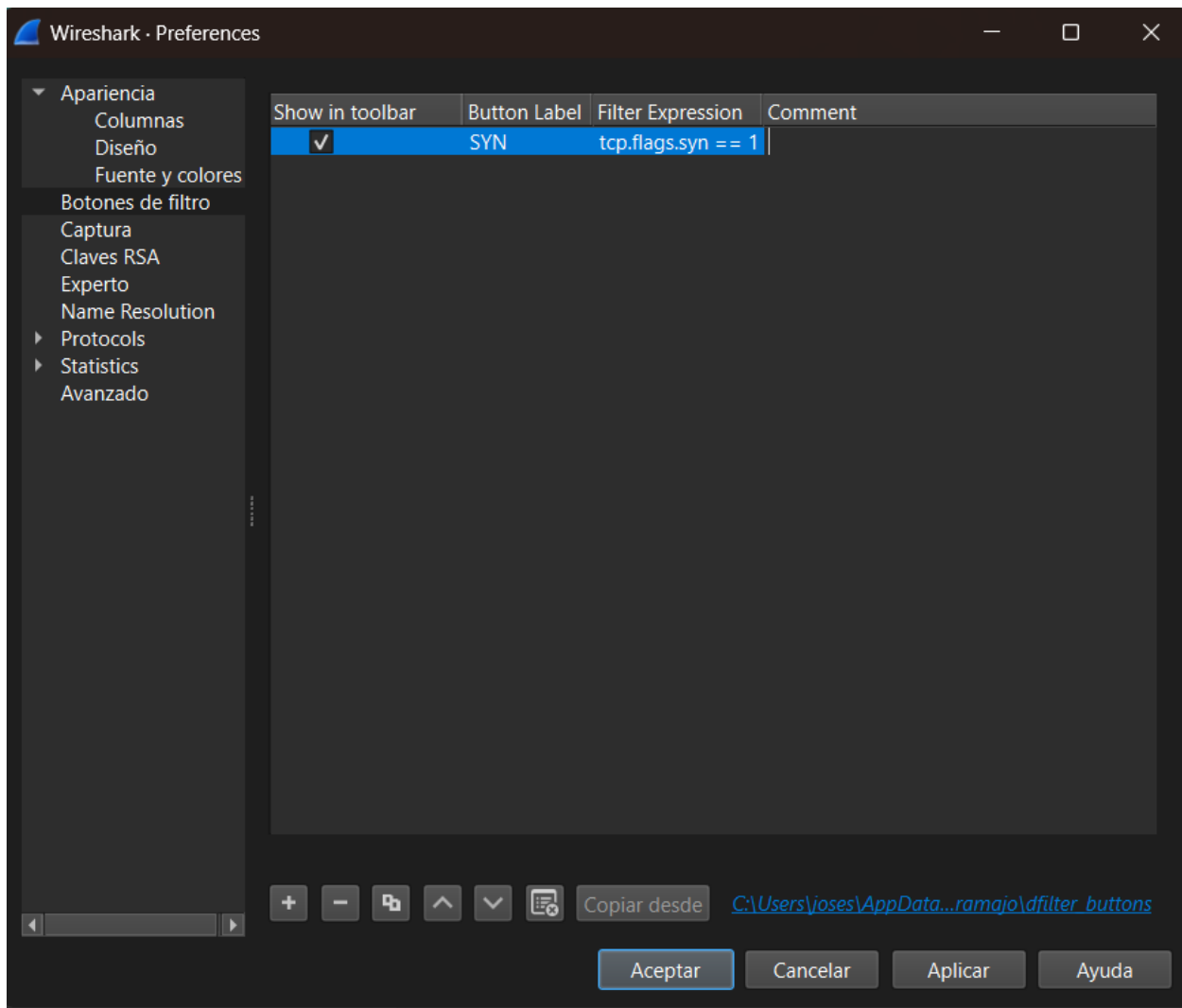
- ¿Qué ventajas/desventajas se tienen al momento de agregar más conmutadores al sistema?
 - Agregar más conmutadores permite repartir la carga de mensajes, haciendo el sistema más eficiente y escalable. Sin embargo, también lo hace más complejo, aumenta la necesidad de coordinación y puede generar más errores si no se gestionan correctamente los turnos y los destinatarios.
- ¿Qué posibilidades incluye la introducción de un conmutador en el sistema?
 - La introducción de un conmutador centraliza la comunicación, facilitando que los mensajes lleguen correctamente al destinatario sin que los clientes tengan que comunicarse entre sí directamente. También permite controlar el flujo de mensajes y mantener un orden, pero introduce el riesgo de saturar al conmutador si no se organiza adecuadamente.

Protocolo Conmutador

Implementamos un sencillo protocolo de conmutación de circuitos en tres fases: primero, en el SETUP el conmutador, tras anunciarse “libre”, recibe la solicitud de envío junto con el identificador del destinatario y reserva el canal; luego, en la fase TRANSFER el emisor transmite el mensaje en fragmentos que el conmutador reensambla y reenvía, confirmando cada tramo con ACKs para garantizar fiabilidad; al finalizar, en el TEARDOWN el emisor envía un END-OF-MESSAGE, el destinatario confirma con ACK-EOM y el conmutador libera los recursos. Para no sobrecargarlo, sólo admite una reserva activa a la vez (rechaza nuevas solicitudes con BUSY o las encola), aplica time-outs automáticos y puede gestionar colas FIFO o prioridades. Añadir más conmutadores distribuye la carga y mejora la escalabilidad, aunque incrementa la complejidad de coordinación entre ellos.

Parte 2 (Individual)





Wireshark_Masterclass_Lesson1_Setup.pcapng

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización... <Ctrl-F>

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
3	2021-02-01 11:16:47.177959	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
8	2021-02-01 11:16:47.267999	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=518 Ack=1809 Win=131584 Len=0
19	2021-02-01 11:16:49.123997	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1371 Ack=2368 Win=130816 Len=0
27	2021-02-01 11:16:49.389546	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1371 Ack=10871 Win=131584 Len=0
47	2021-02-01 11:16:49.465037	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1371 Ack=37921 Win=131584 Len=0
52	2021-02-01 11:16:49.465399	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1371 Ack=42508 Win=131584 Len=0
58	2021-02-01 11:16:49.524874	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [ACK] Seq=1371 Ack=48840 Win=131584 Len=0
5	2021-02-01 11:16:47.255959	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [ACK] Seq=1 Ack=518 Win=67584 Len=0
12	2021-02-01 11:16:49.073867	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [ACK] Seq=1809 Ack=582 Win=67584 Len=0
15	2021-02-01 11:16:49.079528	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [ACK] Seq=2337 Ack=674 Win=67584 Len=0
16	2021-02-01 11:16:49.079528	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [ACK] Seq=2337 Ack=1340 Win=68608 Len=0
18	2021-02-01 11:16:49.099770	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [ACK] Seq=2368 Ack=1371 Win=68608 Len=0
1	2021-02-01 11:16:47.126585	192.168.0.46	172.67.75.39	TCP		0 51111 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	2021-02-01 11:16:47.177831	172.67.75.39	192.168.0.46	TCP		0 443 → 51111 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM WS=1024
83	2021-02-01 11:16:49.645336	172.67.75.39	192.168.0.46	QUIC		Protected Payload (KPB)
84	2021-02-01 11:16:49.645336	172.67.75.39	192.168.0.46	QUIC		Protected Payload (KPB)
14	2021-02-01 11:16:49.074157	192.168.0.46	172.67.75.39	TLSv1.3		31 Application Data
17	2021-02-01 11:16:49.079528	172.67.75.39	192.168.0.46	TLSv1.3		31 Application Data
86	2021-02-01 11:16:49.645802	192.168.0.46	172.67.75.39	QUIC		Protected Payload (KPB), DCID=010727f5fc9976b540072bf7c29946618bd397b3
87	2021-02-01 11:16:49.646088	192.168.0.46	172.67.75.39	QUIC		Protected Payload (KPB), DCID=010727f5fc9976b540072bf7c29946618bd397b3
85	2021-02-01 11:16:49.645336	172.67.75.39	192.168.0.46	QUIC		Protected Payload (KPB)
81	2021-02-01 11:16:49.645336	172.67.75.39	192.168.0.46	QUIC		Handshake, SCID=010727f5fc9976b540072bf7c29946618bd397b3
60	2021-02-01 11:16:49.593243	172.67.75.39	192.168.0.46	QUIC		Initial, SCID=010727f5fc9976b540072bf7c29946618bd397b3, PKN: 0, ACK
65	2021-02-01 11:16:49.606789	192.168.0.46	172.67.75.39	QUIC		Handshake, DCID=010727f5fc9976b540072bf7c29946618bd397b3
9	2021-02-01 11:16:49.048600	192.168.0.46	172.67.75.39	TLSv1.3		64 Change Cipher Spec, Application Data
10	2021-02-01 11:16:49.048918	192.168.0.46	172.67.75.39	TLSv1.3		92 Application Data
61	2021-02-01 11:16:49.606275	172.67.75.39	192.168.0.46	QUIC		Initial, SCID=010727f5fc9976b540072bf7c29946618bd397b3
26	2021-02-01 11:16:49.389360	172.67.75.39	192.168.0.46	TLSv1.3		157 Application Data
66	2021-02-01 11:16:49.608420	192.168.0.46	172.67.75.39	QUIC		Protected Payload (KPB), DCID=010727f5fc9976b540072bf7c29946618bd397b3
64	2021-02-01 11:16:49.606275	172.67.75.39	192.168.0.46	QUIC		Handshake, SCID=010727f5fc9976b540072bf7c29946618bd397b3
51	2021-02-01 11:16:49.465251	172.67.75.39	192.168.0.46	TLSv1.3		207 Application Data, Application Data
7	2021-02-01 11:16:47.267985	172.67.75.39	192.168.0.46	TLSv1.3		348 Application Data

0000 50 19 f9 d1 2e aa 34 f6 46 a3 29 0b 08 00 45 00 X...K...E
0010 00 28 6e e4 40 00 80 0e d3 ea c0 a8 00 2e ac 43 (n...C
0020 4b 27 c7 a7 01 bb 0e 47 8a 2e c4 9f 08 87 50 10 K'...G...P
0030 02 02 c6 92 00 00

```

Windows PowerShell
PS C:\> ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet 2:

Sufijo DNS específico para la conexión. . :
Vínculo: dirección IPv6 local. . . : fe80::d8ea:dc32:3ae3:67e0%17
Dirección IPv4. . . . . : 192.168.56.1
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . . :

Adaptador de LAN inalámbrica Local Area Connection* 1:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Local Area Connection* 2:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . :
Dirección IPv6. . . . . : 2800:98:111e:72e::5
Dirección IPv6. . . . . : 2800:98:111e:72e:15b8:c4ca:ff39:72c9
Dirección IPv6 temporal. . . . . : 2800:98:111e:72e:1d98:a96d:3c89:b16
Vínculo: dirección IPv6 local. . . : fe80::3b11:d5f:d4d6:df60%8
Dirección IPv4. . . . . : 192.168.1.7
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . : fe80::1a34:afff:fe85:e139%8
192.168.1.1

Adaptador de Ethernet Bluetooth Network Connection:

```

Adaptador de LAN inalámbrica Wi-Fi

- IPv4: 192.168.1.7
- Máscara: 255.255.255.0
- Puerta de enlace: 192.168.1.1

lab1_22907_2020717163227_0001

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Filter: http

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
218	2025-07-17 16:32:33.221202	192.168.1.7	128.119.245.12	HTTP	533	GET /wireshark-labs/INTRO-wireshark-file.html HTTP/1.1
239	2025-07-17 16:32:33.329320	128.119.245.12	192.168.1.7	HTTP	438	HTTP/1.1 200 OK (text/html)

Ethernet II, Src: RealtekGroup (8c:8e:38:4d:34:f5), Dst: Realtek (8c:8e:38:4d:34:f5), Encapsulation: IEEE 802.3 (Standard)
 Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.7
 Transmission Control Protocol, Src Port: 80, Dst Port: 54765, Seq: 1, Ack: 534, Len: 438
 Hypertext Transfer Protocol
 HTTP/1.1 200 OK\r\n
 Response Version: HTTP/1.1
 Status Code: 200
 [Status Code Description: OK]
 Response Phrase: OK
 Date: Thu, 17 Jul 2025 22:31:14 GMT\r\n
 Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
 Last-Modified: Thu, 17 Jul 2025 05:59:01 GMT\r\n
 ETag: "51-63a19d9c5911"\r\n
 Accept-Ranges: bytes\r\n
 Content-Length: 81\r\n
 Keep-Alive: timeout=5, max=100\r\n
 Connection: Keep-Alive\r\n
 Content-Type: text/html; charset=UTF-8\r\n
 \r\n
 [Request in Frame: 218]
 [Time since request: 0.108118000 seconds]
 [Request URI: /wireshark-labs/INTRO-wireshark-file.html]
 [Full request URI: http://lab1_22907_2020717163227_0001/wireshark-labs/INTRO-wireshark-file.html]

TCP Segment Len (tcp.len)

Packets: 342 - Displayed: 2 (0.6%) - Percido: 0 (0.0%)

Perfil: José Gran...

- 1) Versión HTTP del navegador (cliente):
 - a) En la Request Line del GET (#326) aparece HTTP/1.1 → HTTP/1.1.
- 2) Versión HTTP del servidor:
 - a) En la Status Line de la respuesta (#239) HTTP/1.1 200 OK → HTTP/1.1.
- 3) Lenguajes que acepta el navegador:
 - a) En el header Accept-Language:
es-419,es;q=0.9,es-ES;q=0.8,en;q=0.7,en-GB;q=0.6,en-US;q=0.5
Acepta español latinoamericano, español genérico, español de España, inglés genérico, inglés del Reino Unido e inglés de EE. UU.
- 4) Bytes de contenido devueltos:
 - a) En el header Content-Length: 81 → 81 bytes.
- 5) Dónde “escuchar” en caso de rendimiento pobre:
 - a) Lo ideal es en el switch de acceso (con puerto SPAN/mirror) o en el router/gateway de borde, donde pasa todo el tráfico de usuarios.
 - b) No conviene instalar Wireshark en el servidor de producción porque consume recursos CPU/I/O, degrada su rendimiento y no siempre ve el tráfico cifrado antes de procesarlo.

