# 4. Chat 0.2

**4.3. Incidencias a resolver**

Se propone al alumnado comprobar si estas incidencias están resueltas y resolver todas aquellas no resueltas en el programa:

* + T01: Agregar información sobre el socket del cliente y del servidor antes del apodo: [ip:puerto]. Ejemplo: [172.25.240.1:53630] ANA: Hola
  + T02: Modificar el programa del cliente para que reciba la despedida del servidor y la muestre por pantalla
  + ~~T03: Permitir que la despedida sea parametrizable en línea de comandos~~
  + T04: Crear un método llamado finalizar que reciba un código de error y lo use como código de salida del proceso para el sistema operativo. Ejemplo del cuerpo del método finalizar: System.exit(e.getNumero())
  + T05: Controlar con try-catch en el método main de la clase ClienteChat que el puerto del servidor recibido en línea de comandos es un número entero. E.o.c. error, uso y finalizar.
  + T06: En ClienteChat, usar parámetro args[1] como IP del servidor controlando formato
  + T07: Cazar las distintas excepciones (ver documentación de la API de Java) que pueden ocurrir en el bloque try del método iniciar de la clase ClienteChat. Agregar errores asociados a EchoError. Usar ls métodos error y finalizar
  + T08: Controlar el rango válido para un puerto (1-65535). Crear y usar un tipo enumerado llamado Conf que contenga los elementos MIN\_PUERTO y MAX\_PUERTO con sus valores como atributos privados del tipo enumerado avanzado.
  + T09: El formato en línea de comandos sería el siguiente:  
    ServidorChat [puerto] [despedida] -> Por defecto usar parámetros de Conf (definir DESPEDIDA en Conf)  
    ClienteChat apodo ip [puerto]
  + T10: Definir la palabra de finalización en Conf y usarla en el código (FIN)
  + T11: Definir el mensaje de despedida del servidor en Conf y usarlo en el código
  + T12: Definir MAX\_CLIENTES en Conf y usarlo en el código (máximo número de clientes conectados al servidor en un momento determinado)
  + M01: Modificar el programa para que sea configurable la visualización (o no) de la información del socket en un mensaje del cliente y/o del servidor
  + M02: Permitir que la palabra clave de finalización sea parámetrizable en línea de comandos
  + M03: Modificar la notación para clienteSocket en las clases ServidorChat y GestorCliente. En ServidorChat se usará el identificador gestorClienteSocket y en GestorCliente se usará simplemente el identificador socket. La idea es hacer hincapié en que el socket es un extremo de la conexión con el cliente que está en la parte del servidor.
  + M04: Limpieza: eliminación de manifiesto sobrante en src e imagen jpg sobre incidencias en raíz. Quito import de DatagramSocket (no necesario) en ServidorChat. Elimino variable linea (no usada) en ClienteChat. Elimino @NotNull de parámetro primitivo puertoServidor en ClienteChat
  + M05: Mostrar nombres y versiones del cliente y servidor