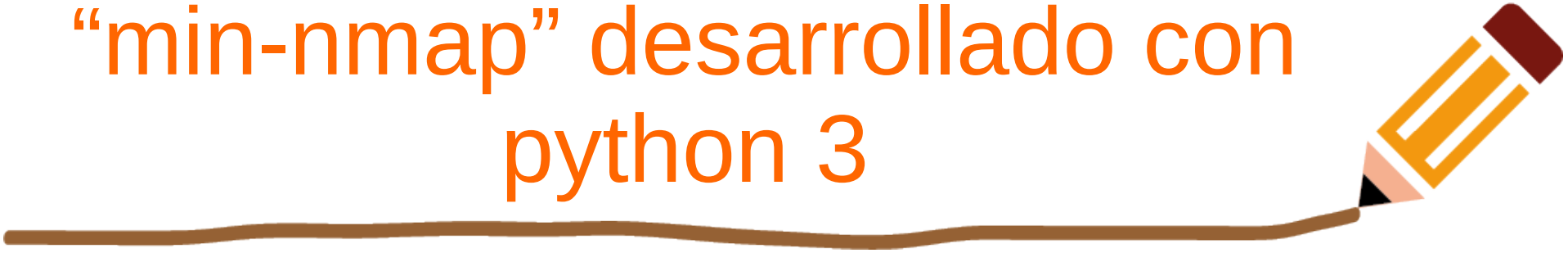
“min-nmap” desarrollado con python 3



Eduardo Eliezar Castillo Hernández

Contenido

- Definición de sondeo de puertos
- Estados de los puertos de un equipo
- Protocolos necesarios
  - IP
  - ICMP
  - TCP
  - UDP
- Detección de hosts objetivos
  - ICMP PING
  - TCP SYN
- Sondeo de puertos
  - TCP SYN
  - UDP PING

Sondeo de puertos

- El termino se refiere a la acción de analizar el estado de los puertos de un equipo conectado a una red de comunicaciones.
- Con el objetivo de determinar que servicios esta ofreciendo y las posibles vulnerabilidades que pueda tener según los puertos que se encuentren abiertos.

Estados de los puertos de un equipo

Un puerto puede tener 3 estados básicos:

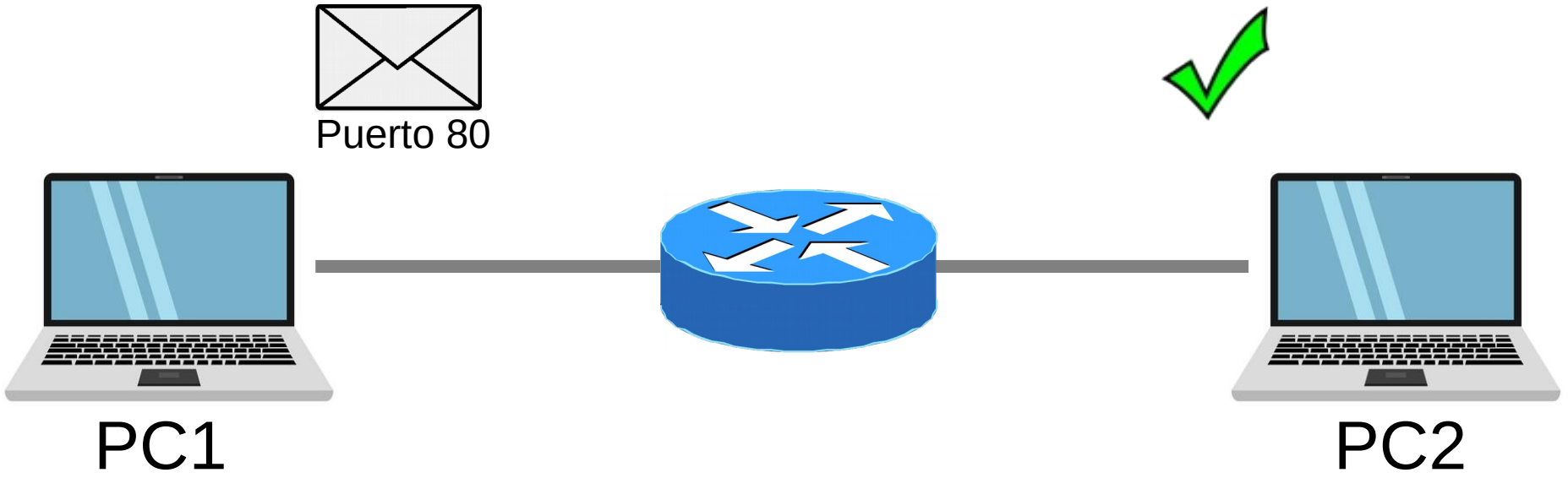
- **Abierto:** el puerto tiene una aplicación escuchando en el, y acepta conexiones TCP o paquetes UDP.
- **Cerrado:** el puerto es accesible y alcanzable, pero no tiene una aplicación escuchando en él.
- **Filtrado:** el puerto no es accesible debido a que a un filtro esta impidiendo que los paquetes lo alcancen. Esto puede deberse a un dispositivo cortafuegos, las reglas de acceso de un router o a un cortafuegos instalado en el equipo.

Estados de los puertos de un equipo (II)

Ademas de los anteriores 3 estados, se puede clasificar un puerto en 3 estados mas:

- **No filtrado:** el puerto es accesible pero no se puede determinar si esta abierto o cerrado.
- **Abierto | Filtrado:** no es posible determinar si el puerto esta abierto o filtrado.
- **Cerrado | Filtrado:** no es posible determinar si el puerta esta cerrado o filtrado.

Puerto: Abierto

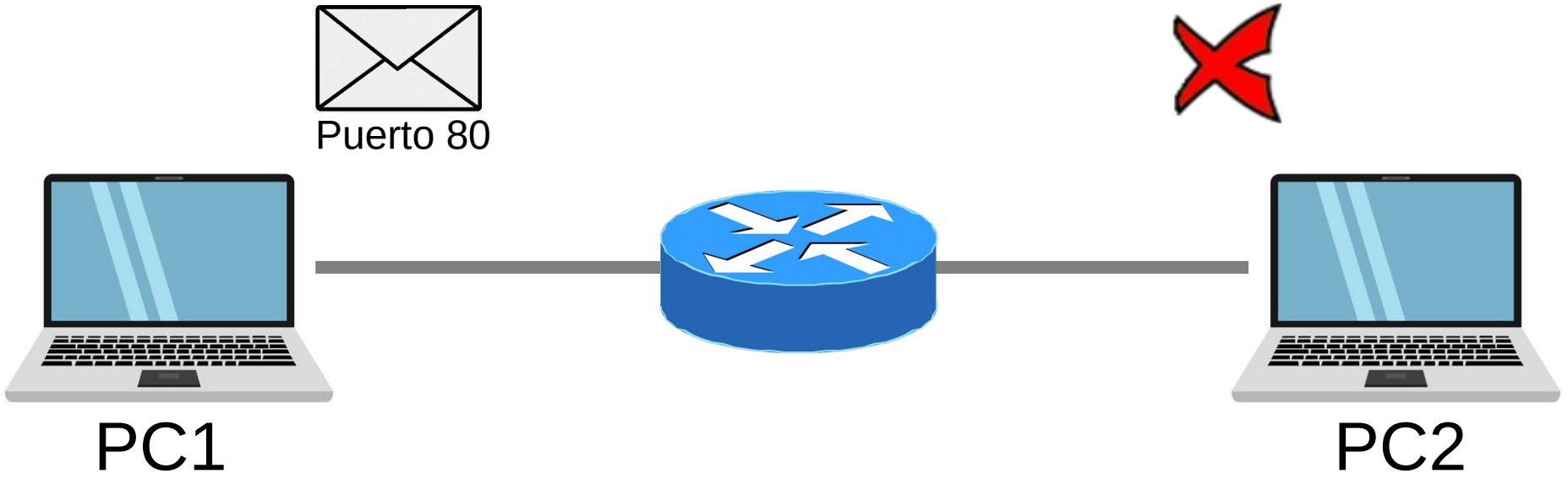


***PC2 recibe el paquete, lo acepta e informa a PC1 que lo ha recibido y aceptado.**

UNAN - León
Departamento de Computación

python™

Puerto: Cerrado

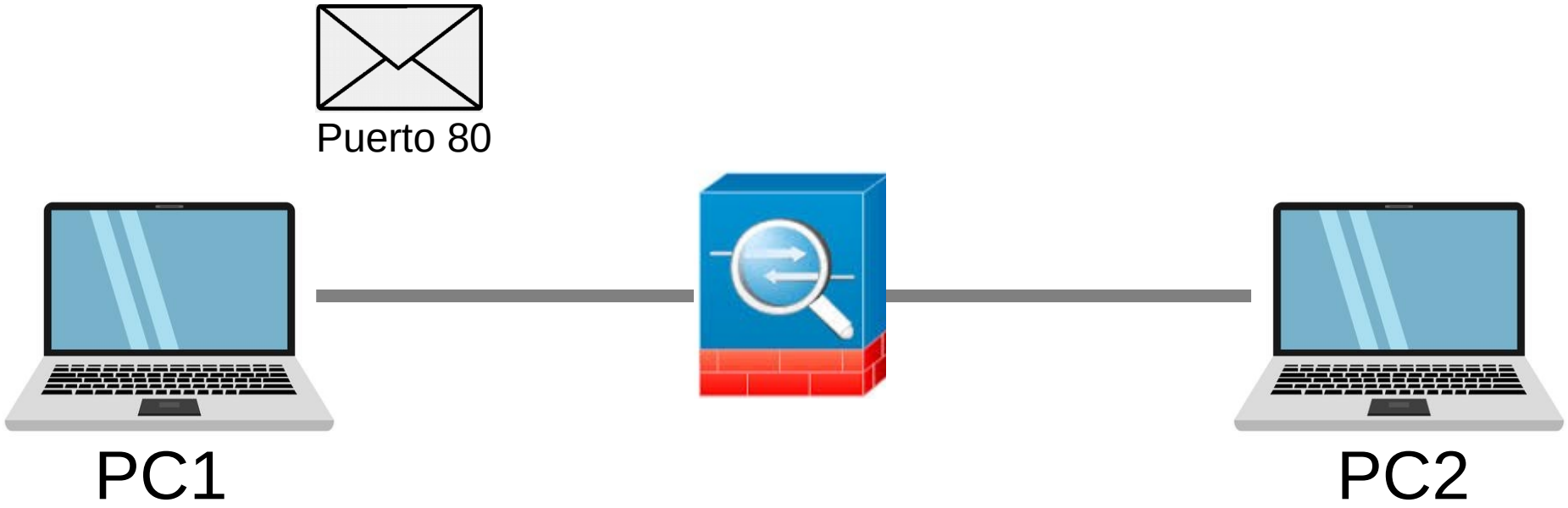


***PC2 recibe el paquete, no lo acepta, pero informa a PC1 que lo ha recibido y no lo ha aceptado.**

UNAN - León
Departamento de Computación

python™

Puerto: Filtrado



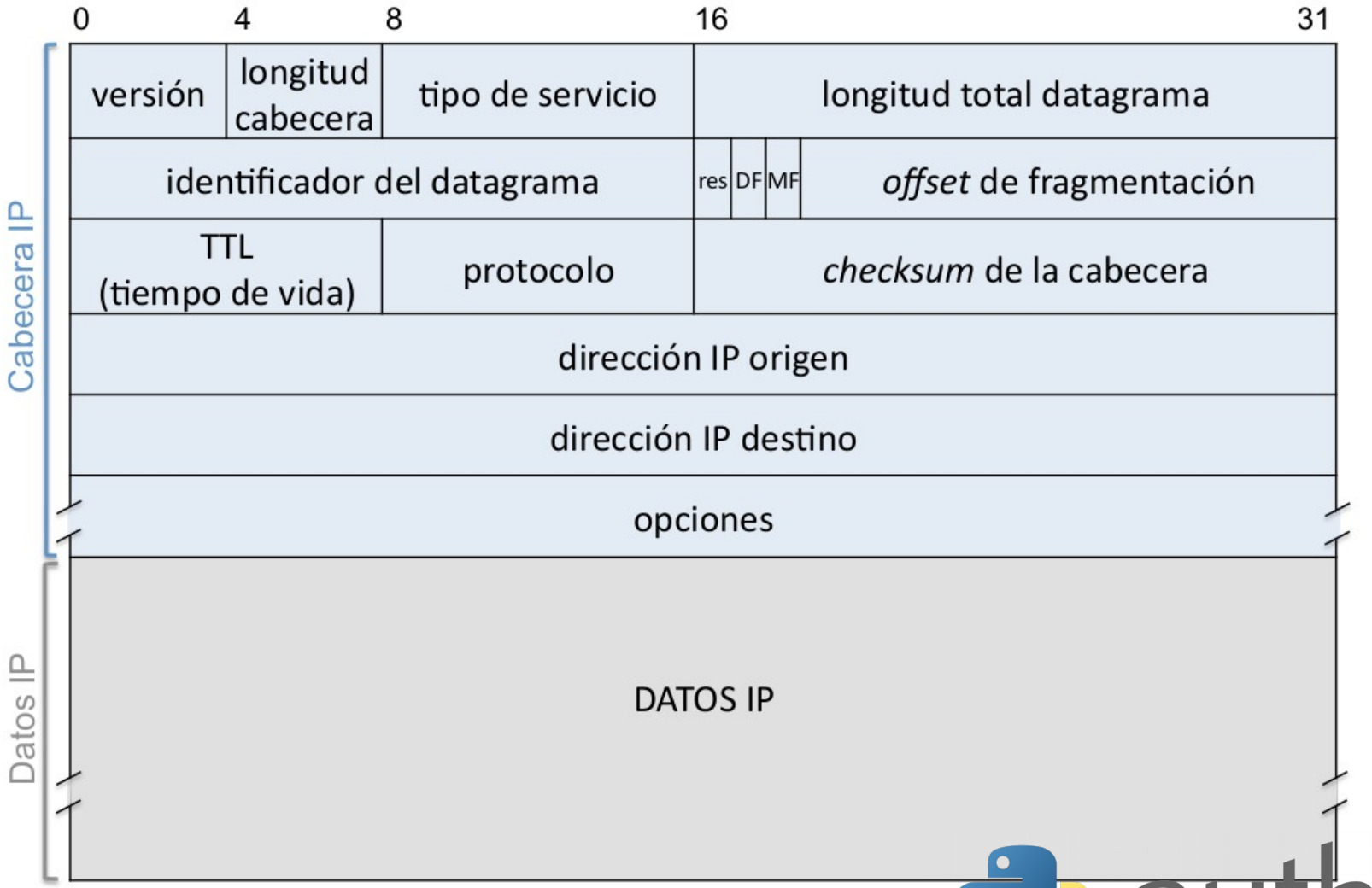
***PC2 nunca recibe el paquete.**

***PC1 no sabe si el paquete fue recibido o no, ya que no recibe ninguna respuesta.**

UNAN - León
Departamento de Computación

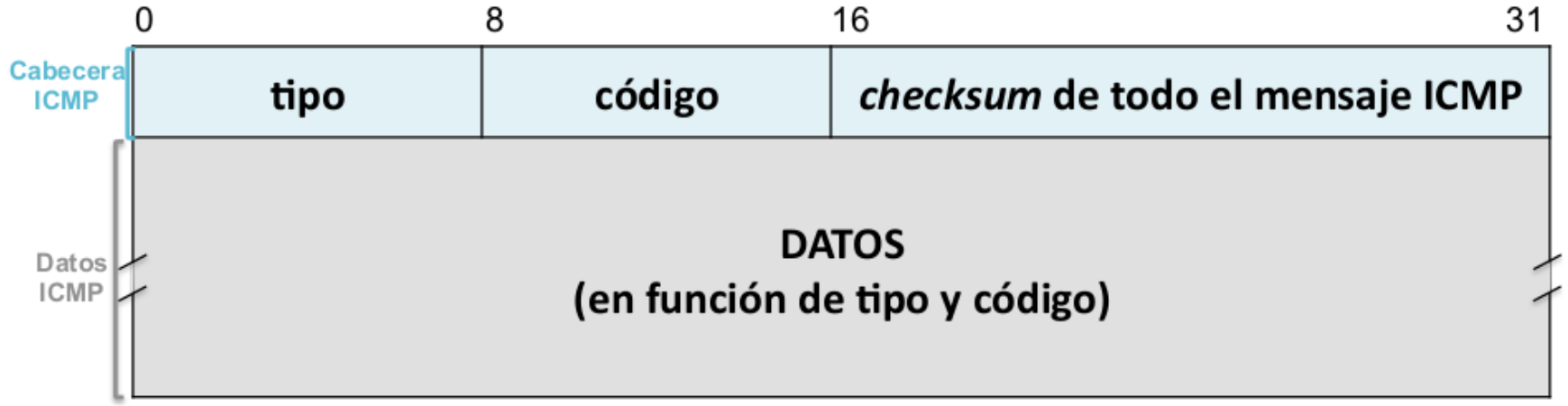
python™

Protocolos necesarios: IP



**Min-map no calcula el Checksum del paquete, delega esta tarea al sistema operativo*.

Protocolos necesarios: ICMP



UNAN - León
Departamento de Computación

python™

ICMP: Tipos y códigos utilizados

Tipo	Código	Descripción
0	0	Respuesta de ECO (ECHO REPLY)
3	1	Host inalcanzable
3	2	Protocolo inalcanzable
3	3	Puerto inalcanzable
3	9	Red de destino prohibida administrativamente
3	10	Host de destino prohibido administrativamente
3	13	Comunicación prohibida administrativamente mediante filtrado.
8	0	Solicitud de eco (ECHO REQUEST)

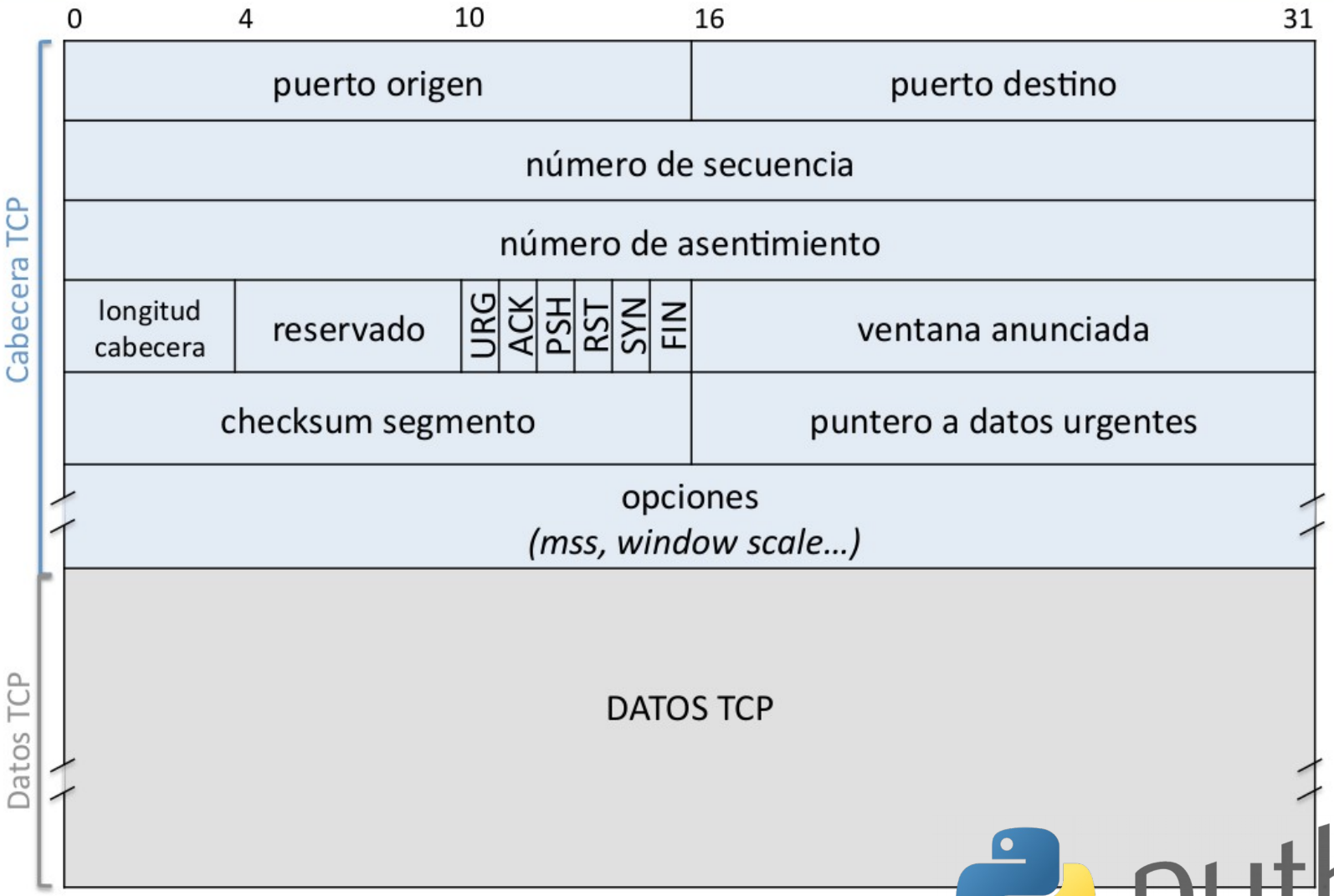
Checksum ICMP

Tipo 1 Byte
Código 1 Byte
Checksum 2 Bytes
Identificador 2 Bytes
Secuencia 2 Bytes
Datos N Bytes

Pasos:

- Rellenar campo Checksum con ceros.
- Dividir el paquete completo en palabras de 2 Bytes (16 bits).
  - Si el numero de palabras resultantes es impar, agregar una palabra (2 Bytes) de 0.
- Sumar las palabras obtenidas.
- Si el resultado de la suma es de mas de 16 bits (2 Bytes), tomar los primeros 16 bits de menos significativos y sumarle los bits sobrantes.

Protocolos necesarios: TCP



Pseudocabecera TCP