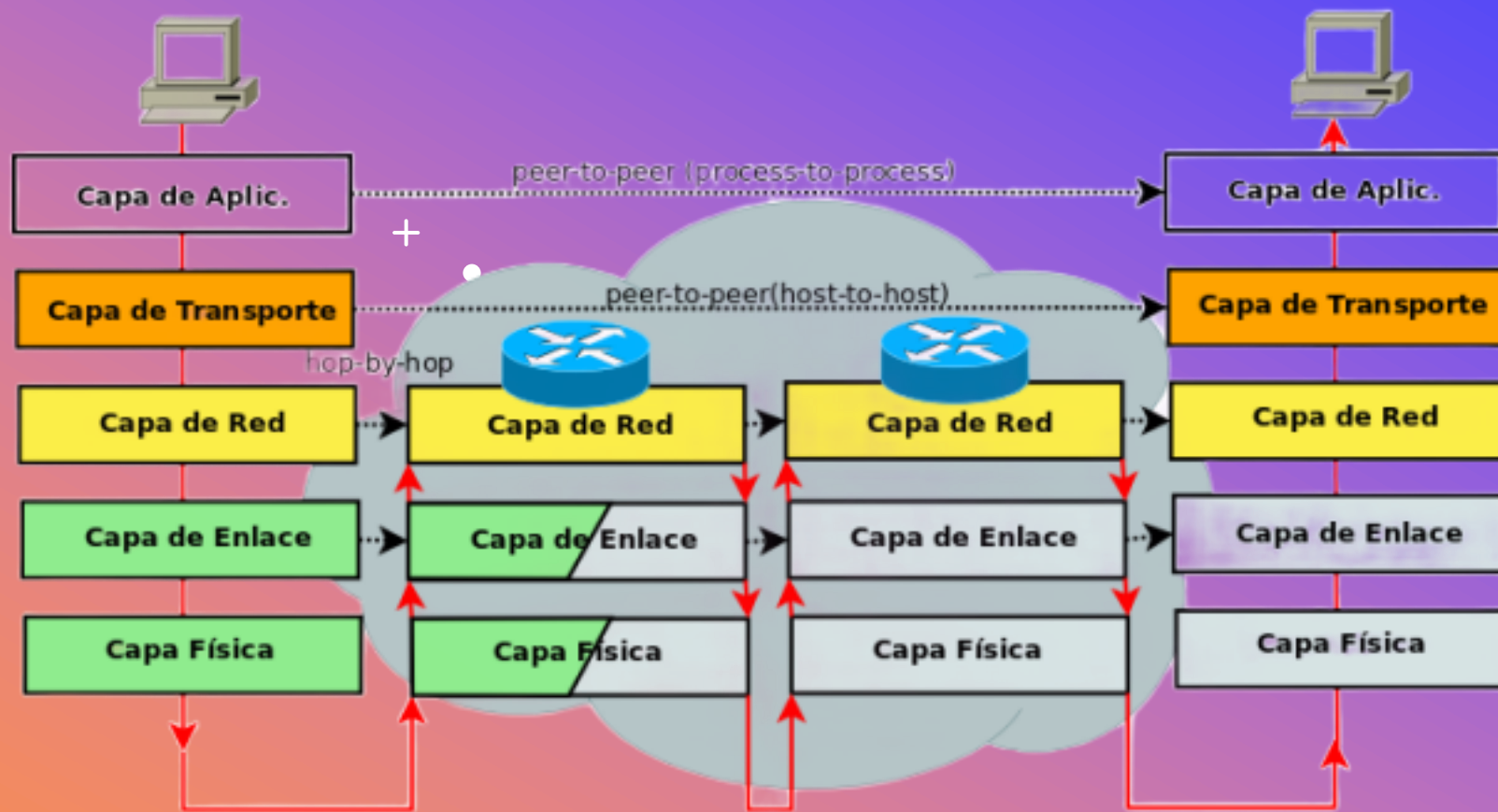


# CAPA TRANSPORTE <sup>+</sup><sub>o</sub> •



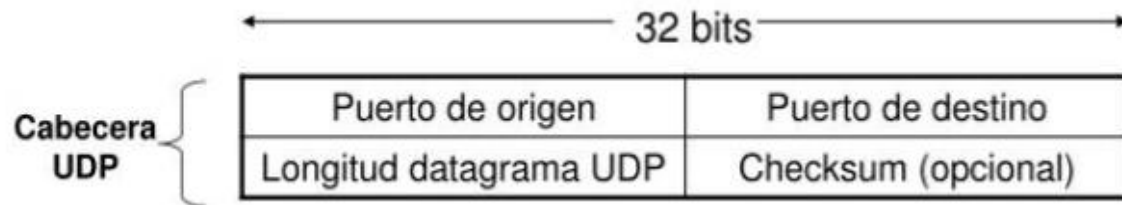
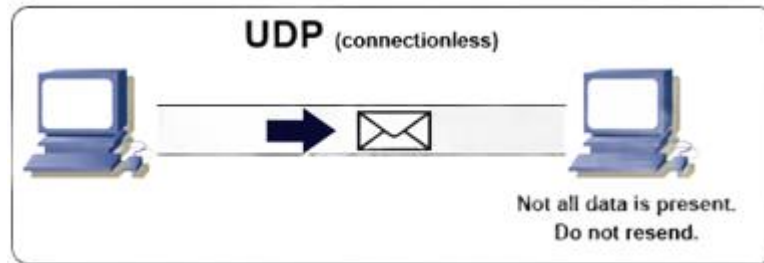


## Funcionalidad de la Capa de Transporte



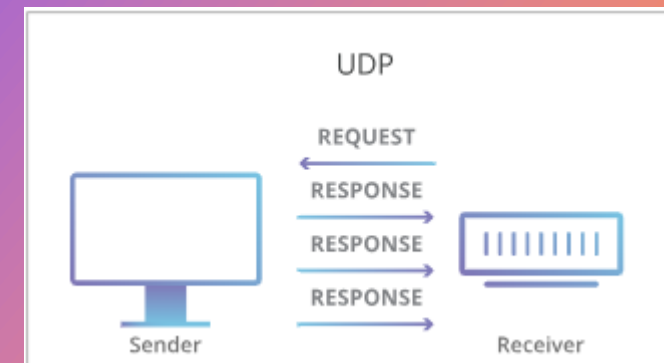
- Encapsulación, define PDU donde se envía los mensajes de la aplicación.
- MUX/DEMUX process-to-process (puertos).
- Soporte de datos de tamaños arbitrarios.
- Control y Detección de Errores, pérdida, duplicación, se corrompen.
- Control de Flujo.
- Control de Congestión.
- Dos modelos básicos:
- Modelo Confiable: TCP. Modelo NO Confiable: UDP.

# UDP

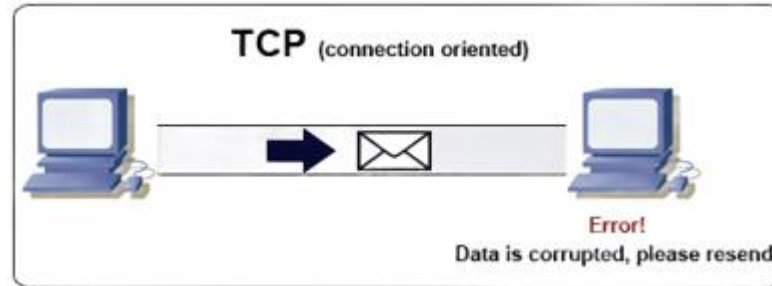


Bits	0 - 15	16 - 31
0	Puerto origen	Puerto destino
32	Longitud del mensaje	Suma de verificación
64	Datos	

- Protocolo liviano.
- Best Effort.
- MUX/DEMUX.
- Detección de algunos errores – Checksum
- No requiere establecimiento de conexión.
- Aplicaciones: Rápidas / eficientes.
- No incrementa Overhead end-to-end.
- No requiere establecimiento de conexión



# TCP

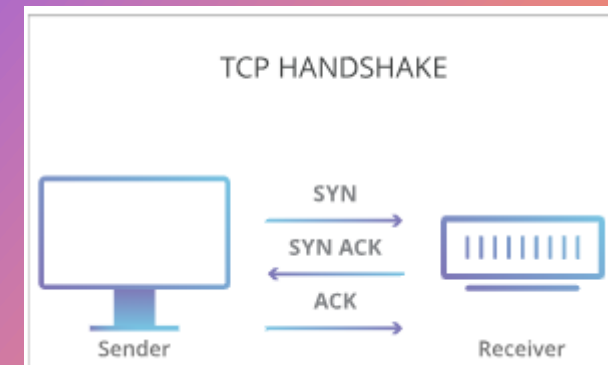


Cabecera  
TCP

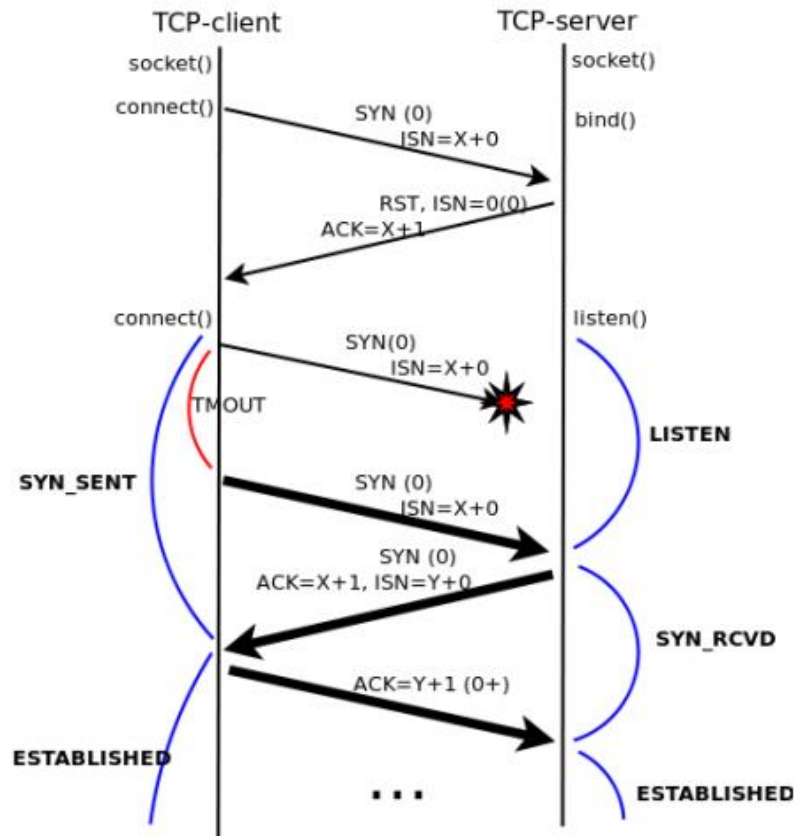
Puerto de origen		Puerto de destino	
Número de secuencia			
Número de acuse de recibo			
L. Cab. (4 bits)	Resv. (4 bits)	Flags (8 bits)	Tamaño ventana
Checksum			Puntero datos urgentes
Opciones			Relleno

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

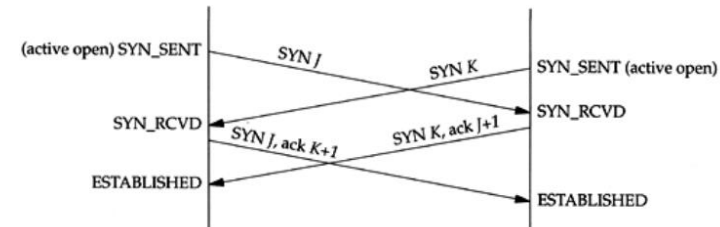
- Protocolo confiable, ordenado, con buffering, control de errores, flujo y de congestión.
- Provee MUX/DEMUX.
- Incrementa Overhead end-to-end para ofrecer confiabilidad.
- Requiere establecimiento de conexión (y cierre).
- Aplicaciones: transferencia de archivos, FTP/HTTP/SMTP/acceso remoto(SSH, telnet,...)
- Incrementa Overhead end-to-end para ofrecer confiabilidad.
- Requiere establecimiento de conexión (y cierre).



# 3 WAY HANDSHAKE

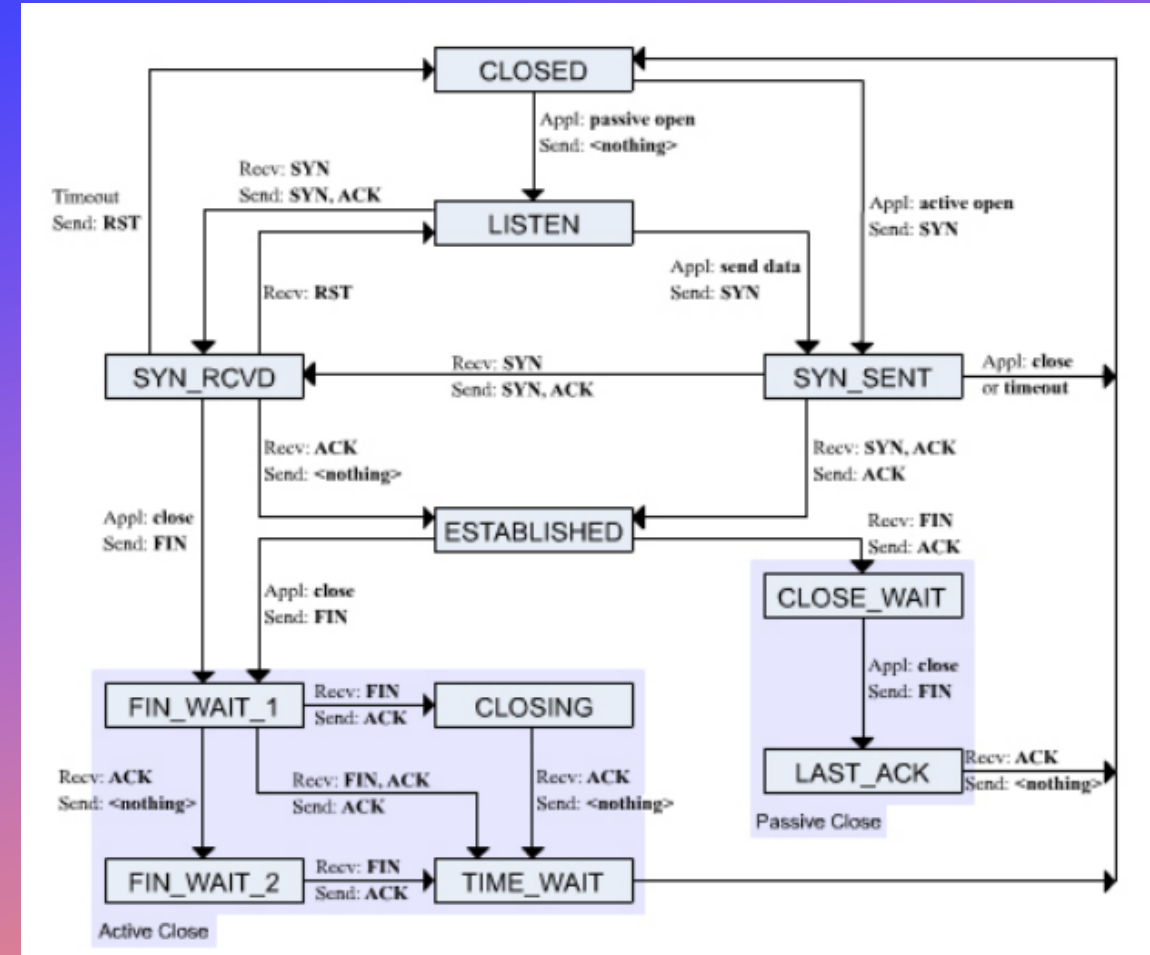
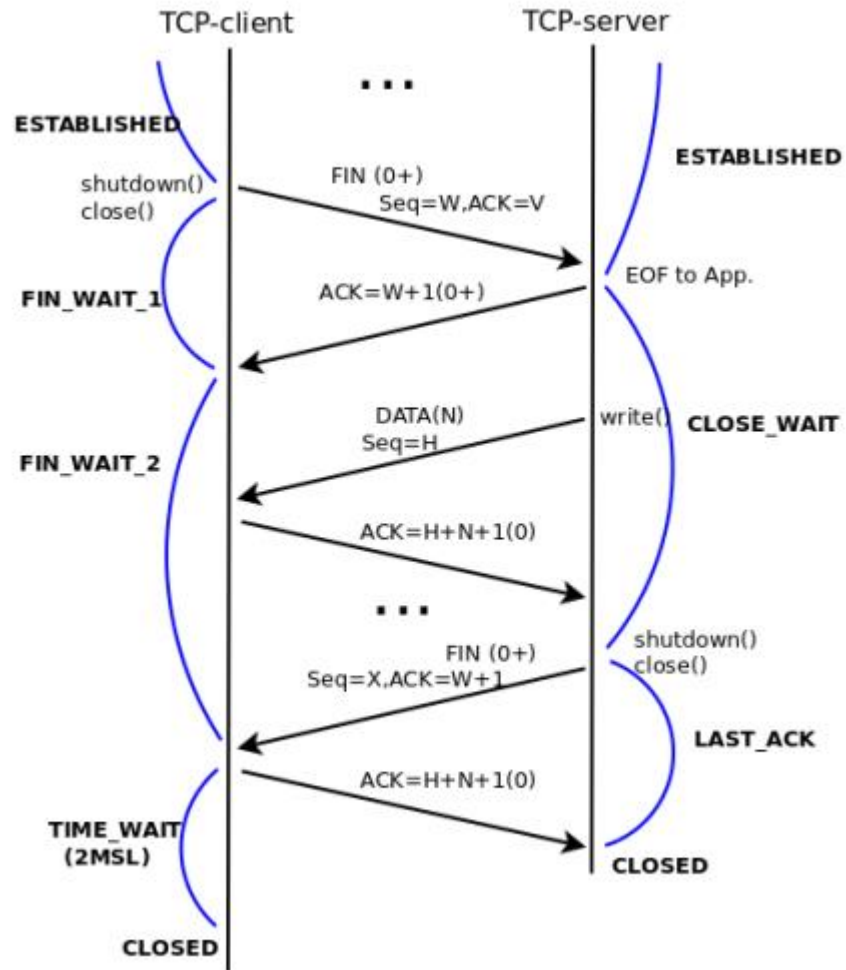


- Flags: SYN (Synchronize), ACK (Acknowledge) y RST (Reset). 3Way-Handsake (3WH).
- En el 3 segmento se puede enviar info.
- El ISN (Initial Sequence Number), se utiliza un contador que se incrementa cada 4 mseg.
- RST si no hay proceso en estado LISTEN.
- Open Pasivo (servidor) y Activo (cliente).
- Open simultáneo.





# TCP Close





netstat -an      ss -an

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	127.0.0.1:49670	127.0.0.1:49671	ESTABLISHED
TCP	127.0.0.1:49671	127.0.0.1:49670	ESTABLISHED
TCP	127.0.0.1:49672	127.0.0.1:49673	ESTABLISHED
TCP	127.0.0.1:49673	127.0.0.1:49672	ESTABLISHED
UDP	0.0.0.0:5050	*:*	
UDP	0.0.0.0:5353	*:*	
UDP	0.0.0.0:5355	*:*	
UDP	0.0.0.0:50921	216.239.32.116:443	
UDP	0.0.0.0:53781	*:*	
UDP	0.0.0.0:54097	142.251.133.14:443	
UDP	0.0.0.0:54562	142.251.133.46:443	

Local Address (Dirección Local):  
Indica la dirección IP y el número de puerto en la máquina local que está utilizando la conexión.

Foreign Address (Dirección Remota):  
Esta columna muestra la dirección IP y el número de puerto del dispositivo remoto al que tu máquina está conectada o que intentará conectarse a tu máquina.



```
hping3 127.0.0.1 -p 22 -S
```

```
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes  
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=22 flags=SA seq=0 win=43690 rtt=2.4 ms
```

**Descripción:** Como hay un servicio (por ejemplo, un **servidor SSH**) escuchando en el puerto **22**, la respuesta es un segmento con los flags **SYN y ACK activados**, lo que indica que el servicio está listo para establecer una conexión.

**Resultado esperado:** El servidor SSH recibe el SYN, y responde con SYN-ACK. Esto confirma que está listo para seguir con el handshake.

```
hping3 127.0.0.1 -p 40 -S
```

```
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes  
len=40 ip=127.0.0.1 ttl=64 DF id=17557 sport=40 flags=RA seq=0 win=0 rtt=1.3 ms
```

**Descripción:** No hay ningún servicio escuchando en el puerto **40**, por lo que el host responde con un segmento con los flags **RST y ACK activados**

**Resultado esperado:** El servidor recibe el SYN, pero no tiene ningún servicio en el puerto 40, entonces envía un RST para terminar el intento de conexión.





```
servidor# ss -natu | grep 110
```

```
tcp    LISTEN 0 0          *:110          *:*
tcp    SYN-RCV 0 0          157.0.0.1:110  157.0.11.1:52843
```

```
cliente# ss -natu | grep 110
```

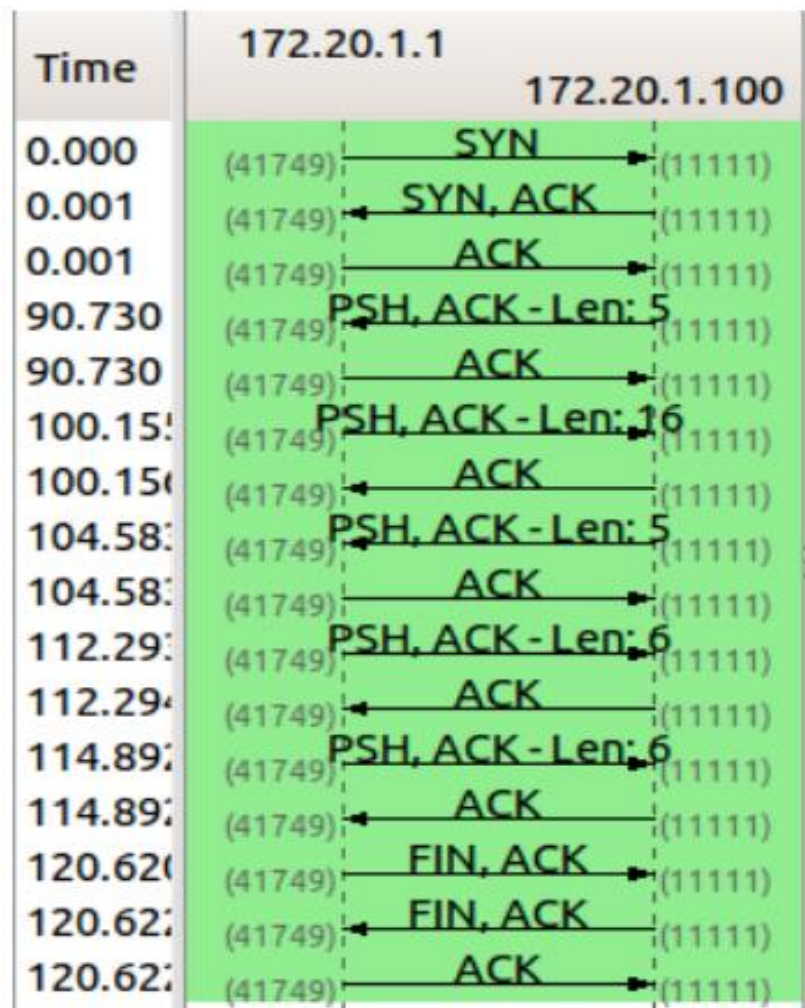
```
tcp    SYN-SENT 0 1          157.0.11.1:52843  157.0.0.1:110
```

¿Qué segmentos llegaron y cuáles se están perdiendo en la red?

¿A qué protocolo de capa de aplicación y de transporte se está intentando conectar el cliente?

¿Qué flags tendría seteado el segmento perdido?

# TCP – EJERCICIO



Ejercicio;  
Tips:

Posicionarnos mentalmente , veo flags, veo conexión, PIENSO EN TCP

Identificar secuencias de establecimiento de conexión (Flags)

Identificar los paquetes que envían datos.

Pensar en qué momentos incremento en 1.

Pensar en las transiciones de estados que ocurren.

# GRACIAS! 😊

× •  
○ ¿Preguntas?