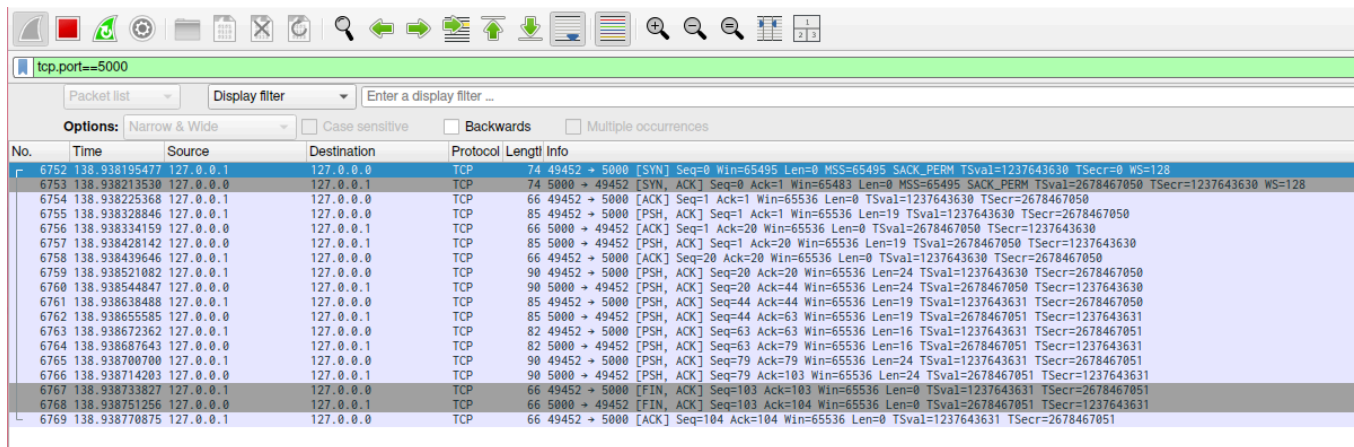


Informe de Captura de Paquetes

TCP



No.	Time	Source	Destination	Protocol	Length	Info
6752	138.938195477	127.0.0.1	127.0.0.0	TCP	74	49452 → 5000 [SYN, Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1237643630 TSecr=0 WS=128
6753	138.938213530	127.0.0.0	127.0.0.1	TCP	74	5000 → 49452 [SYN, Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=2678467050 TSecr=1237643630 WS=128
6754	138.938225368	127.0.0.1	127.0.0.0	TCP	66	49452 → 5000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1237643630 TSecr=2678467050
6755	138.938328846	127.0.0.1	127.0.0.0	TCP	85	49452 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=19 TSval=1237643630 TSecr=2678467050
6756	138.938341159	127.0.0.0	127.0.0.1	TCP	66	5000 → 49452 [ACK] Seq=1 Ack=20 Win=65536 Len=0 TSval=2678467050 TSecr=1237643630
6757	138.938428142	127.0.0.0	127.0.0.1	TCP	85	5000 → 49452 [PSH, ACK] Seq=1 Ack=20 Win=65536 Len=19 TSval=2678467050 TSecr=1237643630
6758	138.938439646	127.0.0.1	127.0.0.0	TCP	66	49452 → 5000 [ACK] Seq=20 Ack=20 Win=65536 Len=0 TSval=1237643630 TSecr=2678467050
6759	138.938521082	127.0.0.1	127.0.0.0	TCP	90	49452 → 5000 [PSH, ACK] Seq=20 Ack=20 Win=65536 Len=24 TSval=1237643630 TSecr=2678467050
6760	138.938544847	127.0.0.0	127.0.0.1	TCP	90	5000 → 49452 [PSH, ACK] Seq=20 Ack=44 Win=65536 Len=24 TSval=2678467050 TSecr=1237643630
6761	138.938638488	127.0.0.1	127.0.0.0	TCP	85	49452 → 5000 [PSH, ACK] Seq=44 Ack=44 Win=65536 Len=19 TSval=1237643631 TSecr=2678467050
6762	138.938655585	127.0.0.0	127.0.0.1	TCP	85	5000 → 49452 [PSH, ACK] Seq=44 Ack=63 Win=65536 Len=19 TSval=2678467051 TSecr=1237643631
6763	138.938672362	127.0.0.1	127.0.0.0	TCP	82	49452 → 5000 [PSH, ACK] Seq=63 Ack=63 Win=65536 Len=16 TSval=1237643631 TSecr=2678467051
6764	138.938687643	127.0.0.0	127.0.0.1	TCP	82	5000 → 49452 [PSH, ACK] Seq=63 Ack=79 Win=65536 Len=16 TSval=2678467051 TSecr=1237643631
6765	138.938700700	127.0.0.1	127.0.0.0	TCP	90	49452 → 5000 [PSH, ACK] Seq=79 Ack=79 Win=65536 Len=24 TSval=1237643631 TSecr=2678467051
6766	138.938714203	127.0.0.0	127.0.0.1	TCP	90	5000 → 49452 [PSH, ACK] Seq=79 Ack=103 Win=65536 Len=24 TSval=2678467051 TSecr=1237643631
6767	138.938733827	127.0.0.1	127.0.0.0	TCP	66	49452 → 5000 [FIN, ACK] Seq=103 Ack=103 Win=65536 Len=0 TSval=1237643631 TSecr=2678467051
6768	138.938751256	127.0.0.0	127.0.0.1	TCP	66	5000 → 49452 [FIN, ACK] Seq=103 Ack=104 Win=65536 Len=0 TSval=2678467051 TSecr=1237643631
6769	138.938770875	127.0.0.1	127.0.0.0	TCP	66	49452 → 5000 [ACK] Seq=104 Ack=104 Win=65536 Len=0 TSval=1237643631 TSecr=2678467051

```
adrianql5 ~/Desktop/REDES/Practica2 git-[?] main]- >> ./servidormay 5000
Servidor escuchando por el puerto 5000...
La dirección IP de la conexión entrante es: 127.0.0.1:40124
Recibiendo archivo y convirtiendo los datos
Todos los datos enviados
-----
Esperando nueva conexión. Escuchando por el puerto 5000...
La dirección IP de la conexión entrante es: 127.0.0.1:49452
Recibiendo archivo y convirtiendo los datos
Todos los datos enviados
-----
Esperando nueva conexión. Escuchando por el puerto 5000...
```

```
adrianql5 ~/Desktop/REDES/Practica2 git-[?] main]- >> ./cliente may archivo.txt 127.0.0.0 5000
Servidor conectado-----
Conexión cortada. Datos recibidos: 102 bytes
Conexión cortada. Datos enviados: 102 bytes
adrianql5 ~/Desktop/REDES/Practica2 git-[?] main]- >>
```

1. Identificación de Puertos Origen y Destino

- **Puerto de Origen:** En la captura, observamos que el puerto de origen del cliente es dinámico y asignado automáticamente por el sistema operativo (en este caso, **49452**).
- **Puerto de Destino:** Es el puerto donde está escuchando el servidor, definido en el programa como **5000**.

¿**Cómo se elige el puerto de origen?** El puerto de origen es asignado automáticamente por el sistema operativo desde el rango de **puertos efímeros** (generalmente del 49152 al 65535), asegurando que no haya conflictos con otros procesos.

2. Paquetes con Flag PSH

- En los paquetes capturados, se puede observar que algunos contienen el flag **PSH (Push)** en la cabecera TCP.
- **Análisis:**
 - Esto indica que el cliente o servidor solicitó al receptor que procesara inmediatamente los datos sin esperar más datos adicionales.
 - En este caso, los paquetes con PSH transportan los datos del mensaje enviado.

3. Análisis de la Etapa de Conexión (SYN y ACK)

La conexión TCP sigue el modelo de **handshake de tres vías**:

1. Primer Paquete (SYN):

- Paquete con el flag SYN enviado desde el cliente al servidor.
- **Detalles:**
 - Secuencia inicial (**Seq=0**).
 - Indicador de que el cliente desea iniciar la conexión.

2. Segundo Paquete (SYN-ACK):

- El servidor responde con un paquete con los flags SYN y ACK.
- **Detalles:**
 - Confirma la solicitud del cliente (**Ack=1**).
 - Envía su propia secuencia inicial (**Seq=0**).

3. Tercer Paquete (ACK):

- El cliente envía un paquete con el flag ACK.
- **Detalles:**
 - Confirma la recepción del paquete SYN-ACK (**Ack=1**).

Este intercambio establece la conexión TCP.

4. Análisis de la Etapa de Desconexión (FIN y ACK)

La desconexión TCP sigue un proceso de cuatro pasos:

1. Primer Paquete (FIN, ACK):

- El cliente envía un paquete con los flags FIN y ACK para indicar que desea cerrar la conexión.
- **Detalles:**
 - Secuencia del cliente (**Seq=103**).
 - Confirmación de recepción de datos anteriores (**Ack=103**).

2. Segundo Paquete (FIN, ACK):

- El servidor envía un FIN para cerrar su lado de la conexión.
- **Detalles:**
 - Secuencia del servidor (**Seq=103**).
 - Confirmación (**Ack=104**).

3. Tercer Paquete (ACK):

- El cliente confirma el FIN del servidor con un paquete ACK (**104**).

5. Comprobación del Tamaño de los Datos

El número de bytes de datos transmitidos en los paquetes con PSH coincide con el tamaño del mensaje definido en el programa porque estamos enviando por ejemplo en uno de los segmentos 24 bytes. El wireshark pone que payload de ese paquete es 24.

0000	00 00 00 00 00 00 00 00	00 00 00 00 08 00 45 00E-
0010	00 4c 35 ad 40 00 40 06	06 fe 7f 00 00 01 7f 00	..L5: @:
0020	00 00 c1 2c 13 88 ad 23	5c d3 5e f3 b6 9d 80 18	...# \^.....
0030	02 00 fe 3f 00 00 01 01	08 0a 49 c4 f1 6e 9f a6	...?....I..n..
0040	29 ea 4c 4b 4a 41 6c 61	6b 6c 64 66 6c 6b 61 66)LKJA1a kldflkaf
0050	73 64 6a 68 66 64 73 6b	6a 00	sdjhfdsk j:

6. Mensaje Grande y Segmentación TCP

1. Número de Paquetes Recibidos

- En la captura, se observa que el mensaje grande fue dividido en **3 paquetes con datos**:
 - Paquete con **Seq=1** y longitud de datos (**Len=1000**).
 - Paquete con **Seq=1001** y longitud de datos (**Len=1000**).
 - Paquete con **Seq=2001** y longitud de datos (**Len=553**).

El resto de los paquetes capturados corresponden a la conexión inicial, los ACK y el cierre de la conexión.

2. Números de Secuencia y Segmentación

- **Números de Secuencia:**
 - Cada paquete transporta un bloque consecutivo del flujo de datos:
 - Paquete 1: **Seq=1**, longitud de datos 1000 → bytes del 1 al 1000.
 - Paquete 2: **Seq=1001**, longitud de datos 1000 → bytes del 1001 al 2000.
 - Paquete 3: **Seq=2001**, longitud de datos 553 → bytes del 2001 al 2553.
 - Los números de secuencia se incrementan según el tamaño de los datos transportados en cada segmento.
- **Segmentación:**
 - El mensaje original fue dividido en segmentos de tamaño manejable por TCP:
 - Dos segmentos de 1000 bytes.
 - Un segmento final de 553 bytes, lo que indica el final del mensaje.

3. Segmentos con Flag PSH

- En la captura, los segmentos de datos (**Len=1000** y **Len=553**) contienen el flag **PSH (Push)**
 - Esto sugiere que el remitente (servidor) está solicitando que los datos se procesen inmediatamente al ser recibidos, sin esperar más datos.

No.	Time	Source	Destination	Protocol	Length	Info
223	4.527986277	127.0.0.1	127.0.0.0	TCP	74	46438 → 5000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=1240348786 TSecr=0 WS=128
224	4.527924168	127.0.0.0	127.0.0.1	TCP	74	5000 → 46438 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=1240348786 TSecr=1240348786 WS=128
225	4.527936631	127.0.0.1	127.0.0.0	TCP	66	46438 → 5000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1240348786 TSecr=2681172206
226	4.528166252	127.0.0.1	127.0.0.0	TCP	1066	46438 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=1000 TSval=1240348786 TSecr=2681172206
227	4.528174382	127.0.0.0	127.0.0.1	TCP	66	5000 → 46438 [ACK] Seq=1 Ack=1001 Win=68224 Len=0 TSval=2681172206 TSecr=1240348786
228	4.528296153	127.0.0.0	127.0.0.1	TCP	1066	5000 → 46438 [PSH, ACK] Seq=1 Ack=1001 Win=68224 Len=1000 TSval=2681172206 TSecr=1240348786
229	4.528312245	127.0.0.1	127.0.0.0	TCP	66	46438 → 5000 [ACK] Seq=1001 Ack=1001 Win=68224 Len=0 TSval=1240348786 TSecr=2681172206
230	4.528418104	127.0.0.1	127.0.0.0	TCP	1066	46438 → 5000 [PSH, ACK] Seq=1001 Ack=1001 Win=68224 Len=1000 TSval=1240348787 TSecr=2681172206
231	4.528486795	127.0.0.0	127.0.0.1	TCP	1066	5000 → 46438 [PSH, ACK] Seq=1001 Ack=2001 Win=68224 Len=1000 TSval=2681172207 TSecr=1240348787
232	4.528578610	127.0.0.1	127.0.0.0	TCP	619	46438 → 5000 [PSH, ACK] Seq=2001 Ack=2001 Win=68224 Len=553 TSval=1240348787 TSecr=2681172207
233	4.528645215	127.0.0.0	127.0.0.1	TCP	619	5000 → 46438 [PSH, ACK] Seq=2001 Ack=2554 Win=68480 Len=553 TSval=2681172207 TSecr=1240348787
234	4.528768696	127.0.0.1	127.0.0.0	TCP	66	46438 → 5000 [FIN, ACK] Seq=2554 Ack=2554 Win=68480 Len=0 TSval=1240348787 TSecr=2681172207
235	4.528842529	127.0.0.0	127.0.0.1	TCP	66	5000 → 46438 [FIN, ACK] Seq=2554 Ack=2555 Win=68480 Len=0 TSval=2681172207 TSecr=1240348787
236	4.528869703	127.0.0.1	127.0.0.0	TCP	66	46438 → 5000 [ACK] Seq=2555 Ack=2555 Win=68480 Len=0 TSval=1240348787 TSecr=2681172207

UDP

No.	Time	Source	Destination	Protocol	Length	Info
451	9.238234418	127.0.0.1	127.0.0.0	UDP	68	5001 → 5000 Len=26
452	9.238375323	127.0.0.1	127.0.0.1	UDP	68	5000 → 5001 Len=26
453	9.238549910	127.0.0.1	127.0.0.1	UDP	87	5001 → 5000 Len=45
454	9.238689984	127.0.0.1	127.0.0.1	UDP	87	5000 → 5001 Len=45
455	9.238825558	127.0.0.1	127.0.0.1	UDP	65	5001 → 5000 Len=23
456	9.238918833	127.0.0.1	127.0.0.1	UDP	65	5000 → 5001 Len=23
457	9.238954718	127.0.0.1	127.0.0.1	UDP	60	5001 → 5000 Len=18
458	9.239049143	127.0.0.1	127.0.0.1	UDP	60	5000 → 5001 Len=18
459	9.239078152	127.0.0.1	127.0.0.1	UDP	56	5001 → 5000 Len=14
460	9.239167790	127.0.0.1	127.0.0.1	UDP	56	5000 → 5001 Len=14
461	9.239265910	127.0.0.1	127.0.0.1	UDP	74	5001 → 5000 Len=32
462	9.239368433	127.0.0.1	127.0.0.1	UDP	74	5000 → 5001 Len=32

1. Número de Paquetes Transmitidos

- Se observa que se transmitieron **13 paquetes UDP** entre el cliente y el servidor.
- Cada paquete contiene un tamaño de datos específico (indicados en el campo **Len**).

2. Identificación de Puertos de Origen y Destino

- **Puerto de Origen:**
 - Los paquetes originados por el cliente utilizan el puerto dinámico **5001**.
- **Puerto de Destino:**
 - Los paquetes tienen como destino el puerto **5000**, que es el puerto configurado en el servidor.

```
adrianql5 ~/Desktop/REDES/Practica3 git-[?] main - >> ./servidorUP
D 5000
Servidor UDP escuchando por el puerto 5000...
Mensaje recibido de 127.0.0.1:5001
Contenido convertido: SDFKJASFLKJASFKJ

Mensaje recibido de 127.0.0.1:5001
Contenido convertido: JKASDHFASKJH

Mensaje recibido de 127.0.0.1:5001
Contenido convertido: ADFASDFAFASFASFDASDFASFASFASDFA

Mensaje recibido de 127.0.0.1:5001
Contenido convertido: ASDFJAHSJKASKJFADFSK

Mensaje recibido de 127.0.0.1:5001
Contenido convertido: ASDFLKASLKJDSFALKSADKJL
```

```
adrianql5 ~/Desktop/REDES/Practica3 git-[?] main- >> ./cliente
UDP asa.txt 5001 127.0.0.0 5000
Socket UDP creado.
Cliente asociado al puerto 5001.

Mensaje recibido: ASDFASFASDFAIJJHJFAHJSDF
Bytes: 26

Mensaje recibido: ASFSJSFBSFBASASDFKLASDFASFASDFAIJJHJFAHJSDF
Bytes: 45

Mensaje recibido: ASFSJSFBSFBASASDFKLJJ
Bytes: 23

Mensaje recibido: SDFKJASFLKJASFKJ
Bytes: 18

Mensaje recibido: JKASDHFASKJH
Bytes: 14

Mensaje recibido: ADFASDFAFASFASDFASFSASFASDFA
Bytes: 32
```

¿Son los puertos los esperados? Sí, los puertos coinciden con los definidos en el programa:

- El puerto del cliente es asignado dinámicamente (**5001**).
- El puerto del servidor es el configurado como destino (**5000**).

3. Tamaño de los Mensajes y Bytes Transmitidos

- Los valores de longitud (**Len**) en la captura muestran que el tamaño de los datos transmitidos coincide con los mensajes definidos en el programa.

- Ejemplo:
 - Paquete inicial: **Len=26**.
 - Otros paquetes: Varían desde **Len=18** hasta **Len=87**, dependiendo del tamaño del mensaje enviado.

La longitud del mensaje UDP corresponde al tamaño total de datos enviados por el cliente y recibidos por el servidor.

4. Transmisión de un Mensaje Grande

- Cuando se envió un mensaje de mayor tamaño, los resultados fueron los siguientes:
 - **Tamaño del Paquete:** No hubo segmentación en los mensajes UDP.
 - **Paquete Único:** Cada mensaje fue transmitido en un solo paquete.

No.	Time	Source	Destination	Protocol	Length	Info
6000	123.592574431	127.0.0.1	127.0.0.0	UDP	1042	5001 → 5000 Len=1000
6001	123.592676761	127.0.0.1	127.0.0.1	UDP	1041	5000 → 5001 Len=999
6002	123.592760174	127.0.0.1	127.0.0.1	UDP	1042	5001 → 5000 Len=1000
6003	123.592924202	127.0.0.1	127.0.0.1	UDP	1041	5000 → 5001 Len=999
6004	123.592955523	127.0.0.1	127.0.0.1	UDP	648	5001 → 5000 Len=606
6005	123.592991695	127.0.0.1	127.0.0.1	UDP	648	5000 → 5001 Len=606

¿UDP segmenta los mensajes? No. UDP no realiza segmentación como TCP. Si el mensaje es más grande que el tamaño máximo permitido la segmentación la realizan otros protocolos de otras capas como el protocolo IP.

Página web

1. Análisis de un mensaje DNS:

- **Consulta DNS estándar:** En el paquete, observamos una consulta del cliente al servidor DNS (IP de destino: **193.144.75.15**). Por ejemplo:
 - **Dominio consultado:** **citius.gal**.
 - **Tipo de consulta:** **A** (IPv4) y **AAAA** (IPv6).
 - **Resultado:** El servidor responde con una dirección IPv4 (**193.144.83.107**).
- **Consultas adicionales:** El navegador también realiza consultas DNS para recursos externos como **fonts.googleapis.com** y **fonts.gstatic.com**.

ip.addr == 193.144.75.15						
No.	Time	Source	Destination	Protocol	Length	Info
8	3.672652181	172.18.11.199	193.144.75.15	DNS	81	Standard query 0x302f A citius.gal OPT
9	3.672740340	172.18.11.199	193.144.75.15	DNS	81	Standard query 0x9079 AAAA citius.gal OPT
10	3.672786640	172.18.11.199	193.144.75.15	DNS	81	Standard query 0x0bdf A citius.gal OPT
11	3.681137747	193.144.75.15	172.18.11.199	DNS	146	Standard query response 0x9079 AAAA citius.gal SOA ns.dinahosting.com OPT
12	3.681143497	193.144.75.15	172.18.11.199	DNS	97	Standard query response 0x0bdf A citius.gal A 193.144.83.107 OPT
13	3.681151290	193.144.75.15	172.18.11.199	DNS	146	Standard query response 0xf8ee AAAA citius.gal SOA ns.dinahosting.com OPT
14	3.681235035	193.144.75.15	172.18.11.199	DNS	97	Standard query response 0x302f A citius.gal A 193.144.83.107 OPT
45	3.777481266	172.18.11.199	193.144.75.15	DNS	80	Standard query 0xe4bc HTTPS fonts.googleapis.com
46	3.777915898	172.18.11.199	193.144.75.15	DNS	91	Standard query 0x2862 AAAA fonts.googleapis.com OPT
47	3.777964139	172.18.11.199	193.144.75.15	DNS	91	Standard query 0xd8c4 AAAA fonts.googleapis.com OPT
51	3.782220300	193.144.75.15	172.18.11.199	DNS	119	Standard query response 0x2862 AAAA fonts.googleapis.com AAAA 2a00:1450:4003:808::200a OPT
52	3.782220681	193.144.75.15	172.18.11.199	DNS	119	Standard query response 0xd8c4 AAAA fonts.googleapis.com AAAA 2a00:1450:4003:808::200a OPT
63	3.786698921	193.144.75.15	172.18.11.199	DNS	80	Standard query response 0xe4bc HTTPS fonts.googleapis.com
136	3.957082416	172.18.11.199	193.144.75.15	DNS	88	Standard query 0x75ad AAAA fonts.gstatic.com OPT
137	3.957150397	172.18.11.199	193.144.75.15	DNS	88	Standard query 0xe0af A fonts.gstatic.com OPT
138	3.957197757	172.18.11.199	193.144.75.15	DNS	88	Standard query 0x7674 A fonts.gstatic.com OPT
139	3.957232595	172.18.11.199	193.144.75.15	DNS	88	Standard query 0x12d2 AAAA fonts.gstatic.com OPT
141	3.960024776	193.144.75.15	172.18.11.199	DNS	116	Standard query response 0x75ad AAAA fonts.gstatic.com AAAA 2a00:1450:4003:803::2003 OPT
142	3.960026808	193.144.75.15	172.18.11.199	DNS	104	Standard query response 0x7674 A fonts.gstatic.com A 142.250.185.3 OPT
143	3.960027266	193.144.75.15	172.18.11.199	DNS	116	Standard query response 0x12d2 AAAA fonts.gstatic.com AAAA 2a00:1450:4003:803::2003 OPT
144	3.960076168	193.144.75.15	172.18.11.199	DNS	104	Standard query response 0xe0af A fonts.gstatic.com A 142.250.185.3 OPT
145	3.962877084	172.18.11.199	193.144.75.15	DNS	77	Standard query 0x39ba HTTPS fonts.gstatic.com
162	3.991077129	193.144.75.15	172.18.11.199	DNS	77	Standard query response 0x39ba HTTPS fonts.gstatic.com
306	4.175232194	172.18.11.199	193.144.75.15	DNS	89	Standard query 0xc5d6 AAAA apps.citius.usc.es OPT
307	4.175289880	172.18.11.199	193.144.75.15	DNS	89	Standard query 0x6758 A apps.citius.usc.es OPT
308	4.175322640	172.18.11.199	193.144.75.15	DNS	89	Standard query 0x3684 AAAA apps.citius.usc.es OPT
309	4.175353820	172.18.11.199	193.144.75.15	DNS	89	Standard query 0xb9f0 A apps.citius.usc.es OPT
312	4.179058101	193.144.75.15	172.18.11.199	DNS	139	Standard query response 0x6758 A apps.citius.usc.es CNAME ctcloud53.inv.usc.es A 172.16.242.53 OPT
313	4.179096364	193.144.75.15	172.18.11.199	DNS	181	Standard query response 0x3684 AAAA apps.citius.usc.es CNAME ctcloud53.inv.usc.es SOA secus.usc.es OPT
314	4.180506329	193.144.75.15	172.18.11.199	DNS	139	Standard query response 0xb9f0 A apps.citius.usc.es CNAME ctcloud53.inv.usc.es A 172.16.242.53 OPT
315	4.180927398	193.144.75.15	172.18.11.199	DNS	181	Standard query response 0xc5d6 AAAA apps.citius.usc.es CNAME ctcloud53.inv.usc.es SOA secus.usc.es OPT
321	4.180922240	172.18.11.199	193.144.75.15	DNS	91	Standard query 0x7415 AAAA ctcloud53.inv.usc.es OPT
322	4.180994137	172.18.11.199	193.144.75.15	DNS	91	Standard query 0x5232 AAAA ctcloud53.inv.usc.es OPT
329	4.187023703	193.144.75.15	172.18.11.199	DNS	149	Standard query response 0x7415 AAAA ctcloud53.inv.usc.es SOA secus.usc.es OPT
330	4.187023709	193.144.75.15	172.18.11.199	DNS	149	Standard query response 0x5232 AAAA ctcloud53.inv.usc.es SOA secus.usc.es OPT
331	4.189579035	172.18.11.199	193.144.75.15	DNS	78	Standard query 0xa424 HTTPS apps.citius.usc.es
333	4.192010400	193.144.75.15	172.18.11.199	DNS	170	Standard query response 0xa424 HTTPS apps.citius.usc.es CNAME ctcloud53.inv.usc.es SOA secus.usc.es

2. HTTP/HTTPS:

- Una vez que las resoluciones DNS son exitosas, es de esperar que el navegador inicie:
 - **Peticiones HTTP/HTTPS:** Para descargar contenido (HTML, CSS, imágenes, etc.).

- **Respuesta del servidor:** Generalmente incluye un código de estado (como **200 OK**) y el contenido solicitado.

10169	188.793494190	172.18.11.199	185.199.109.153	HTTP	259 GET /pkgs/x86_64/gh0stzk-dotfiles.db HTTP/1.1
10178	188.819672591	185.199.109.153	172.18.11.199	HTTP	228 HTTP/1.1 301 Moved Permanently (text/html)
10377	192.750298144	172.18.11.199	95.216.195.133	HTTP	154 GET /nm-check.txt HTTP/1.1
10379	192.824894529	95.216.195.133	172.18.11.199	HTTP	270 HTTP/1.1 200 OK (text/plain)