

# Preguntas Codigos C2

---

## Stop & Wait

**1. ¿Qué pasaría si en medio de estar recibiendo un mensaje se recibe un segmento FIN? Responda en base a su propio código.**

R: Como no existe un control respecto a este tipo de casos, en base al propio codigo simplemente se recibiría el mensaje, enviando el ACK de respuesta correspondiente, ahora bien como el mensaje FIN no posee area de datos, o mejor dicho segun la implementacion posee un area de datos de largo 0 entonces lo procesa como si fuese un mensaje como cualquier otro.

**2. En su implementación de socketTCP implementa 2 metodos `recv`, `recv` y `_recv`, explique como consolidaría el funcionamiento de ambos bajo un único método.**

R: Bastaria con migrar el codigo de `_recv` hacia `recv` pues la implementacion de `recv` se mantiene en loop esperando recibir el mensaje TCP que posee el largo del mensaje a recibir, si este no es recibido simplemente se mantiene en el loop, al obtener el largo se sale del loop para seguir al proceso de recibir el mensaje como tal. Cabe mencionar que `recv` tiene implementado el control de mensajes para verificar si el mensaje TCP que llega es la continuacion de otro mensaje o es el mensaje que indica el largo del mensaje a enviar.

**3. Explique lo mismo para sus métodos `send` y `_send`**

Para consolidar ambos metodos en este caso bastaria con envolver el contenido de la funcion `send` en un loop con condicion siempre verdadera, al igual que para `recv`, donde la condicion para salir de este sera el momento en que llegue el ACK enviado por la otra parte (ACK respectivo al mensaje con el largo del mensaje que enviaremos), en tal caso lo que continuaria seria el contenido de la funcion `_send` para el manejo del envio del mensaje como tal.