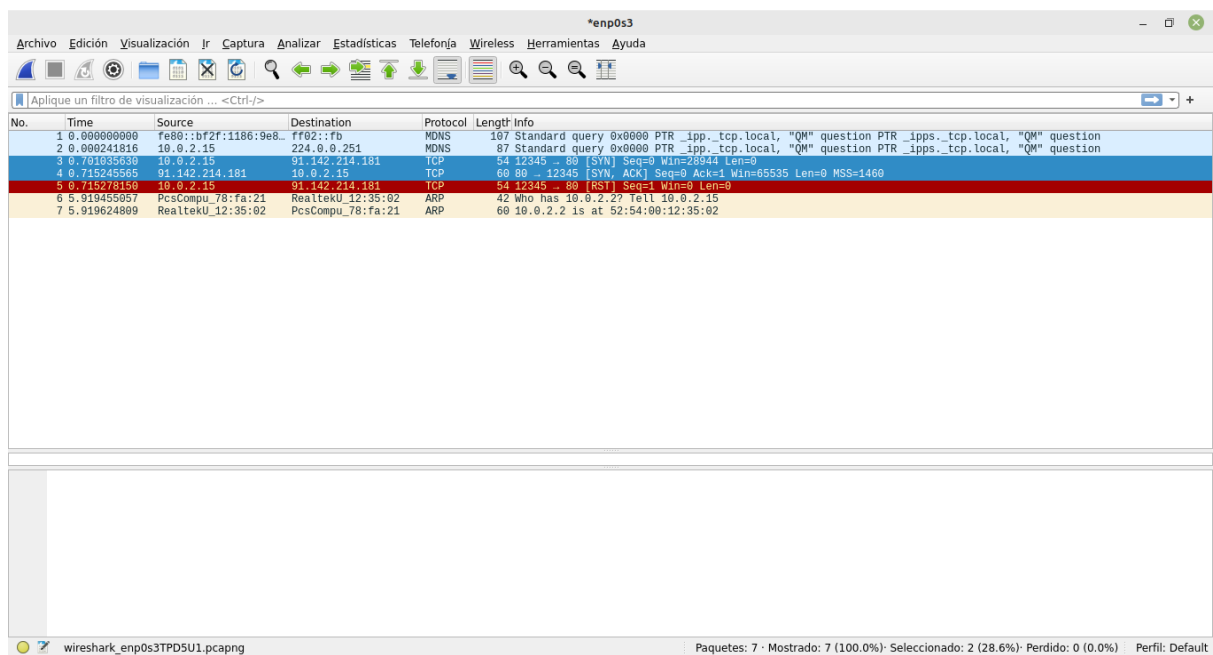


Crea un paquete TCP SYN que vaya a 91.142.214.181 , escucha con Wireshark y observa si obtienes la respuesta.

```
dagazo@dagazo: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
dagazo@dagazo:~$ sudo python3 send_first_packet.py  
[sudo] contraseña para dagazo:  
dagazo@dagazo:~$ cat send_first_packet.py  
import socket  
  
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)  
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)  
  
ip_header = b'\x45\x00\x00\x28' # Version, IHL, Type of Service | Total Length  
ip_header += b'\xab\xcd\x00\x00' # Identification | Flags, Fragment Offset  
ip_header += b'\x40\x06\x0a\xec' # TTL, Protocol | Header Checksum  
ip_header += b'\x0a\x00\x02\x0f' # Source Address  
ip_header += b'\x5b\x8e\xd6\xb5' # Destination Address  
  
tcp_header = b'\x30\x39\x00\x50' # Source Port | Destination Port  
tcp_header += b'\x00\x00\x00\x00' # Sequence Number  
tcp_header += b'\x00\x00\x00\x00' # Acknowledgement Number  
tcp_header += b'\x50\x02\x71\x10' # Data Offset, Reserved, Flags | Window Size  
tcp_header += b'\xc1\xf6\x00\x00' # Checksum | Urgent Pointer  
  
packet = ip_header + tcp_header  
s.sendto(packet, ('91.142.214.181', 0))  
dagazo@dagazo:~$
```

1-Crea un pantallazo de lo mostrado en Wireshark

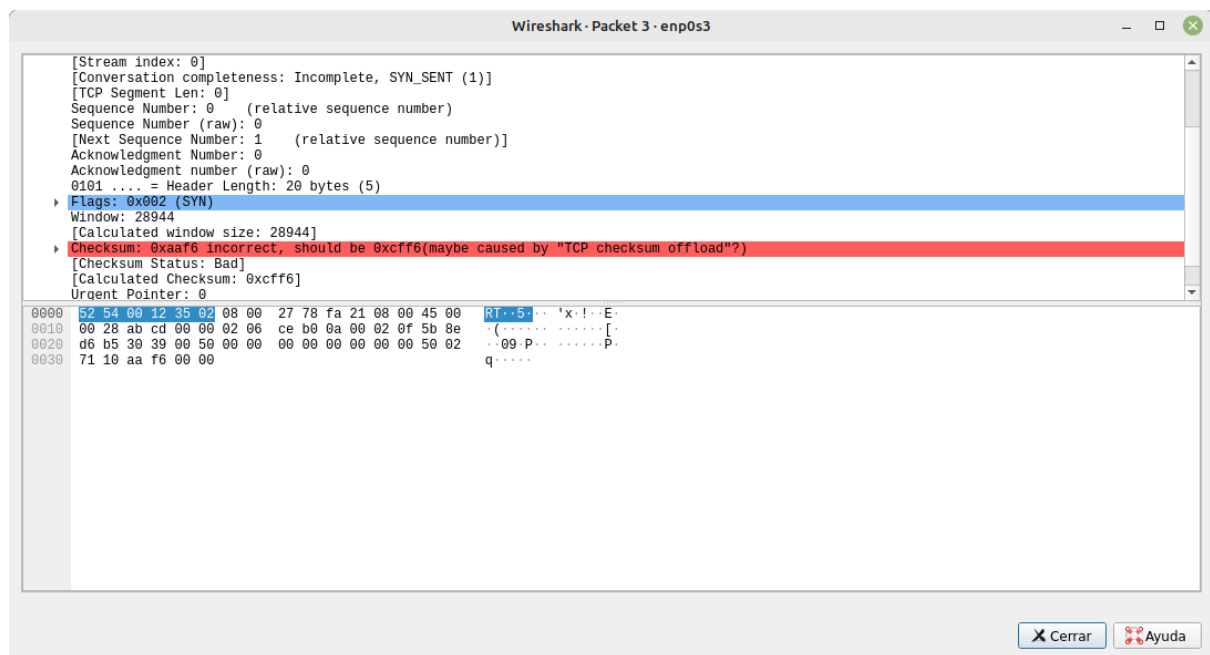


La Arboleda		Curso 2022/2023
David García Zorzo	REDES	

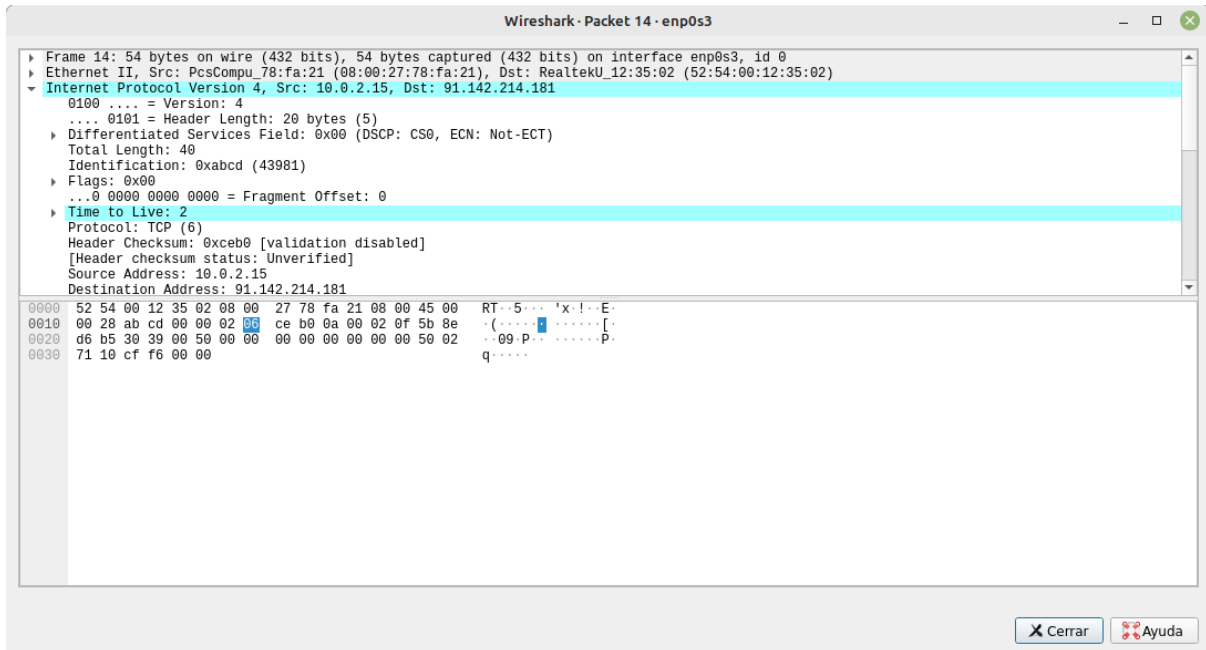
2-¿Qué flags tiene "encendidos" tu paquete?, ¿y el de vuelta?

SYN, El de vuelta tiene SYN ACK

3-Pon mal el checksum y observa qué pasa



4-Pon un TTL=2 y observa qué pasa



## Creación de paquete ICMP

```
dagazo@dagazo: ~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
GNU nano 6.2  paqueteicmp.py  
import socket  
  
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)  
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)  
  
ip_header = b'\x45\x00\x00\x1c' # Version, IHL, Type of Service | Total Length  
ip_header += b'\xab\xcd\x00\x00' # Identification | Flags, Fragment Offset  
ip_header += b'\x40\x01\x6b\xd8' # TTL, Protocol | Header Checksum  
ip_header += b'\x0a\x00\x02\x0f' # Source Address  
ip_header += b'\x08\x08\x08\x08' # Destination Address  
  
icmp_header = b'\x08\x00\xe5\xca' # Type of message, Code | Checksum  
icmp_header += b'\x12\x34\x00\x01' # Identifier | Sequence Number  
  
packet = ip_header + icmp_header  
s.sendto(packet, ('8.8.8.8', 0))  
  
[ 16 líneas leídas ]  
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación  
^X Salir  ^R Leer fich.  ^\ Reemplazar  ^U Pegar  ^J Justificar  ^_ Ir a línea
```

Observamos en wireshark la respuesta al paquete ICMP

The image shows a Wireshark packet capture on interface enp0s3. The packet list pane displays 12 packets, all of which are ICMP Echo (ping) requests and replies. The first packet is a request from 10.0.2.15 to 8.8.8.8. The subsequent packets are replies from 8.8.8.8 to 10.0.2.15. The packet details pane shows the structure of the ICMP Echo (ping) request, including the type, code, and sequence number. The packet bytes pane shows the raw data of the first packet, which is a 42-byte ICMP Echo (ping) request.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x1234, seq=1/256, ttl=64 (reply in 2)
2	0.01003383	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x1234, seq=1/256, ttl=111 (request in 1)
3	2.171314713	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x1234, seq=1/256, ttl=64 (reply in 4)
4	2.180878786	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x1234, seq=1/256, ttl=111 (request in 3)
5	3.549359248	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x1234, seq=1/256, ttl=64 (reply in 6)
6	3.563617249	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x1234, seq=1/256, ttl=111 (request in 5)
7	4.253604696	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x1234, seq=1/256, ttl=64 (reply in 8)
8	4.262715763	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x1234, seq=1/256, ttl=111 (request in 7)
11	5.328856173	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x1234, seq=1/256, ttl=64 (reply in 12)
12	5.338624045	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x1234, seq=1/256, ttl=111 (request in 11)

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0

```
0000 52 54 00 12 35 02 08 00 27 78 fa 21 08 00 45 00  RT..5... 'x'!..E..
0010 00 1c ab cd 00 00 40 01 b2 15 0a 00 02 0f 08 08  ....@.....
0020 08 08 08 00 e5 ca 12 34 00 01  ....4..
```

Internet Control Message Protocol: Protocol Paquetes: 12 · Mostrado: 10 (83.3%) · Perdido: 0 (0.0%) Perfil: Default